МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет	*Кафедра*
вычислительной техники	МОиПЭВМ

Направление подготовки 09.03.04 «Программная инженерия»

Направленность Программная инженерия

БАКАЛАВРСКАЯ РАБОТА

на тему

МОБИЛЬНОЕ ПРОГРАММНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОДДЕРЖКИ ПРОЦЕССОВ ФУНКЦИОНИРОВАНИЯ АВТОШКОЛЫ

Студент _____ Ульченко Фёдор Сергеевич
(подпись, дата) (ФИО полностью)

Руководитель _____ Князев В.Н.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр _____ Такташкин Д.В.
(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____№_____)

Заведующий кафедрой _____ Козлов А.Ю.
(подпись) (фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____№____)

Секретарь ГЭК _____ Попова Н.А.
(подпись) (фамилия, инициалы)

Пенза, 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет Вычислительной техники

Кафедра МО и ПЭВМ

«Утверждаю»
Заведующий кафедрой
_____А.Ю. Козлов
«25» декабря 2023 г

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА

1. Студент Ульченко Фёдор Сергеевич гр. 20ВП1 факультета ВТ
<Фамилия, имя, отчество полностью>

2.Тема работы Мобильное программное приложение для поддержки процессов функционирования автошколы

Тема утверждена приказом ПГУ №_____от "_____"_____20_____

3. Руководитель работы Князев В.Н.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

1) Разработать приложение для мобильных устройств, обеспечивающее поддержку процессов функционирования автошколы.
2) Среда выполнения приложения: операционная система Android 9 и выше
Среда выполнения сервера: операционная система Windows 10 и выше
Языковые средства разработки: C#, UML, SQL, XAML
Инструментальные средства разработки: Microsoft SQL Server 2019, Microsoft Visual Studio 2022, Microsoft Project, Visual Paradigm
Технологии разработки: RUP, ООП, Entity Framework, MAUI
3) Основные функции мобильного приложения:
а) запись на практические занятия по вождению;
б) подготовка экзамену по теоретическим вопросам;
в) подготовка экзамену по экзаменационным билетам;
г) контроль успеваемости курсантов.

5. Перечень подлежащих разработке вопросов

1) Анализ предметной области и постановка задачи
2) Проектирование
3) Реализация
4) Контроль качества программного обеспечения
5) Оформление пояснительной записки

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области и требований к системе	10%	08.04.2024 – 13.04.2024	
2	Проектирование программных средств	20%	15.04.2024 – 25.04.2024	
3	Реализация программных средств	30%	26.04.2024 – 12.05.2024	
4	Тестирование программных средств	20%	13.05.2024 – 22.05.2024	
5	Оформление пояснительной записки	20%	23.05.2024 – 03.06.2024	

Дата выдачи задания «21» декабря 2023г.

Руководитель бакалаврской работы _____

(подпись, дата)

Задание к исполнению принял студент _____

(подпись, дата)

Работу к защите допустить

Декан ФВТ _____ Л.Р. Фионова

(подпись, дата)

Реферат

Пояснительная записка содержит 170 листов, 66 рисунков, 16 таблиц, 32 использованных источника, 6 приложений.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, АВТОШКОЛА, ANDROID, MS SQL SERVER, C#, MAUI, .NET, RUP, UML

Объектом разработки является мобильное приложение для поддержки процессов функционирования автошколы.

Цель работы – разработка мобильного программного приложения для поддержки процессов функционирования автошколы, позволяющего создавать и отменять записи на практические занятия по вождения, готовиться к теоретическому экзамену, как по отдельно взятым вопросам, так и по билетам ГИБДД целиком, а также контролировать посещаемость и успеваемость курсантов автошколы.

Разработка серверной части происходила на языке программирования C#. В качестве среды разработки использовалась Microsoft Visual Studio 2022. В качестве СУБД была использована СУБД MS SQL Server.

Разработка приложения производилась на языке программирования C#. В качестве среды разработки использовалась среда программирования Microsoft Visual Studio 2022, операционная система Windows 10. Приложение предназначено для использования под управлением ОС Android версии 9.0 и выше, а его серверная часть работает под управлением операционной системы Windows 10 и выше.

Содержание

Введение 7
1 Анализ предметной области и требований 8
1.1 Анализ предметной области и постановка задачи 8
1.2 Анализ программных продуктов аналогичного назначения 8
1.2.1 Мобильное приложение «Автошкола Реал» 8
1.2.2 Мобильное приложение «Дром ПДД» 10
1.2.3 Мобильное приложение «Автошкола-контроль» 11
1.2.4 Сравнение аналогов 12
1.3 Анализ целевой аудитории 12
1.4 Анализ функциональных требований 14
1.5 Анализ нефункциональных требований 22
1.6 Планирование разработки и оценка бюджета 22
2 Проектирование мобильного приложения 36
2.1 Выбор технологий и инструментальных средств разработки 36
2.2 Проектирование архитектуры программного обеспечения 37
2.3 Проектирование интерфейса пользователя 38
2.4 Логическое проектирование базы данных 47
2.5 Проектирование серверной части приложения 50
3 Реализация мобильного приложения 52
3.1 Реализация базы данных 52
3.2 Разработка серверной части 53
3.3 Реализация клиентской части приложения 55
4 Тестирование программного продукта 57
4.1 Тестирование серверной части приложения 57
4.2 Тестирование клиентской части приложения 58
4.2.1 Функциональное тестирование 58
4.2.2 Модульное тестирование 59
4.3 Метрики кода 60
4.3.1 Метрики размера программы 62

4.3.2 Метрики Холстеда 62
Заключение 65
Список использованных источников 66
Приложение А (справочное) Глоссарий 70
Приложение Б (рекомендуемое) Исходный код скриптов базы данных 72
Приложение В (рекомендуемое) Исходный код серверной части 77
Приложение Г (рекомендуемое) Исходный код клиентской части 106
Приложение Д (рекомендуемое) Результаты тестирования серверной части приложения 160
Приложение Е (рекомендуемое) Результаты тестирования клиентской части приложения 164

Введение

В современном мире информационные технологии приобретают всё большее распространение во всех сферах человеческой жизни. Год от года компьютеры все чаще применяют в различных частях образовательной сферы во всех уголках планеты.

Эта тенденция не обошла стороной и Российскую Федерацию. В связи с тем, что интеграция информационных технологий способна многократно повысить удобство и эффективность этих самых процессов, правительство Российской Федерации в 2017 году утвердило программу «Цифровая экономика Российской Федерации». Данная программа подразумевает «развитие цифровой экономики Российской Федерации, в которой данные в цифровой форме являются ключевым фактором производства во всех сферах экономической деятельности, что повышает конкурентоспособность страны, качество жизни граждан, обеспечивает экономический рост и экономический суверенитет» [1].

Одной из важнейших особенностей развития информационных технологий в нашей стране, да и во всем современном мире, является повсеместное распространение мобильных устройств с доступом к интернету. При этом мобильный интернет постепенно оттесняет на второй план традиционный доступ к сети с использованием стационарных компьютеров по причине своего существенно большего удобства. В частности, как отмечено в государственной программе «Цифровая экономика», в 2016 году в нашей стране «на 100 человек приходилось 159,95 мобильного телефона и из 100 человек 71,29 человека использовали мобильный доступ к сети Интернет» [2]. Со временем эта тенденция лишь усиливается.

Опираясь на приведённые выше факторы можно сделать вывод о том, что разработка мобильного приложения для автошколы весьма актуальна и перспективна в условиях современного общества.

1 Анализ предметной области и требований

1.1 Анализ предметной области и постановка задачи

Разрабатываемое мобильное приложение должно обеспечить пользователям максимально удобный и интуитивно понятный интерфейс.

Таким образом, в приложении должны быть представлены следующие основные функции:

- регистрация новых пользователей;
- вход зарегистрированных пользователей в систему;
- запись на занятие по вождению;
- подготовка к экзамену по теории ПДД;
- контроль за успеваемостью;
- контроль за посещаемостью.

Помимо рядовых пользователей в системе будут присутствовать и администраторы, которые будут иметь более широкие возможности действий. В частности, они должны быть способны добавлять в систему новых пользователей вручную, например: уволившихся инструкторов или курсантов окончивших обучение.

1.2 Анализ программных продуктов аналогичного назначения

В настоящее время существует целый ряд приложений, использующихся при обучении вождению. Рассмотрим наиболее известные мобильные приложения такого рода и произведём для них сравнительный анализ.

1.2.1 Мобильное приложение «Автошкола Реал»

Автошкола Реал сеть современных автошкол в Москве и Московской области, обладающая своим собственным мобильным приложением [3]. Пройдя авторизацию пользователь попадает на главную страницу приложения, с которой есть доступ ко всему функционалу: изучение теории и запись на

занятие по вождению. Навигация в приложении устроена с помощью нижней панели, в которой размещены вкладки приложения.

Главная страница мобильного приложения «Автошкола Реал» представлена на рисунке 1.

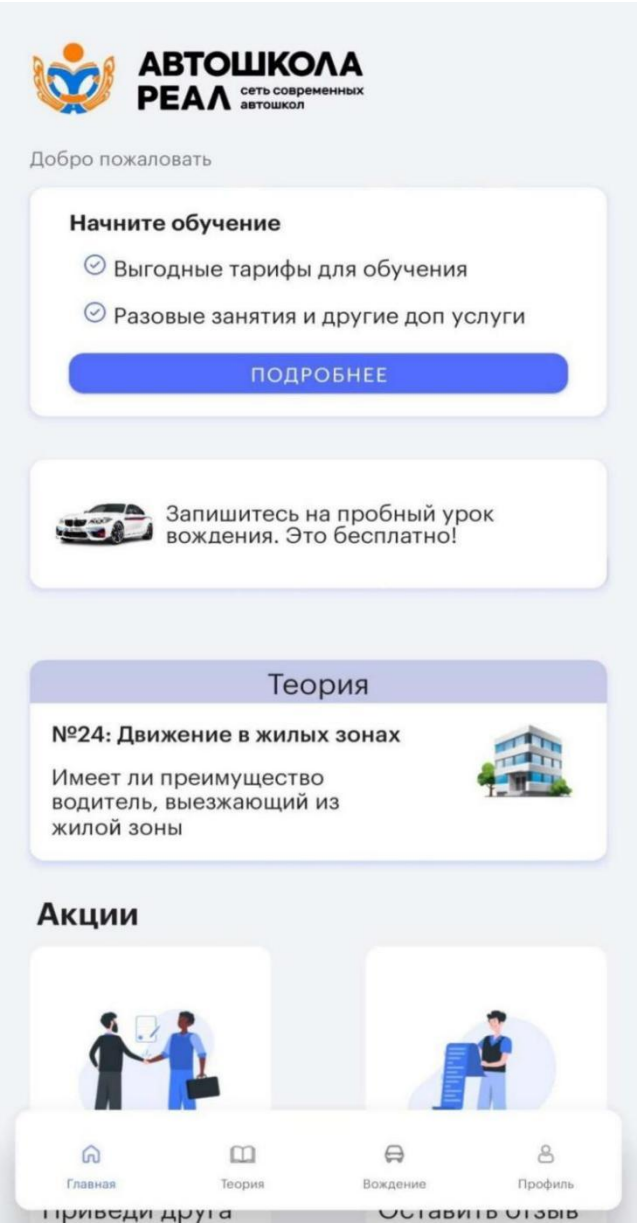


Рисунок 1 – Главная страница мобильного приложения «Автошкола Реал»

К достоинствам приложения относится возможность записи на практическое занятие и возможность подготовки по теории ПДД.

К недостаткам мобильного приложения относится перегруженность интерфейса, отсутствие поддержки тёмной темы, плохая оптимизация: постоянные зависания и вылеты приложения.

1.2.2 Мобильное приложение «Дром ПДД»

Дром ПДД – это автомобильный интернет-портал, сочетающий многогранную и обширную базу информации по автомобилям с современными технологиями общения. Дром ПДД – приложение от Дрома, предоставляющее функционал подготовки к теоретическому экзамену. Экзаменационные билеты взяты с официального сайта ГИБДД МВД РФ. Не является официальным приложением ГИБДД МВД РФ [4].

Главная страница мобильного приложения «Дром ПДД» представлена на рисунке 2.

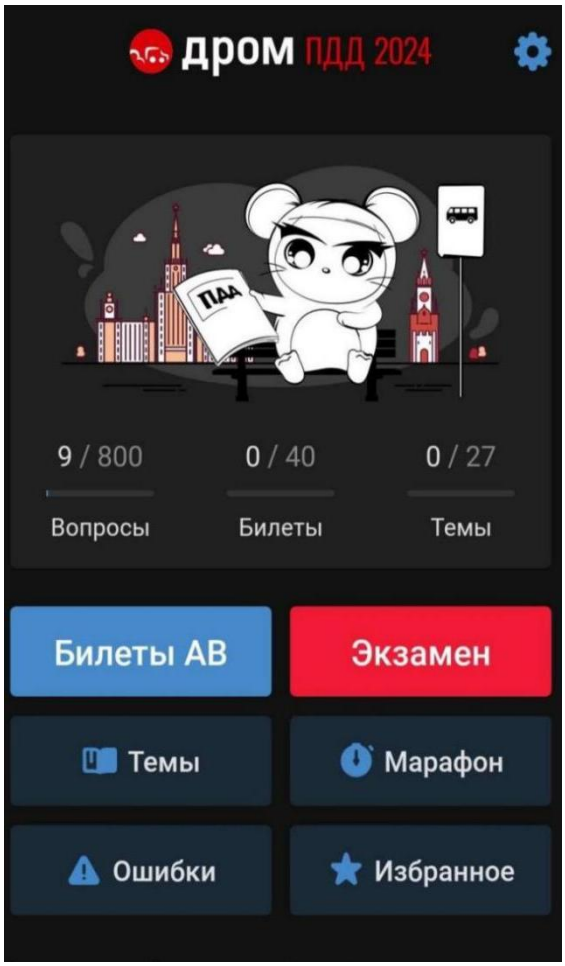


Рисунок 2 – Главная страница мобильного приложения «Дром ПДД»

К достоинствам приложения относится возможность подготовки по теории ПДД, хорошая оптимизация и поддержка тёмной темы.

К недостаткам мобильного приложения относится отсутствие возможности записи на занятие по вождению.

1.2.3 Мобильное приложение «Автошкола-Контроль»

Сервис «Автошкола-Контроль» был запущен в январе 2016 года в Республике Татарстан. Стартовой площадкой послужило наличие в регионе профессионального сообщества автошкол – Федерации Автошкол Республики Татарстан [5]. Пройдя авторизацию пользователь попадает на главную страницу приложения, с которой есть доступ ко всему функционалу: изучение теории и запись на занятие по вождению. Навигация в приложении устроена с помощью нижней панели, в которой размещены вкладки приложения.

Главная страница мобильного приложения «Автошкола-Контроль» представлена на рисунке 3.

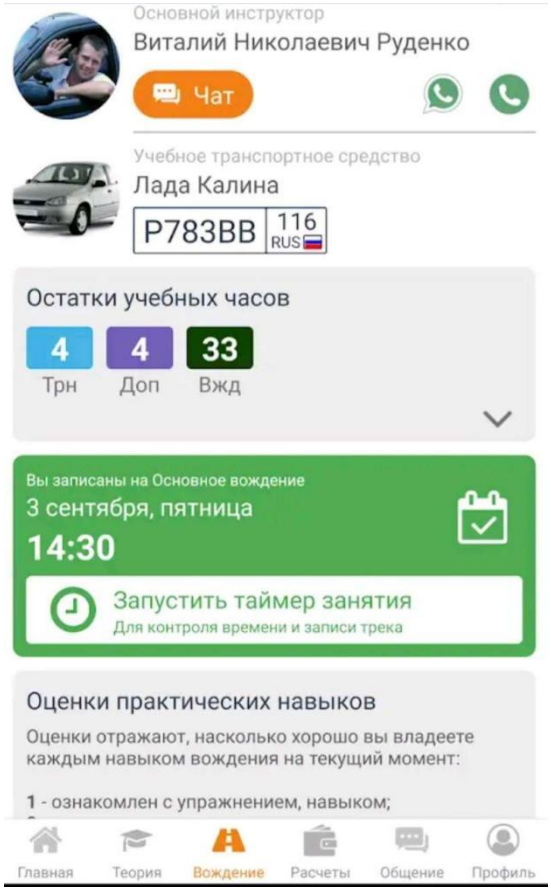


Рисунок 3 – Главная страница мобильного приложения «Автошкола-Контроль»

К достоинствам приложения относится возможность записи на практическое занятие и возможность подготовки по теории ПДД.

К недостаткам мобильного приложения относится перегруженность интерфейса, отсутствие поддержки тёмной темы.

1.2.4 Сравнение аналогов

Сравнительный анализ рассмотренных выше мобильных приложений представлен в таблице 1.

Таблица 1 – Сравнение аналогов

Мобильное приложение	Запись на занятие	Подготовка к экзамену	Удобство интерфейса	Оптимизация	Поддержка тёмной темы
Автошкола Реал	Да	Да	Перегруженный	Плохая	Нет
Дром ПДД	Нет	Да	Понятный	Хорошая	Да
Автошкола-контроль	Да	Да	Перегруженный	Хорошая	Нет
Разрабатываемое приложение	Да	Да	Понятный	Хорошая	Да

Таким образом, в результате анализа систем аналогичного назначения было принято решение о разработке собственного приложения, которое будет лишено недостатков, выявленных у рассмотренных выше мобильных приложений.

1.3 Анализ целевой аудитории

Целевой аудиторией проекта являются работники и учащиеся автошкол. Согласно годовой отчётности ГИБДД, которая содержит информацию также о водительских удостоверениях – в прошлом году было выдано 1,37 млн прав – немногим больше, чем в 2022 году (1,35 млн). Также выросло общее количество автошкол в 2023 году с 7,2 тыс. до 7,54 тыс. [6].

На рисунке 4 представлены доли основных операционных систем смартфонов на мировом рынке за последний год [7].

Рисунок 4 – Доли операционных систем смартфонов в мировом рынке

Можно заметить, что подавляющее большинство смартфонов на мировом рынке используют операционную систему Android. Единственное значимое исключение – компания Apple, использующая собственную систему iOS. Но,

при всей популярности Apple, ее продукция даже в периоды пикового спроса составляет лишь в районе 20% от мирового рынка, что в несколько раз меньше, чем доля устройств с Android. В РФ доля смартфонов Apple ниже, чем в среднем по планете [7]. Поэтому наиболее выгодным решением с точки зрения доступности продукта для пользователей будет реализация разрабатываемого приложения именно для операционной системы Android.

Статистика распространения разных версий системы Android представлена на рисунке 5 [8].

Рисунок 5 – Распространение различных версий системы Android

Из приведённой диаграммы можно увидеть, что 86,1 % пользователей системы Android использует версии не старее 9.0. Поэтому приложение будет разрабатываться таким образом, чтобы оно успешно функционировало для Android 9.0 и более новых версий системы.

1.4 Анализ функциональных требований

Приложение может применяться инструкторами и курсантами различных автошкол. Приложение позволяет создавать и отменять записи на практические занятия, вести контроль за успеваемостью и посещаемостью курсантов. Так же курсант может изучать актуальные правила дорожного движения путём тестирования по теоретическим вопросам и билетам ПДД.

Обучение в автошколе подразумевает два вида занятий: теоретические и практические.

Изучение теории курсант осуществляет самостоятельно, после чего может закрепить полученные знания в соответствующем разделе своего приложения.

Практические занятия проводит инструктор и требуют предварительной записи на конкретную дату и время. Осуществить запись могут как инструктор, так и курсант. При необходимости данная запись может быть отменена.

Допуск к экзаменам в ГИБДД курсант получает только после полного прохождения теоретического и практического курсов. Для того чтобы теоретический курс считался пройденным курсант должен успешно пройти тестирование по всем 40 билетам ПДД. Согласно приказу Министерства просвещения Российской Федерации от 8 ноября 2021 г. N 808. «Об утверждении примерных программ профессионального обучения водителей транспортных средств соответствующих категорий и подкатегорий» от курсанта требуется прохождение не менее 56 часов вождения для допуска к практическому экзамену [9].

Отслеживать успеваемость курсанта в подготовке по экзаменационным билетам ПДД, а также количество часов, посещённых практических занятий по вождению, имеют возможность как курсант, так и его инструктор, в соответствующем разделе мобильного приложения.

В результате анализа требований задания была создана диаграмма вариантов использования (рисунок 6).

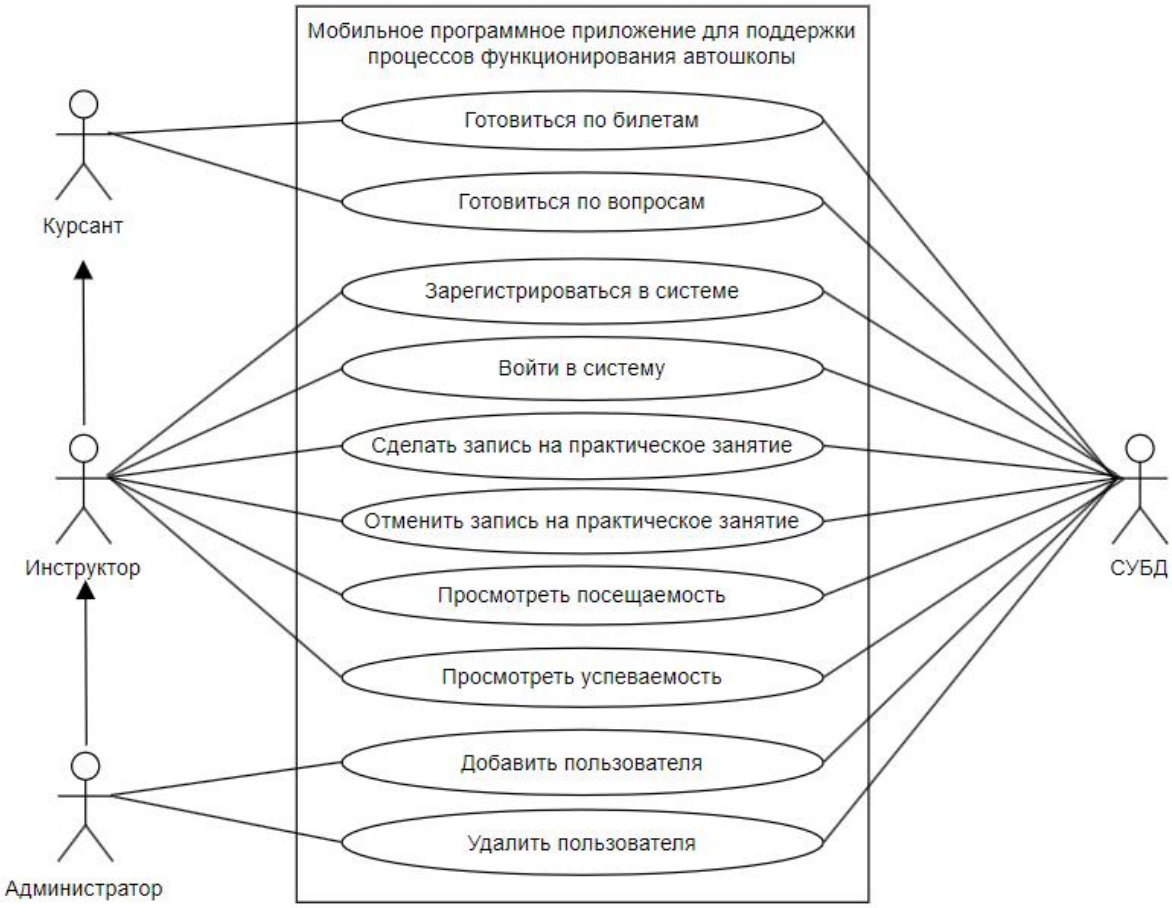


Рисунок 6 – Диаграмма вариантов использования мобильного приложения

Составим спецификацию прецедента «Сделать запись на практическое занятие» (рисунок 7).

Название: Сделать запись на практическое занятие
Назначение: резервация времени для практического занятия
Актеры: инструктор/курсант, СУБД
Предусловие: время свободно для записи
Основной сценарий: 1.Курсант или инструктор выбирает дату и время для записи. 2.Система запрашивает у СУБД сведения о том, свободно ли выбранное время. 3.СУБД передает сведения о наличии аккаунта с такими логином и паролем 4. Если выбранное время свободно 4.1. Система создает запись для ученика и инструктора на заданное время. 5.Иначе система выводит сообщение о том, что время занято.
Постусловие: запись создана

Рисунок 7 – Спецификация варианта использования «Сделать запись на практическое занятие»

Составим спецификацию прецедента «Зарегистрироваться в системе» (рисунок 8).

Наименование: Зарегистрироваться в системе
Назначение: регистрация нового аккаунта пользователя
Действующие лица: курсант, инструктор, СУБД
Предусловие: приложение запущено
Основной сценарий: 1. Пользователь выбирает регистрацию 2. Пользователь вводит логин и пароль. 3. Система проверяет логин и пароль на корректность. 4. Если логин и пароль корректны. 4.1.Система запрашивает у СУБД сведения о наличии аккаунта с введенным логином. 4.2. СУБД передает сведения о наличии аккаунта с введенным логином. 4.3. Если такого аккаунта нет 4.3.1. СУБД добавляет в БД информацию о новом пользователе. 4.3.2. Система осуществляет вход нового пользователя. 4.4. Иначе система выводит сообщение о том, что выбранный логин занят. 5. Иначе система выводит сообщение о недопустимом логине или пароле.
Постусловие: регистрация выполнена

Рисунок 8 – Спецификация варианта использования «Зарегистрироваться в системе»

Составим спецификацию прецедента «Войти в систему» (рисунок 9).

Название: войти в систему
Назначение: вход в систему с существующего аккаунта пользователя
Актеры: пользователь, СУБД
Предусловие: приложение запущено
Основной сценарий: 1.Пользователь выбирает авторизацию 2.Пользователь вводит логин и пароль. 3.Система запрашивает у СУБД сведения о наличии аккаунта с введенным логином и паролем. 4.СУБД передает сведения о наличии аккаунта с такими логином и паролем 5. Если такой аккаунт существует 5.1. Система проверяет наличие у него активной блокировки. 5.2. Если активной блокировки нет 5.2.1. Система осуществляет вход нового пользователя. 5.3. Иначе система сообщает о наличии у пользователя блокировки. 6.Иначе система выводит сообщение о неверном логине или пароле.
Постусловие: вход в систему выполнен

Рисунок 9 – Спецификация варианта использования «Войти в систему»

Составим спецификацию прецедента «Отменить запись на практическое занятие» (рисунок 10).

Название: Отменить запись на практическое занятие
Назначение: отмена резервации времени для практического занятия
Актеры: инструктор/курсант, СУБД
Предусловие: запись на занятие существует
Основной сценарий: 1.Курсант или инструктор выбирает дату и время отменяемой записи. 2.Система передает СУБД сведения о том, какую запись необходимо удалить. 3.СУБД удаляет сведения о записи.
Постусловие: запись удалена

Рисунок 10 – Спецификация варианта использования «Отменить запись на практическое занятие»

Составим спецификацию прецедента «Просмотреть посещаемость» (рисунок 11).

Название: Просмотреть успеваемость
Назначение: контроль успеваемости курсанта
Актеры: инструктор/курсант, СУБД
Предусловие: курсант зарегистрирован
Основной сценарий: 1. Курсант переходит на страницу с теорией. 2. Система запрашивает у СУБД сведения о том, на какие вопросы курсант дал верный ответ. 3. Система выводит соотношение верных ответов к неверным на экран. Альтернативный сценарий: 1. Инструктор переходит на страницу с курсантами. 2. Система запрашивает у СУБД сведения о том, на какие вопросы каждый из курсантов, закреплённых за данным инструктором, дал верный ответ. 3. Система выводит полученные дынные на экран.
Постусловие: данные выведены на экран

Рисунок 11 – Спецификация варианта использования «Просмотреть успеваемость»

Составим спецификацию прецедента «Просмотреть успеваемость» (рисунок 12).

Название: просмотреть посещаемость
Назначение: контроль посещаемости курсанта
Актеры: инструктор/курсант, СУБД
Предусловие: курсант вошел в систему
Основной сценарий: 1. Курсант переходит на страницу записи на занятие. 2. Система запрашивает у СУБД сведения о том, сколько раз курсант был на занятиях по вождению. 3. Система выводит полученные дынные на экран. Альтернативный сценарий: 1. Инструктор переходит на страницу с курсантами. 2. Система запрашивает у СУБД сведения о том, сколько занятий по вождению посетил каждый из курсантов, закреплённых за данным инструктором. 3. Система выводит полученные дынные на экран.
Постусловие: данные выведены на экран

Рисунок 12 – Спецификация варианта использования «Просмотреть посещаемость»

Составим спецификацию прецедента «Готовиться по вопросам» (рисунок 13).

Название: готовиться по вопросам
Назначение: изучение вопросов к экзамену ГИБДД
Актеры: курсант, СУБД
Предусловие: курсант вошел в систему
Основной сценарий: 1. Курсант переходит на страницу с вопросами по теории ПДД. 2. Курсант выбирает тему вопросов. 3. Система запрашивает у СУБД сведения о вопросах, относящихся к указанной теме. 4. СУБД возвращает системе список вопросов, относящихся к указанной теме. 5. Система поэтапно выводит на экран вопросы, состоящие из формулировки, одного правильного и трех неправильных ответов. 6. Курсант выбирает один из вариантов ответа. 7. Ответ курсанта фиксируется в СУБД. 8. Система выводит следующий вопрос.
Постусловие: ответы курсанта зафиксированы в СУБД

Рисунок 13 – Спецификация варианта использования «Готовиться по вопросам»

Составим спецификацию прецедента «Готовиться по билетам» (рисунок 14).

Название: готовиться по билетам
Назначение: изучение билетов к экзамену ГИБДД
Актеры: курсант, СУБД
Предусловие: курсант вошел в систему
Основной сценарий: 1. Курсант переходит на страницу с билетами по теории ПДД. 2. Курсант выбирает номер билета. 3. Система запрашивает у СУБД сведения о вопросах, относящихся к указанному билету. 4. СУБД возвращает системе список вопросов, относящихся к указанному билету. 5. Система поэтапно выводит на экран вопросы из билета, состоящие из формулировки, одного правильного и трех неправильных ответов. 6. Курсант выбирает один из вариантов ответа. 7. Система выводит следующий вопрос. 8. Когда курсант дал ответ на все вопросы в билете, результат фиксируется в СУБД. 9. Система возвращает курсанта на страницу с билетами по теории ПДД.
Постусловие: ответы курсанта зафиксированы в СУБД

Рисунок 14 – Спецификация варианта использования «Готовиться по билетам»

Составим спецификацию прецедента «Добавить пользователя» (рисунок 15)

Название: Добавить пользователя
Назначение: добавление нового пользователя в систему
Актеры: администратор, СУБД
Предусловие: данный пользователь отсутствует в базе данных
Основной сценарий: 1. Инструктор заполняет поля: фамилия, имя, отчество, роль. 2. Система передает данные в СУБД. 3. СУБД добавляет нового пользователя, если пользователь с указанными параметрами отсутствует в базе данных. 4. Иначе система выводит сообщение о том, что такой пользователь уже есть.
Постусловие: пользователь создан

Рисунок 15 – Спецификация варианта использования «Добавить пользователя»

Составим спецификацию для прецедента «Удалить пользователя» (рисунок 16).

Название: удалить пользователя
Назначение: удаление пользователя из системы
Актеры: администратор, СУБД
Предусловие: данный пользователь есть в базе данных
Основной сценарий: 1. Инструктор заполняет поля: фамилия, имя, отчество, роль. 2. Система передает данные в СУБД. 3. СУБД добавляет нового пользователя, если пользователь с указанными параметрами отсутствует в базе данных. 4. Иначе система выводит сообщение о том, что такой пользователь уже есть.
Постусловие: пользователь создан

Рисунок 16 – Спецификация варианта использования «Удалить пользователя»

Для некоторых из рассмотренных выше вариантов использования было принято решение разработать диаграммы последовательности и деятельности, с целью рассмотреть данные варианты использования подробнее.

Для варианта использования «Готовиться по вопросам» была составлена диаграмма последовательности, представленная на рисунке 17.

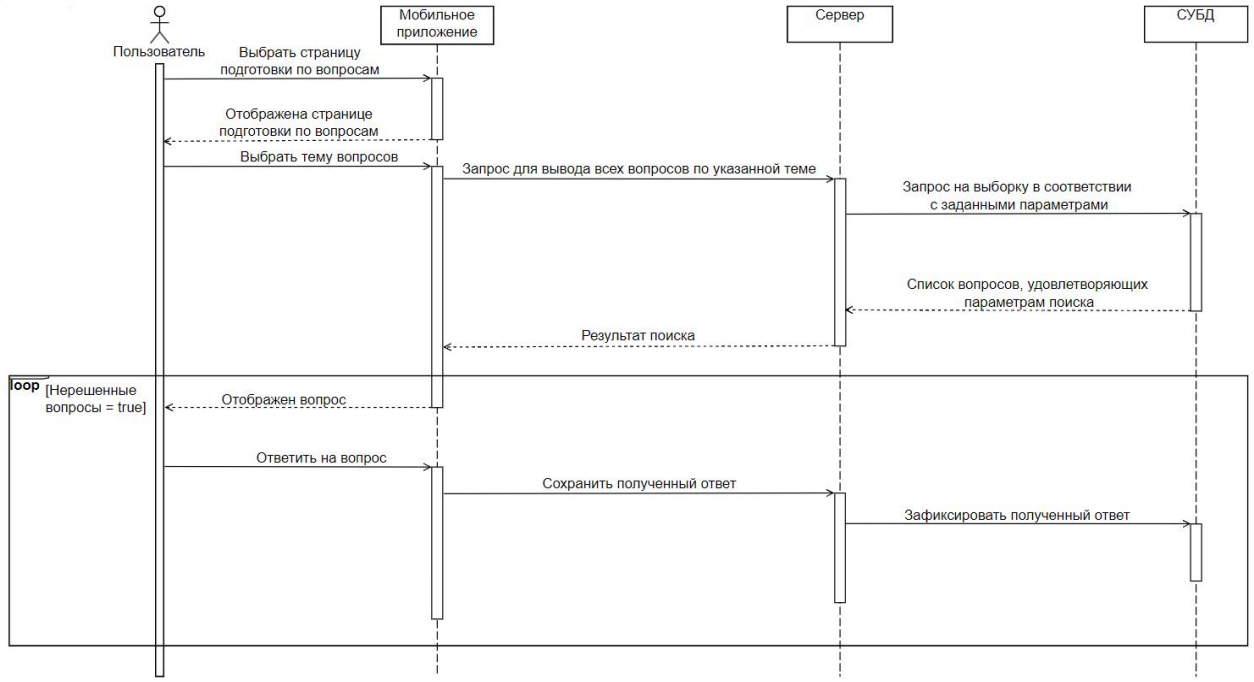


Рисунок 17 – Диаграмма последовательности варианта использования «Готовиться по вопросам»

Для варианта использования «Зарегистрироваться» составлена диаграмма деятельности, представленная на рисунке 18.

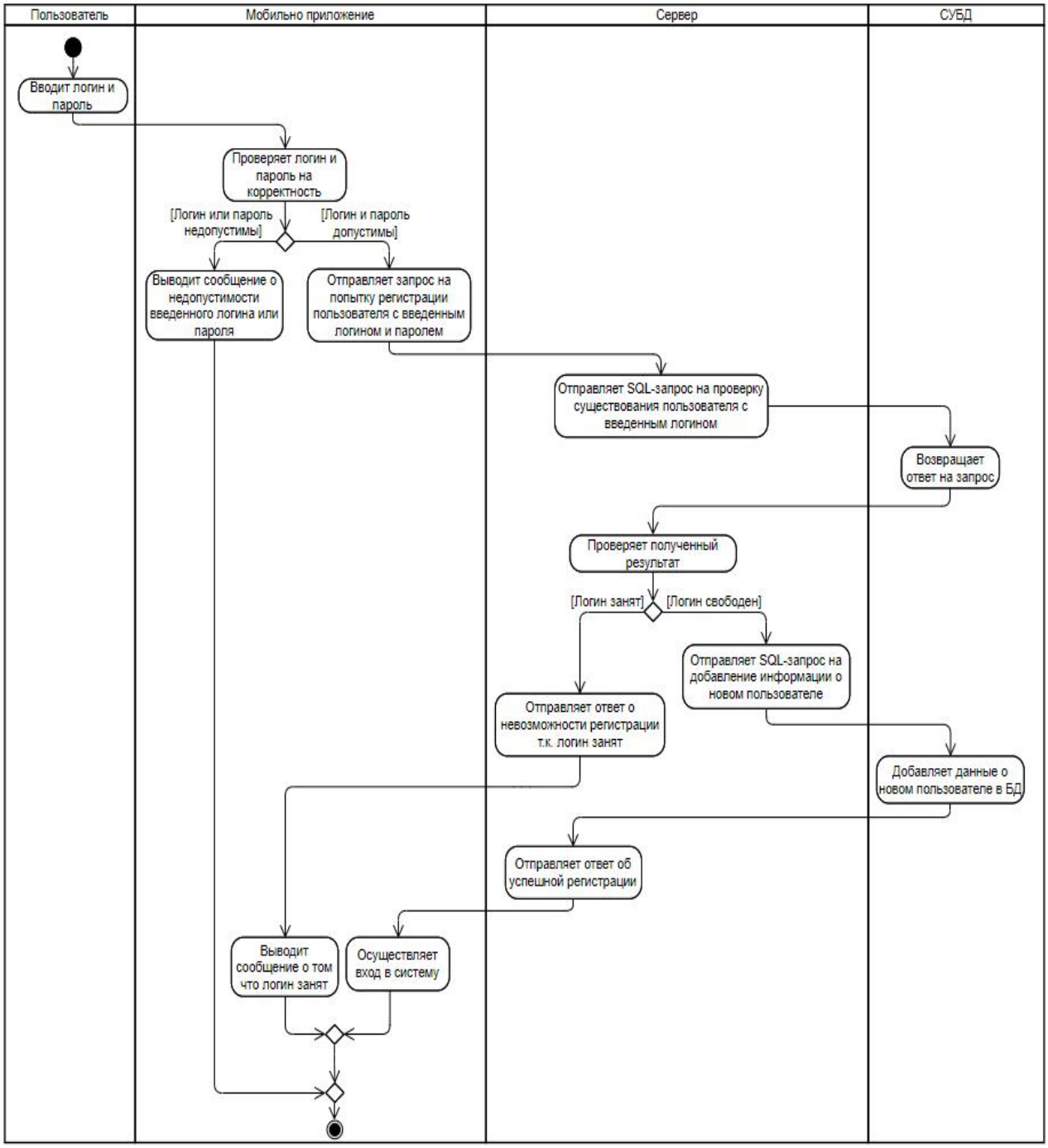


Рисунок 18 – Диаграмма деятельности варианта использования «Зарегистрироваться»

В результате анализа функциональных требований были успешно выявлены основные функции разрабатываемого приложения. При этом для разрабатываемого приложения были построены диаграммы вариантов

использования, деятельности и последовательности, а также для некоторых выявленных вариантов использования были составлены спецификации.

1.5 Анализ нефункциональных требований

Разработка серверной части приложения должна быть выполнена на широкораспространённом языке программирования C#, в среде программирования Microsoft Visual Studio 2022. Данные на сервере хранятся в базе данных, реализованной с использованием СУБД Microsoft SQL Server, запросы к базе данных осуществляются на языке SQL. Серверная должна работать под управлением операционной системы Windows 10 и выше.

Клиентская часть приложения должна быть реализована на языке программирования C# с использованием современного и эффективного кроссплатформенного фреймворка .NET MAUI, в среде программирования Microsoft Visual Studio 2022. Она должна представлять собой мобильное приложение с удобным и интуитивно понятным пользовательским интерфейсом. Клиентское приложение должно быть отлажено и должно успешно работать под управлением ОС Android 9.0 и выше.

1.6 Планирование разработки и оценка бюджета

Качественное выполнение работ по любому проекту, в том числе и в сфере информационных технологий, невозможно при отсутствии грамотного планирования, которое позволяет обеспечить выполнение работ над проектом таким образом, чтобы добиться выполнения всех поставленных целей за минимально возможное время, с минимально возможными затратами и с приемлемым качеством [10 – 15].

Задачи, представляющие собой этапы работы над проектом, представлены в таблице 2.

Таблица 2 – Перечень задач проекта

№	Название задачи	Длительность	Предшественники
1	Начало проекта	–	–
2	Анализ требований	–	–

Продолжение таблицы 2

№	Название задачи	Длительность	Предшественники
3	Анализ предметной области и постановка задачи	2 дня	1
4	Анализ программных продуктов аналогичного назначения	4 дня	1
5	Анализ функциональных требований	1 день	3; 4
6	Анализ требований к интерфейсу	1 день	5
7	Анализ целевой аудитории	2 дня	6
8	Анализ требований завершён	–	7
9	Выбор технологий разработки ПО	–	–
10	Анализ существующих технологий разработки ПО	2 дня	1
11	Выбор технологий разработки ПО	1 день	10
12	Выбор средств разработки ПО завершён	–	11
13	Проектирование программных средств	–	–
14	Проектирование базы данных	6 дней	8; 12
15	Проектирование архитектуры приложения	4 дня	14
16	Проектирование интерфейса пользователя	5 дней	8; 12
17	Проектирование приложения завершено	–	16; 15
18	Реализация программных средств	–	–
19	Реализация базы данных	8 дней	14; 15
20	Реализация серверной части приложения	21 день	17; 19
21	Реализация мобильного приложения	14 дней	17; 19
22	Реализация завершена	–	20; 21
23	Тестирование	–	–
24	Функциональное тестирование	3 дня	22
25	Модульное тестирование	3 дня	22
26	Тестирование завершено	–	24; 25
27	Составление и оформление документации	7 дней	26
28	Конец проекта	–	27

Таким образом, для реализации проекта нужно выполнить 28 задач.

В таблице 3 представлено описание затрачиваемых ресурсов.

Таблица 3 – Список ресурсов проекта

Название ресурса	Тип	Стандартная ставка	Ставка сверхурочных
Программист1	Трудовой	350 Р/ч	450 Р/ч
Программист2	Трудовой	350 Р/ч	450 Р/ч
Программист3	Трудовой	550 Р/ч	750 Р/ч
Дизайнер	Трудовой	250 Р/ч	350 Р/ч
Интернет	Затраты	2500 Р/месяц	
Компьютер	Материальный	500 Р	
Мобильное устройство	Материальный	300 Р	

Таким образом, в общей сложности, потребуется семь видов ресурсов.

Распределение ресурсов по задачам представлено в таблице 4.

Таблица 4 – Назначение ресурсов

№	Название задачи	Ресурс	Единицы
1	Начало проекта	–	–
2	Анализ требований	–	–
3	Анализ предметной области и постановка задачи	Программист 3 Компьютер Интернет	50% 1 шт.
4	Анализ программных продуктов аналогичного назначения	Программист 1 Программист 2 Компьютер Интернет Мобильное устройство	50% 50% 2 шт. 2 шт.
5	Анализ функциональных требований	Программист 3 Программист 1 Компьютер Интернет	50% 50% 2 шт.
6	Анализ требований к интерфейсу	Программист 3 Программист 2 Дизайнер Компьютер Интернет	50% 50% 100% 3 шт.
7	Анализ целевой аудитории	Программист 1 Компьютер Интернет	50% 3 шт.
8	Анализ требований завершён	–	–
9	Выбор технологий разработки ПО	–	–
10	Анализ существующих технологий разработки ПО	Программист 3 Компьютер Интернет	25% 1 шт.
11	Выбор технологий разработки ПО	Программист 3 Программист 2 Программист 1 Компьютер Интернет	50% 50% 50% 3 шт.
12	Выбор средств разработки ПО завершён	–	–
13	Проектирование программных средств	–	–
14	Проектирование базы данных	Программист 3 Программист 1 Компьютер Интернет	50% 50% 2 шт.
15	Проектирование архитектуры приложения	Программист 3 Программист 1 Компьютер Интернет	50% 50% 2 шт.

Продолжение таблицы 4

№	Название задачи	Ресурс	Единицы
16	Проектирование интерфейса пользователя	Программист 3 Программист 2 Дизайнер Компьютер Интернет	50% 50% 100% 3 шт.
17	Проектирование приложения завершено	–	–
18	Реализация программных средств	–	–
19	Реализация базы данных	Программист 1 Компьютер Интернет	100% 1 шт.
20	Реализация серверной части приложения	Программист 3 Программист 1 Компьютер Интернет	50% 100% 2 шт.
21	Реализация мобильного приложения	Программист 3 Программист 2 Дизайнер Компьютер Интернет	50% 100% 100% 3 шт.
22	Реализация завершена	–	–
23	Тестирование	–	–
24	Функциональное тестирование	Программист1 Программист2 Компьютер Интернет Мобильное устройство	50% 50% 2 шт. 2 шт.
25	Модульное тестирование	Программист1 Программист2 Компьютер Интернет Мобильное устройство	50% 50% 2 шт. 2 шт.
26	Тестирование завершено	–	–
27	Составление и оформление документации	Программист 3 Программист 2 Программист 1 Компьютер Интернет	100% 100% 100% 3 шт.
28	Конец проекта	–	–

Таким образом, реализацией проекта будут заниматься три программиста и один дизайнер, которым для работы нужны компьютеры, мобильные устройства и доступ к интернету.

Для вычисления длительности реализации проекта и его критического пути построен сетевой график (рисунок 18).

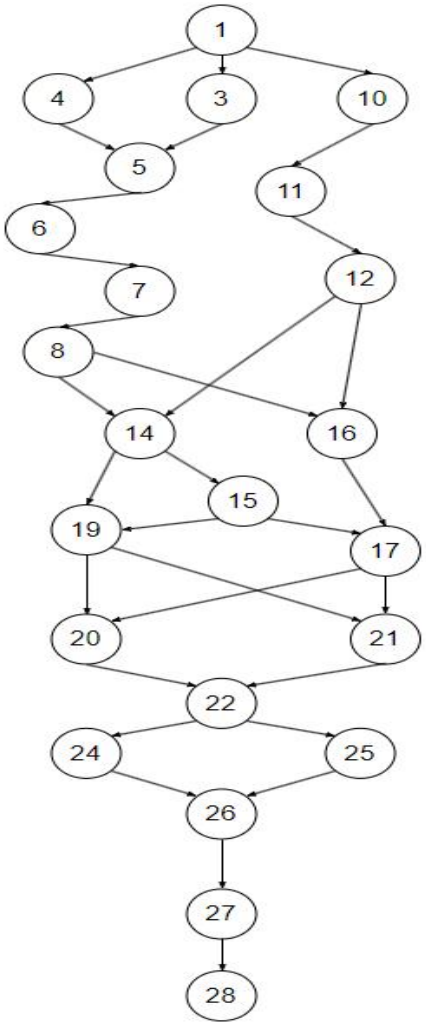


Рисунок 18 – Сетевой график проекта

Рассчитаем вручную критический путь проекта. В таблице 5 представлены расчёты раннего времени начала работ для его задач.

Таблица 5 – Расчёты раннего времени начала работ

Работа	Предшественники			Вычисление	Раннее время начала
	№	Раннее время начала	Длительность		
1	–	–	–	–	0
3	1	0	0	0+0	0
4	1	0	0	0+0	0
5	3 4	0 0	2 4	Max(0+2;0+4)	4
6	5	4	1	4+1	5
7	6	5	1	5+1	6
8	7	6	2	6+2	8

Продолжение таблицы 5

Работа	Предшественники			Вычисление	Раннее время начала
	№	Раннее время начала	Длительность		
10	1	0	0	0+0	0
11	10	0	2	0+2	2
12	11	2	1	2+1	3
14	8 12	8 3	0 0	Max(8+0;3+0)	8
15	14	8	6	8+6	14
16	8 12	8 3	0 0	Max(8+0;3+0)	8
17	16 15	8 14	5 4	Max(8+5;14+4)	18
19	14 15	8 14	6 4	Max(8+6;14+4)	18
20	17 19	18 18	0 8	Max(18+0;18+8)	26
21	17 19	18 18	0 8	Max(18+0;18+8)	26
22	20 21	26 26	21 14	Max(26+21;26+14)	47
24	22	47	0	47+0	47
25	22	47	0	47+0	47
26	24 25	47 47	3 3	Max(47+3;47+3)	50
27	26	50	0	50+0	50
28	27	50	7	50+7	57

В таблице 6 рассчитано позднее время начала работ задач.

Таблица 6 – Расчёты позднего времени начала работ

Работа	Последователи			Вычисление	Позднее время начала
	№	Позднее время начала	Длительность		
28	–	–	–	–	57
27	28	57	0	57-0	57
26	27	57	7	57-7	50
25	26	50	0	50-0	50
24	26	50	0	50-0	50
22	24 25	50 50	3 3	Min(50-3; 50-3)	47
21	22	47	0	47-0	47
20	22	47	0	47-0	47
19	20 21	47 47	21 14	Min(47-21;47-14)	26
17	20 21	47 47	21 14	Min(47-21;47-14)	26
16	17	26	0	26-0	26

Продолжение таблицы 6

Работа	Последователи			Вычисление	Позднее время начала
	№	Позднее время начала	Длительность		
15	17 19	26 26	0 8	Min(26-0;26-8)	18
14	15 19	18 26	4 8	Min(18-4;26-8)	14
12	14 16	14 26	6 5	Min(14-6;26-5)	8
11	12	8	0	8-0	8
10	11	8	1	8-1	7
8	14 16	14 26	6 5	Min(14-6;26-5)	8
7	8	8	0	8-0	8
6	7	8	2	8-2	6
5	6	6	1	6-1	5
4	5	5	1	5-1	4
3	5	5	1	5-1	4
1	3 4 10	4 4 7	2 4 2	Min(4-2;4-4;7-2)	0

На основе найденных значений раннего и позднего времени начала работ можно найти критический путь. Соответствующие вычисления представлены в таблице 7.

Таблица 7 – Вычисление критического пути

Работа №	Раннее время начала	Позднее время начала	Резерв времени
1	0	0	0
3	0	4	4
4	0	4	4
5	4	5	1
6	5	6	1
7	6	8	2
8	8	8	0
10	0	7	7
11	2	8	6
12	3	8	5
14	8	14	6
15	14	18	4
16	8	26	18
17	18	26	8
19	18	26	8
20	26	47	21
21	26	47	21

Продолжение таблицы 7

Работа №	Раннее время начала	Позднее время начала	Резерв времени
22	47	47	0
24	47	50	3
25	47	50	3
26	50	50	0
27	50	57	7
28	57	57	0

На основе полученных можно сделать вывод, что критический путь проходит через работы с номерами 1, 3, 5, 6, 7, 8, 14, 15, 19, 20, 22, 24, 26, 27, 28, а общее время выполнения проекта составляет 57 дней.

На рисунке 19 представлен граф проекта, на котором штриховыми линиями указан критический путь.

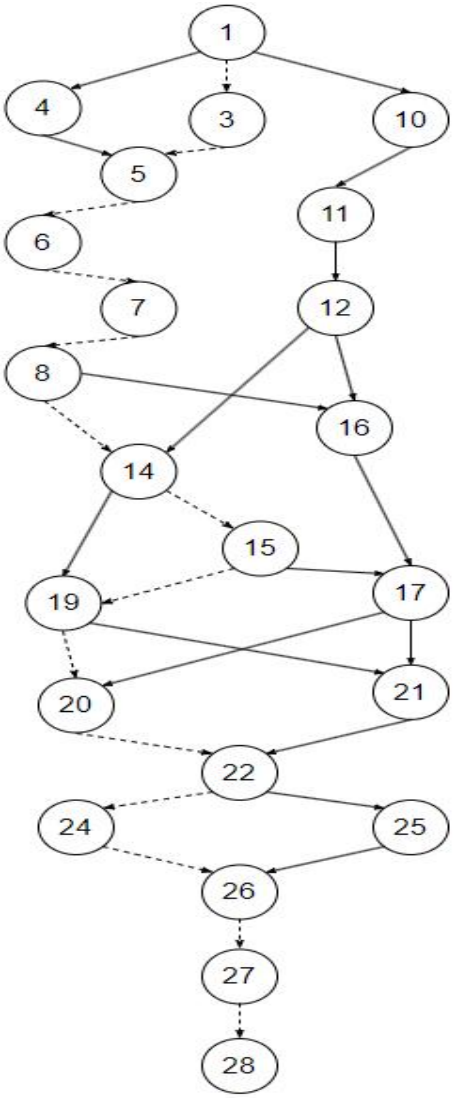


Рисунок 19 – Критический путь проекта

Для того, чтобы проверить корректность расчётов продолжительности проекта, а также вычислить величину его бюджета, воспользуемся системой управления проектами Microsoft Project, которая позволяет автоматически произвести все необходимые вычисления [10 – 15].

Список задач из таблицы 3, внесённый в систему Microsoft Project, представлен на рисунке 20. В качестве времени начала проекта была принята дата 26.02.2024.

	Название задачи	Длительность	Начало	Окончание	Предшественники
1	Начало проекта	0 дней	Пн 26.02.24	Пн 26.02.24	
2	**Анализ требований**	**8 дней**	**Пн 26.02.24**	**Ср 06.03.24**	
3	Анализ предметной области и постановка задачи	2 дней	Пн 26.02.24	Вт 27.02.24	1
4	Анализ программных продуктов аналогичного назначения	4 дней	Пн 26.02.24	Чт 29.02.24	1
5	Анализ функциональных требований	1 день	Пт 01.03.24	Пт 01.03.24	3;4
6	Анализ требований к интерфейсу	1 день	Пн 04.03.24	Пн 04.03.24	5
7	Анализ целевой аудитории	2 дней	Вт 05.03.24	Ср 06.03.24	6
8	Анализ требований завершен	0 дней	Ср 06.03.24	Ср 06.03.24	7
9	**Выбор технологий разработки ПО**	**3 дней**	**Пн 26.02.24**	**Ср 28.02.24**	
10	Анализ существующих технологий разработки ПО	2 дней	Пн 26.02.24	Вт 27.02.24	1
11	Выбор технологий разработки ПО	1 день	Ср 28.02.24	Ср 28.02.24	10
12	Выбор средств разработки ПО завершен	0 дней	Ср 28.02.24	Ср 28.02.24	11
13	**Проектирование программных средств**	**10 дней**	**Чт 07.03.24**	**Ср 20.03.24**	
14	Проектирование базы данных	6 дней	Чт 07.03.24	Чт 14.03.24	8;12
15	Проектирование архитектуры приложения	4 дней	Пт 15.03.24	Ср 20.03.24	14
16	Проектирование интерфейса пользователя	5 дней	Чт 07.03.24	Ср 13.03.24	8;12
17	Проектирование приложения завершено	0 дней	Ср 20.03.24	Ср 20.03.24	16;15
18	**Реализация программных средств**	**29 дней**	**Чт 21.03.24**	**Вт 30.04.24**	
19	Реализация базы данных	8 дней	Чт 21.03.24	Пн 01.04.24	14;15
20	Реализация серверной части приложения	21 дней	Вт 02.04.24	Вт 30.04.24	17;19
21	Реализация мобильного приложения	14 дней	Вт 02.04.24	Пт 19.04.24	17;19
22	Реализация завершена	0 дней	Вт 30.04.24	Вт 30.04.24	20;21
23	**Тестирование**	**3 дней**	**Ср 01.05.24**	**Пт 03.05.24**	
24	Функциональное тестирование	3 дней	Ср 01.05.24	Пт 03.05.24	22
25	Модульное тестирование	3 дней	Ср 01.05.24	Пт 03.05.24	22
26	Тестирование завершено	0 дней	Пт 03.05.24	Пт 03.05.24	24;25
27	Составление и оформление документации	7 дней	Пн 06.05.24	Вт 14.05.24	26
28	Конец проекта	0 дней	Вт 14.05.24	Вт 14.05.24	27

Рисунок 20 – Задачи проекта в среде Microsoft Project

В среде Microsoft Project на основе введённых данных была построена диаграмма Ганта. На её основании можно понять, что критический путь проходит через задачи 1, 3, 5, 6, 7, 8, 14, 15, 19, 20, 22, 24, 26, 27, 28. Можно

заметить, что вычисленный программой критический путь совпал с критическим путём, вычисленным вручную.

Диаграмма Ганта представлена на рисунке 21.

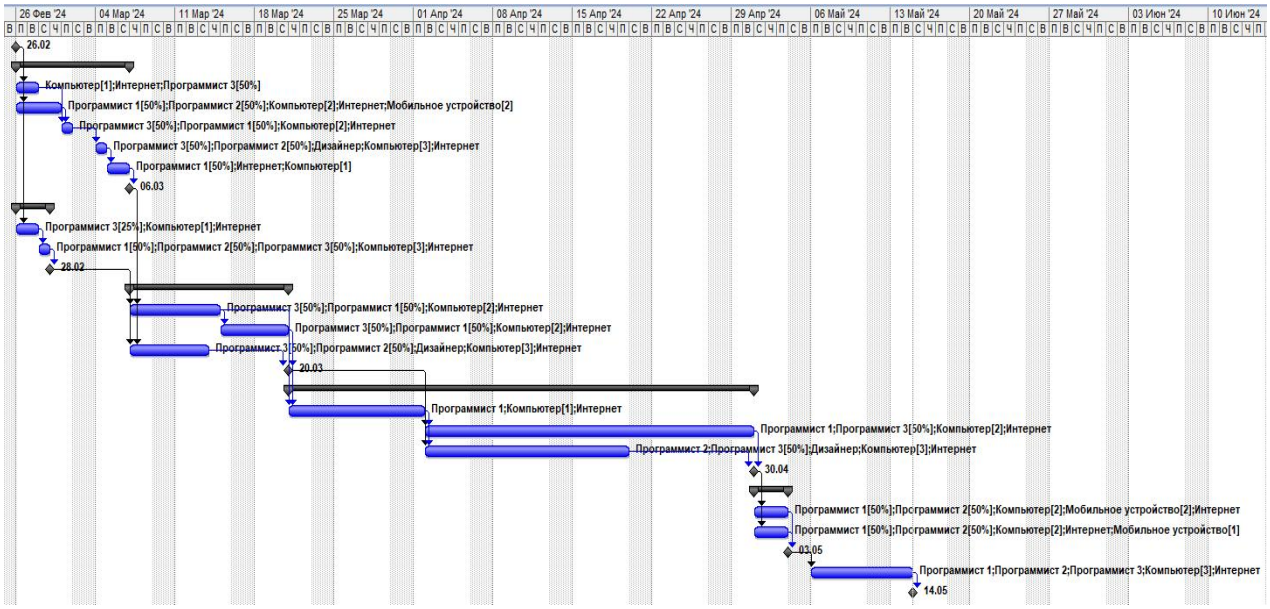


Рисунок 21 – Диаграмма Ганта

На рисунке 22 представлен график трудового ресурса «Программист1».

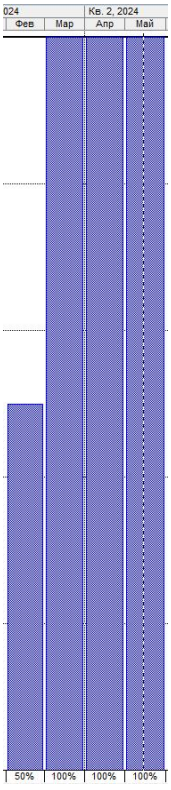


Рисунок 22 – График ресурсов для ресурса «Программист1»

На рисунках 23, 24 представлены графики трудовых ресурсов «Программист2», «Программист3».

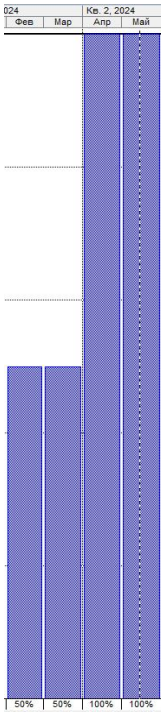


Рисунок 23 – График ресурсов для ресурса «Программист2»

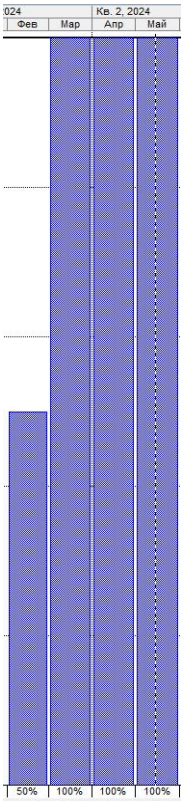


Рисунок 24 – График ресурсов для ресурса «Программист3»

На рисунке 25 представлен график трудового ресурса «Дизайнер».

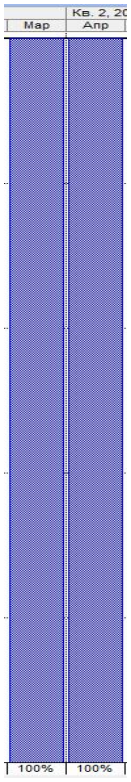


Рисунок 25 – График ресурсов для ресурса «Дизайнер»

На рисунке 26 представлен график материального ресурса «Компьютер».

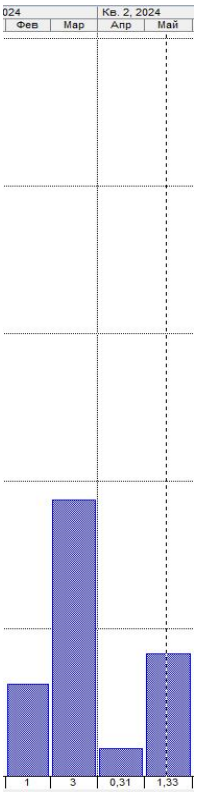


Рисунок 26 – График ресурсов для ресурса «Компьютер»

На рисунке 27 представлен график материального ресурса «Мобильное устройство».

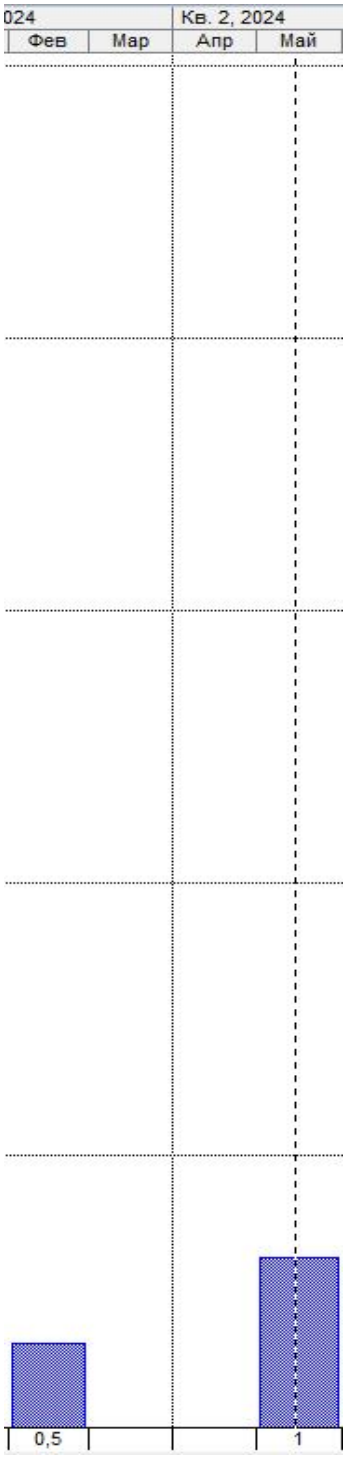
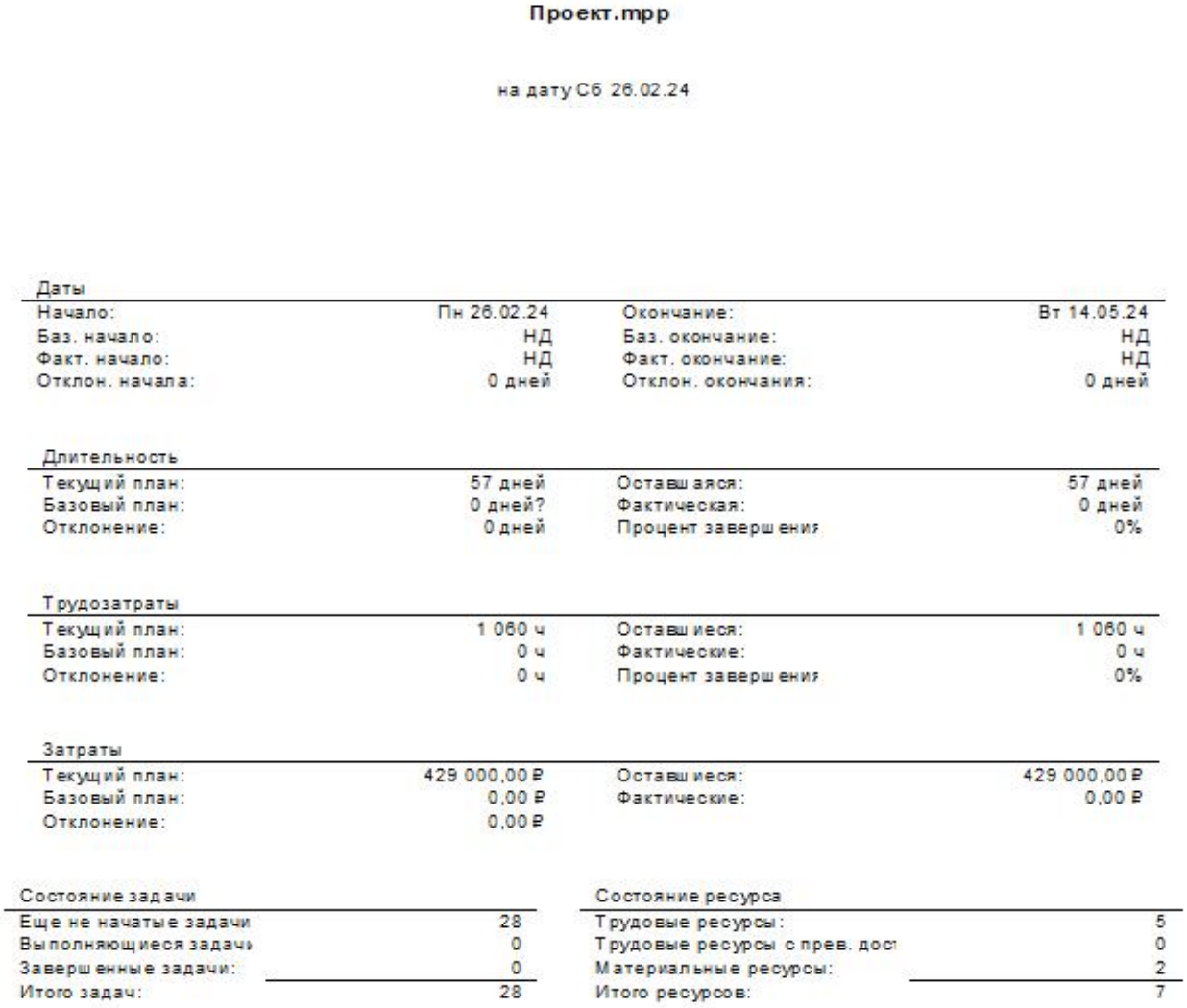


Рисунок 27 – График ресурсов для ресурса «Мобильное устройство»

На рисунке 28 представлен обзор проекта. Из рисунка видно, что общая длительность проекта составляет 57 дней (что совпадает со значением, вычисленным вручную), а предполагаемые затраты – 429 000 руб.



Проект.mpp

на дату Сб 26.02.24

Даты			
Начало:	Пн 26.02.24	Окончание:	Вт 14.05.24
Баз. начало:	НД	Баз. окончание:	НД
Факт. начало:	НД	Факт. окончание:	НД
Отклон. начала:	0 дней	Отклон. окончания:	0 дней

Длительность			
Текущий план:	57 дней	Оставшаяся:	57 дней
Базовый план:	0 дней?	Фактическая:	0 дней
Отклонение:	0 дней	Процент завершения	0%

Трудозатраты			
Текущий план:	1 060 ч	Оставшиеся:	1 060 ч
Базовый план:	0 ч	Фактические:	0 ч
Отклонение:	0 ч	Процент завершения	0%

Затраты			
Текущий план:	429 000,00 ₽	Оставшиеся:	429 000,00 ₽
Базовый план:	0,00 ₽	Фактические:	0,00 ₽
Отклонение:	0,00 ₽		

Состояние задачи	
Еще не начатые задачи	28
Выполняющиеся задачи	0
Завершенные задачи:	0
Итого задач:	28

Состояние ресурса	
Трудовые ресурсы:	5
Трудовые ресурсы с прев. дост	0
Материальные ресурсы:	2
Итого ресурсов:	7

Рисунок 28 – Обзорный отчёт проекта

На основе произведённых оценок стоимости и сроков проекта можно сделать вывод о его экономической целесообразности.

2 Проектирование мобильного приложения

2.1 Выбор технологий и инструментальных средств разработки

В процессе разработки приложения было решено использовать методологию RUP. Разработка приложения в соответствии с методологией RUP носит итеративный характер. На каждой итерации перед разработчиками ставятся определённые цели, которые должны быть достигнуты к её концу. В результате в конце каждой итерации должна получиться промежуточная функциональная версия разрабатываемого ПО. В методологии RUP существуют 4 основных фазы, каждая из которых может включать в себя одну или несколько итераций [16]:

– фаза начала проекта – формируется общее видение проекта и экономическое обоснование;

– уточнение – анализ предметной области и построение архитектуры продукта;

– построение – реализация основного функционала продукта;

– внедрение – получение финальной версии продукта и его передача заказчику, фаза включает в себя бета-тестирование и устранение выявленных в его процессе ошибок, а также обучение пользователей.

Каждая фаза завершается вехой, на которой команда разработчиков оценивает выполнение поставленных целей и принимает критически важные решения о ходе дальнейшей работы над продуктом.

Выбор в пользу данной методологии продиктован рядом её достоинств. С одной стороны, методология RUP обеспечивает гибкость процесса разработки и позволяет быстро вносить коррективы в проект в случае выявления в нем ошибок или изменения требований к разрабатываемому приложению. Кроме того, благодаря данной методологии удаётся выявить основные риски уже на ранних стадиях разработки. При этом RUP, в отличие от некоторых других популярных методологий, например XP, не требует постоянного взаимодействия с заказчиком, которое в рамках данного проекта невозможно.

Для анализа требований, а также моделирования структуры проекта в соответствии с принципами RUP было решено использовать диаграммы на языке моделирования UML [17]. Для построения диаграмм использовалось средство UML-моделирования Visual Paradigm.

На этапе планирования проекта было решено использовать приложение Microsoft Project. Оно позволяет обеспечить эффективное управление проектом на всех стадиях его выполнения. Microsoft Project предоставляет возможность выполнить структуризацию проекта, составить его календарный план, а также рассчитать затраты трудовых, материальных и денежных ресурсов в процессе работы над проектом.

Для создания мобильного приложения было решено использовать среду программирования Microsoft Visual Studio 2022, и язык программирования C#, а также современный кросплатформенный фреймворк .NET MAUI. Серверную часть было решено разрабатывать на языке C# с использованием среды программирования Visual Studio 2022. Для хранения данных используется БД, работающая под управлением СУБД Microsoft SQL Server. Проектирование базы данных выполнялось с использованием приложения Visual Paradigm. Реализация базы данных производилась с использованием языка программирования SQL.

2.2 Проектирование архитектуры программного обеспечения

Было принято решение использовать в приложении двухуровневую архитектуру «клиент-сервер». Её схема представлена на рисунке 29.

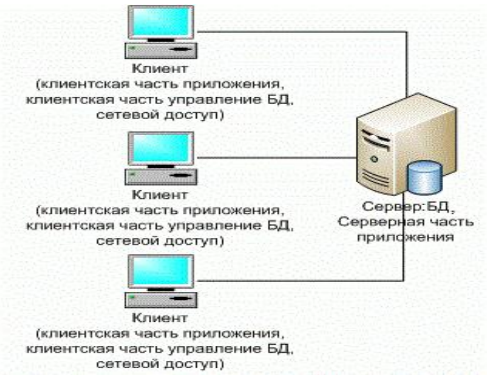


Рисунок 29 – Двухуровневая архитектура «клиент-сервер»

В данной архитектуре на сервере располагаются база данных и СУБД, а также серверная часть приложения, которая отвечает за непосредственное взаимодействие с базой данных и, как правило, реализуется в виде интерфейса прикладного программирования (API).

Клиентская же часть разрабатываемого ПО обеспечивает для пользователя удобный интерфейс, позволяющий пользователю просматривать необходимые ему данные из БД и взаимодействовать с ними, а также реализует возможность удалённо обращаться к серверу. В нашем случае его роль выполняет мобильное приложение.

Взаимодействие клиента и сервера осуществляется следующим образом: клиент делает HTTP-запрос серверу. Серверная часть приложения, в свою очередь, на основе полученного запроса создаёт SQL-запрос к базе данных. СУБД обрабатывает SQL-запрос и возвращает его результаты серверу, который на основе полученных результатов формирует ответ на запрос клиента. Получив ответ, клиент представляет его пользователю в удобной для того форме.

Достоинством архитектуры является то, что все данные хранятся на стороне сервера. Это позволяет минимизировать объем клиентского приложения. Кроме того, поскольку серверная часть приложения не зависит от клиентской, можно создать несколько вариантов клиента, отличающихся друг от друга, что может быть выгодно в случае потенциального распространения приложения и на другие платформы, помимо Android.

2.3 Проектирование интерфейса пользователя

На основе выявленных функциональных требований был спроектирован интерфейс мобильного приложения. Он включает в себя страницы входа, регистрации, главную страницу приложения, выбора вопроса, выбора билета, подготовки по вопросам, подготовки по билетам и записи на занятие.

Страница входа представлена на рисунке 30.

Рисунок 30 – Страница входа

Рассмотрим основные элементы интерфейса страницы входа: 1 – поле ввода логина пользователя, 2 – поле ввода пароля пользователя, 3 – кнопка «Вход», 4 – кнопка перехода на страницу регистрации. При нажатии на кнопку «Вход» система проверяет, существует ли на самом деле пользователь с таким логином и паролем. Если такого пользователя нет, будет выведено сообщение о неверном логине или пароле. Иначе авторизация будет считаться выполненной успешно.

Страница регистрации представлена на рисунке 31.

Рисунок 31 – Страница регистрации

Рассмотрим основные элементы интерфейса страницы регистрации: 1 – поле ввода логина пользователя, 2 – поле ввода пароля пользователя, 3 – кнопка «Регистрация», 4 – кнопка перехода на страницу входа. При нажатии кнопки «Регистрация» система проверяет, являются ли введённые логин и пароль допустимыми. Если это условие не выполняется, будет выведено сообщение о недопустимом логине или пароле. В противном случае будет проверено, не занят ли данный логин. Если он занят, выводится соответствующее сообщение. Иначе авторизация будет считаться выполненной успешно.

Главная страница приложения представлена на рисунке 32.

Рисунок 32 – Главная страница приложения

Рассмотрим основные элементы интерфейса главной страницы приложения: 1 – кнопка перехода на страницу записи на вождение, 2 – кнопка перехода на страницу подготовки по теоретическим вопросам, 3 – кнопка перехода на страницу подготовки по билетам.

Страница записи на вождение представлена на рисунке 33.

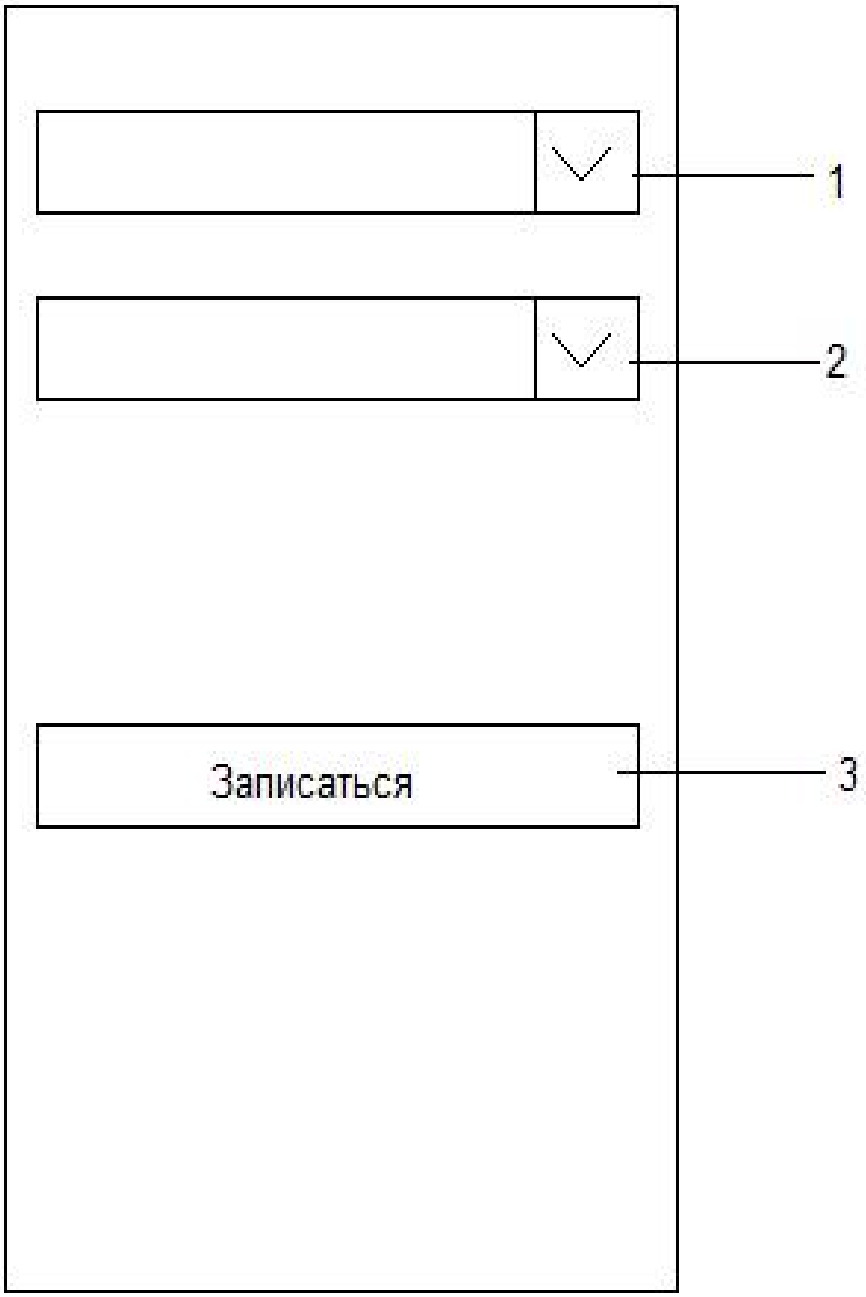


Рисунок 33 – Страница записи на вождение

Рассмотрим основные элементы интерфейса страницы записи на вождение: 1 – выпадающий список для выбора даты занятия, 2 – выпадающий список для выбора времени занятия, 3 – кнопка для записи на выбранные дату и время. При нажатии кнопки «Записаться» система проверяет, свободно ли выбранное время для записи. Если условие не выполняется, будет выведено сообщение о невозможности записи. В противном случае запись будет создана.

Страница выбора вопроса представлена на рисунке 34.

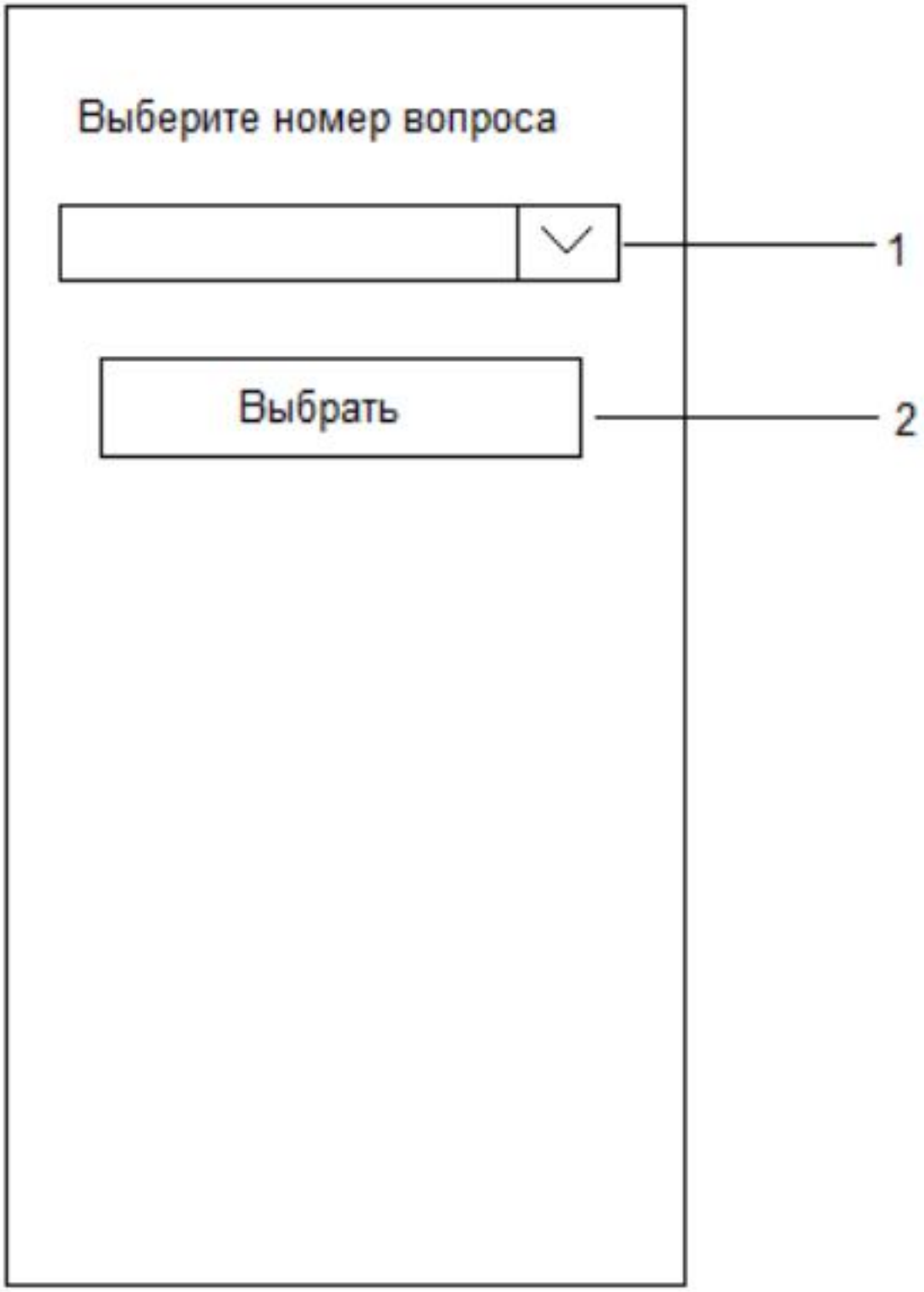


Рисунок 34 – Страница выбора вопроса

Рассмотрим основные элементы интерфейса страницы выбора вопроса: 1 – выпадающий список для выбора номера теоретического вопроса, 2 – кнопка перехода на страницу с выбранным вопросом.

Страница с вопросом представлена на рисунке 35.

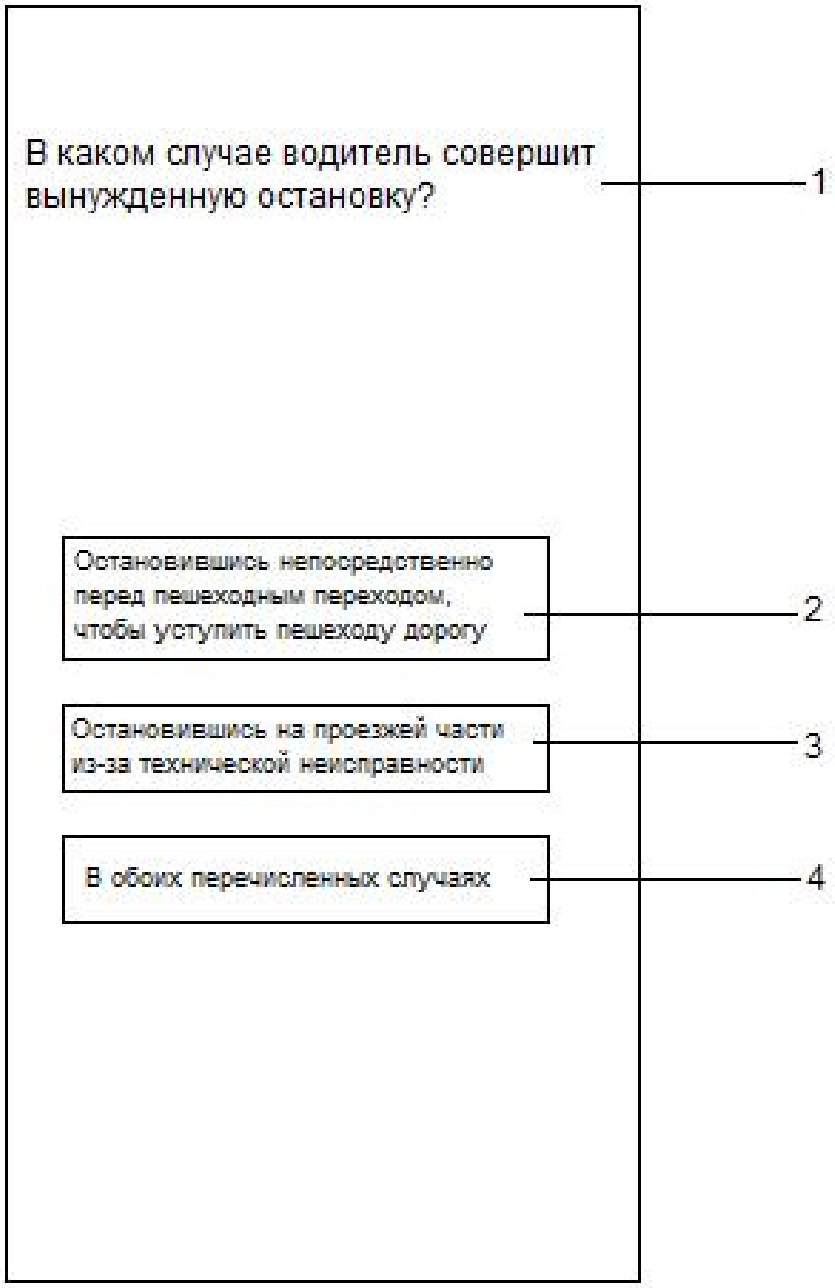


Рисунок 35 – Страница с вопросом

Рассмотрим основные элементы интерфейса страницы с вопросом: 1 – формулировка вопроса, 2, 3, 4 – кнопки для выбора одного из предложенных ответов. При нажатии кнопки с ответом система фиксирует данный ответ в базе данных. Если ответ был неверным на экран выводится краткое пояснение почему данный вариант ответа неверный. В противном случае открывается страница со следующим вопросом.

Страница выбора билета представлена на рисунке 36.

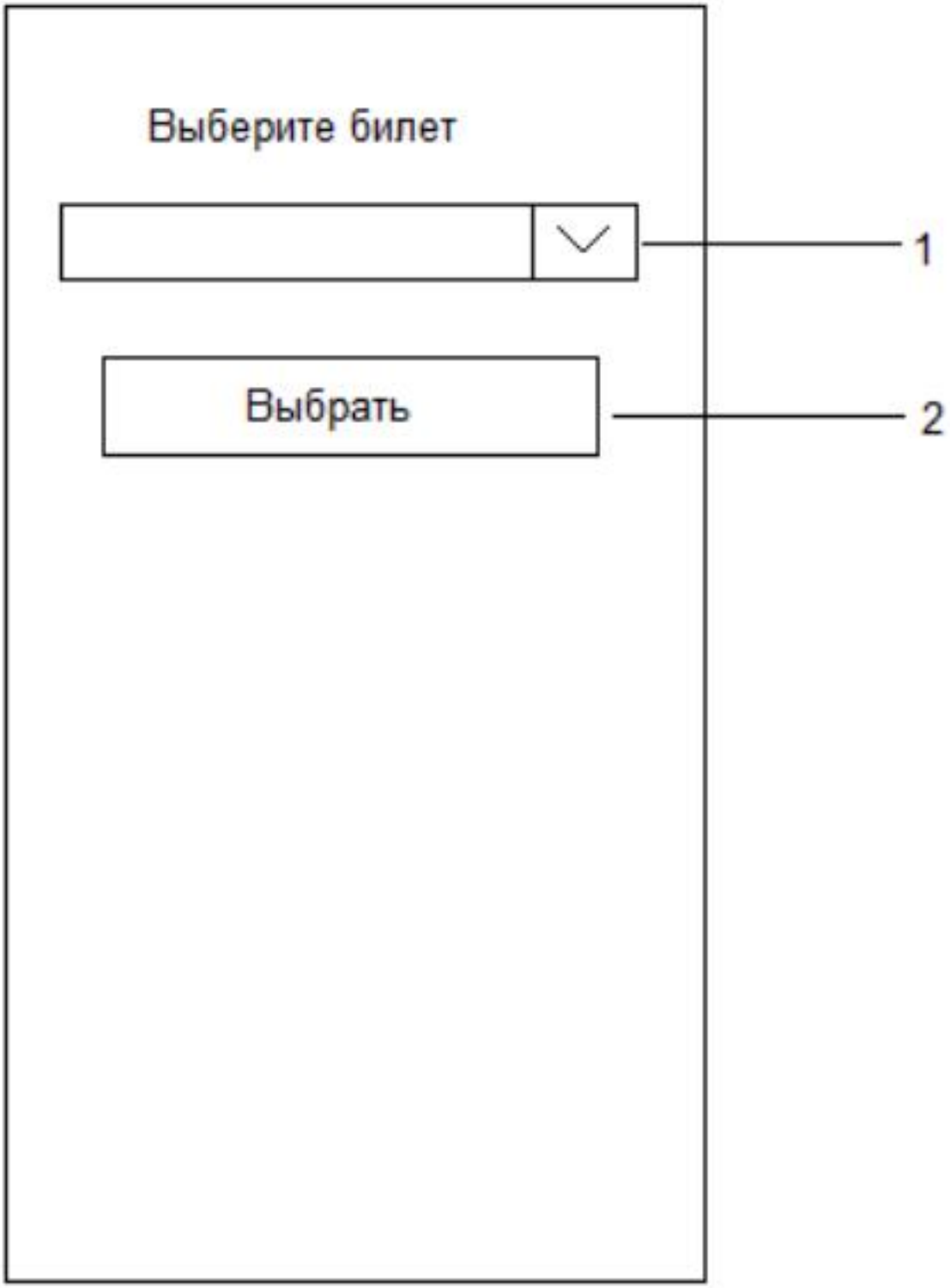


Рисунок 36 – Страница выбора билета

Рассмотрим основные элементы интерфейса страницы выбора билета: 1 – выпадающий список для выбора номера билета, 2 – кнопка перехода на страницу с выбранным билетом.

Страница с билетом представлена на рисунке 37.

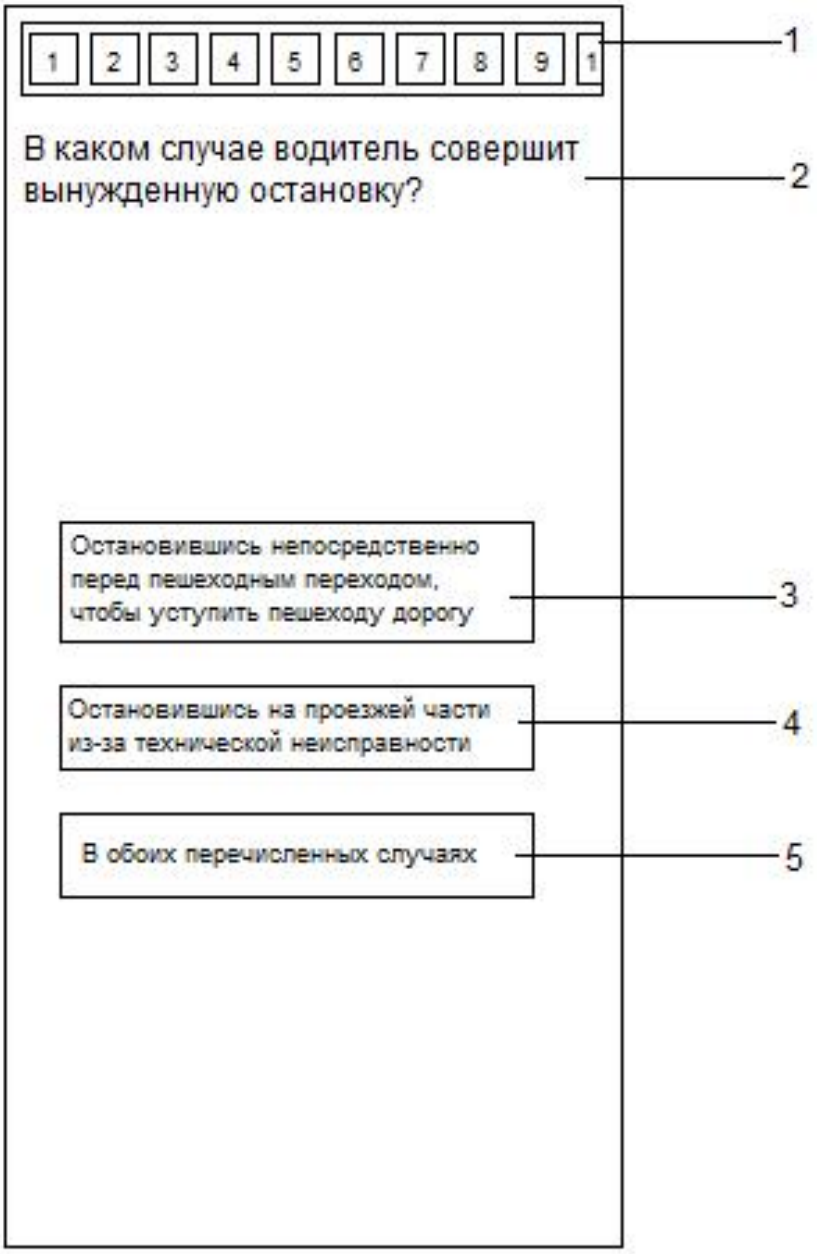


Рисунок 37 – Страница с билетом

Рассмотрим основные элементы интерфейса страницы с билетом: 1 – карусель для выбора вопроса в билете, 2 – формулировка вопроса, 3, 4, 5 – кнопки для выбора одного из предложенных ответов. При нажатии кнопки с ответом система проверяет корректность ответа. Если ответ был неверным, к билету добавляется один теоретический вопрос. Если курсант дал три неверных ответа билет не засчитывается. После проверки корректности открывается следующий вопрос в билете.

2.4 Логическое проектирование базы данных

На основе анализа предметной области и функциональных требований была построена концептуальная модель базы данных [18]. Она представлена в виде диаграммы «сущность-связь» (рисунок 38).

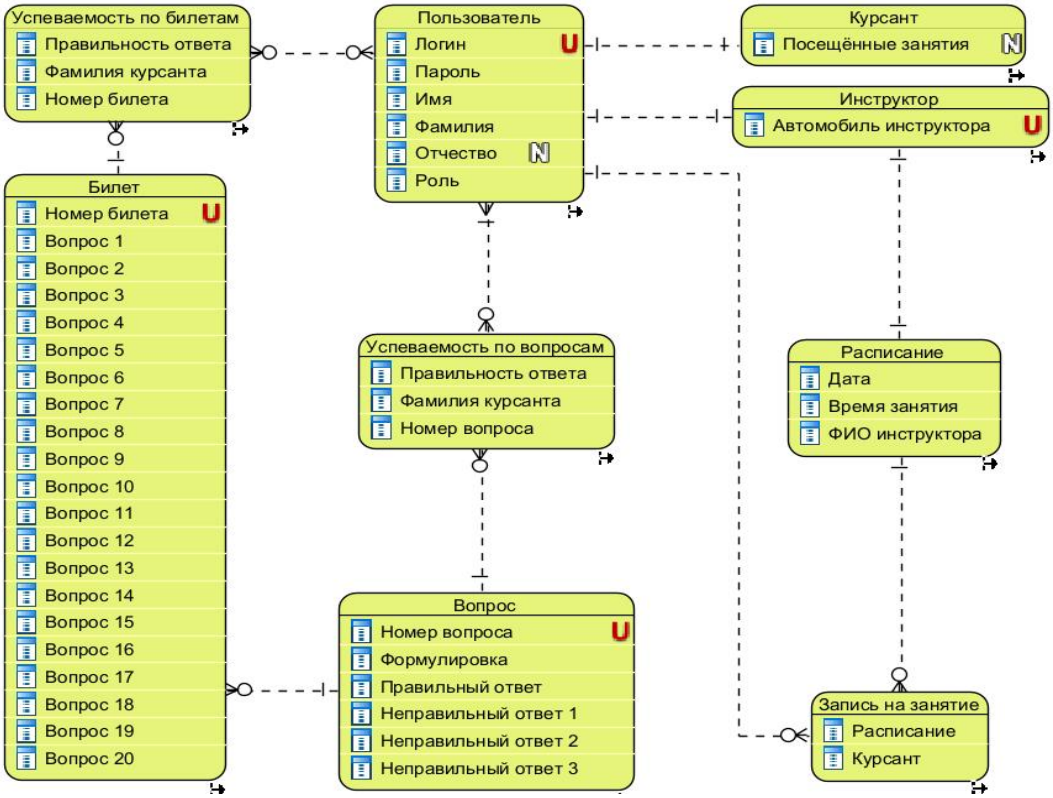


Рисунок 38 – Диаграмма «сущность-связь»

Сущности концептуальной модели данных описаны в таблице 8.

Таблица 8 – Сущности предметной области

№ п/п	Наименование	Описание	Категория	Количество экземпляров	Режим обновления
1	Пользователь	Список пользователей	Стержневая	100	90% в 2 месяца
2	Курсант	Список курсантов	Стержневая	90	100% в 2 месяца
3	Инструктор	Список инструкторов	Стержневая	10	Нет
4	Расписание	Дата и время занятий у инструктора	Ассоциативная	10	100% в неделю
5	Запись на занятие	Запись курсанта на занятие	Ассоциативная	70	Нет
6	Вопрос	Теоретический вопрос	Стержневая	800	Нет

Продолжение таблицы 8

№ п/п	Наименование	Описание	Категория	Количество экземпляров	Режим обновления
7	Успеваемость по вопросам	Правильность ответов на вопросы пользователем	Ассоциативная	72000	100% в 2 месяца
8	Билет	20 теоретических вопросов	Стержневая	40	Нет
9	Успеваемость по билетам	Решённые/нерешенные билеты	Ассоциативная	3600	100% в 2 месяца

В таблице 9 представлено описание связей между сущностями предметной области.

Таблица 9 – Описание связей между сущностями предметной области

№ п/п	Сущ-ность 1	Сущ-ность 2	Семантика связи	Мощ-ность	Обязательность	Ограни-чения
1	Пользо-ватель	Курсант	Пользователь авторизовался как курсант	1:1	Обязательная для пользователя – Пользователь должен быть авторизованным	
2	Пользо-ватель	Инструк-тор	Пользователь авторизовался как инструктор	1:1	Обязательная для пользователя – Пользователь должен быть авторизованным	
3	Инструк-тор	Расписан ие	У инструктора есть расписание Расписание относится к конкретному инструктору	1:M	Обязательно для инструктора – у инструктора только одно расписание Обязательно для расписания – расписание используется только одним инструктором	
4	Расписа-ние	Запись на занятие	В расписании содержатся все будущие занятия инструктора	1:M	Не обязательно для расписания – расписание может содержать несколько занятий Обязательно для записи на занятие – запись может быть только в одном расписании	
5	Пользо-ватель	Успева-емость по вопро-сам	У курсанта есть успеваемость по вопросу Успеваемость по вопросу есть у каждого курсанта	1:M	Обязательно для вопроса – Вопрос может быть решён или не решён Обязательно для курсанта – на все вопросы дан ответ	Вопрос и курсант должны существ овать

Продолжение таблицы 9

№ п/п	Сущ-ность 1	Сущ-ность 2	Семантика связи	Мощность	Обязательность	Ограничения
6	Пользователь	Успеваемость по билетам	У курсанта есть успеваемость по каждому билету Успеваемость по билету есть у каждого курсанта	1:M	Обязательно для билета – Билет может быть решён или не решён Обязательно для курсанта – все билеты решены	Билет и пользователь должны существовать
7	Пользователь	Запись на занятие	Курсант может записаться на несколько занятий На одно занятие может быть записан только один курсант	1:M	Необязательно для пользователя – пользователь может не записываться на занятие Необязательно для занятия – на занятия может никто не записаться	
8	Вопрос	Успеваемость по вопро-сам	На каждый вопрос есть результаты ответа всех курсантов	1:M	Необязательно для вопроса – Вопрос может без успеваемости, если его ещё никто не решал Обязательно для успеваемости по вопросам – вопрос должен существовать	Вопрос должен существовать
9	Вопрос	Билет	Билет содержит 20 вопросов	1:M	Необязательно для вопроса – вопрос может не закрепляться за билетом Обязательно для билета – билет должен иметь 20 вопросов	
10	Билет	Успеваемость по билетам	На каждый билет есть результаты ответа всех курсантов	1:M	Необязательно для билета – Билет может без успеваемости, если его ещё никто не решал Обязательно для успеваемости по билетам – билет должен существовать	Билет должен существовать

На основании концептуальной модели данных в среде Visual Paradigm была построена логическая модель базы данных.

Её структура описана диаграммой в нотации IDEF1X, представленной на рисунке 39.

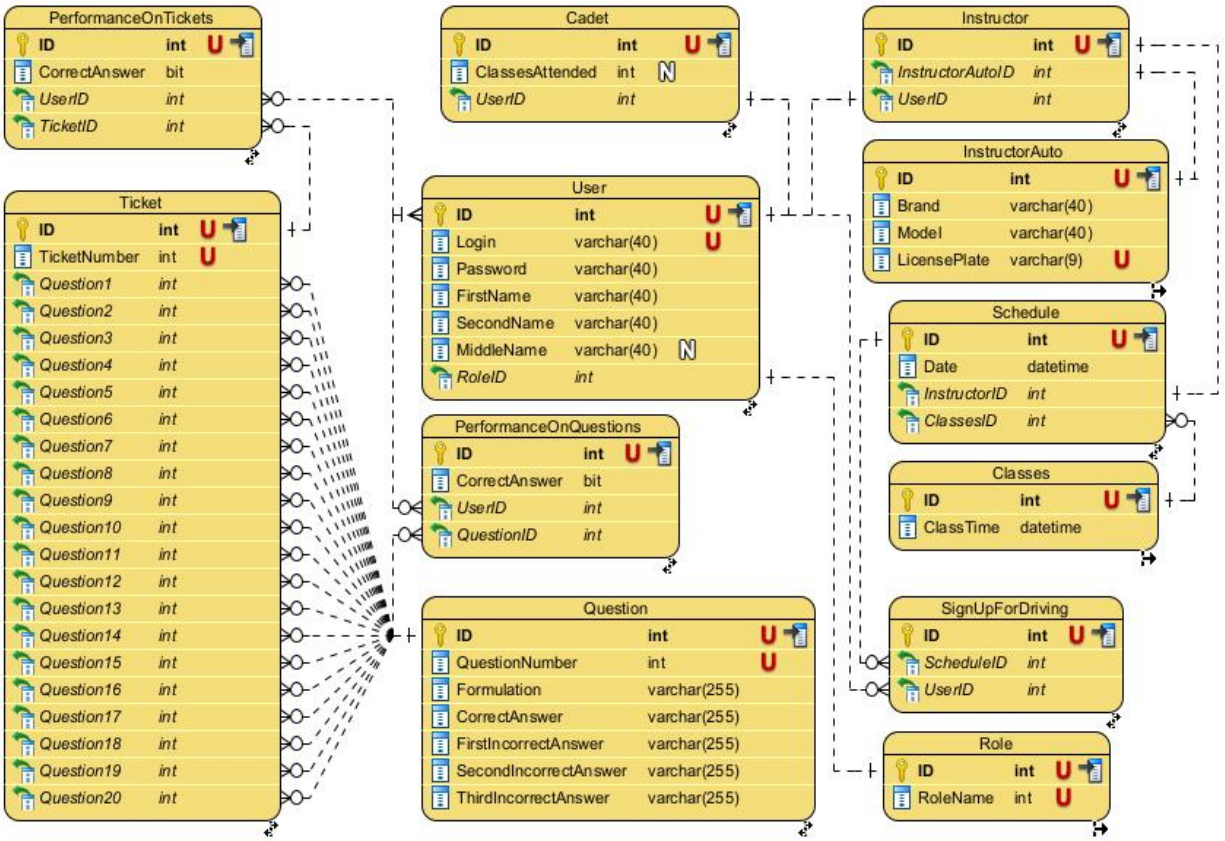


Рисунок 39 – Логическая модель базы данных

На представленной логической модели спроектированная структура БД отображена уже более подробно. Здесь не только присутствуют типы сущностей и связи между ними, но и хранится информация о конкретных атрибутах сущностей [19]. Однако на данный момент сущности все ещё представлены в абстрактной форме, без привязки к конкретной реализации базы данных.

2.5 Проектирование серверной части приложения

Серверная часть приложения представляет собой REST API. Доступ к данным осуществляется при помощи протокола HTTP с использованием его стандартных методов (GET применяется для получения данных ресурса, POST – для создания нового ресурса, PUT – для обновления существующего,

DELETE – для удаления ресурса). Обращение к ресурсам происходит при помощи уникальных идентификаторов ресурсов (URI).

В приложение используется получившая широкое распространение архитектура MVC (Model-View-Controller) [20]. Её схема представлена на рисунке 40.

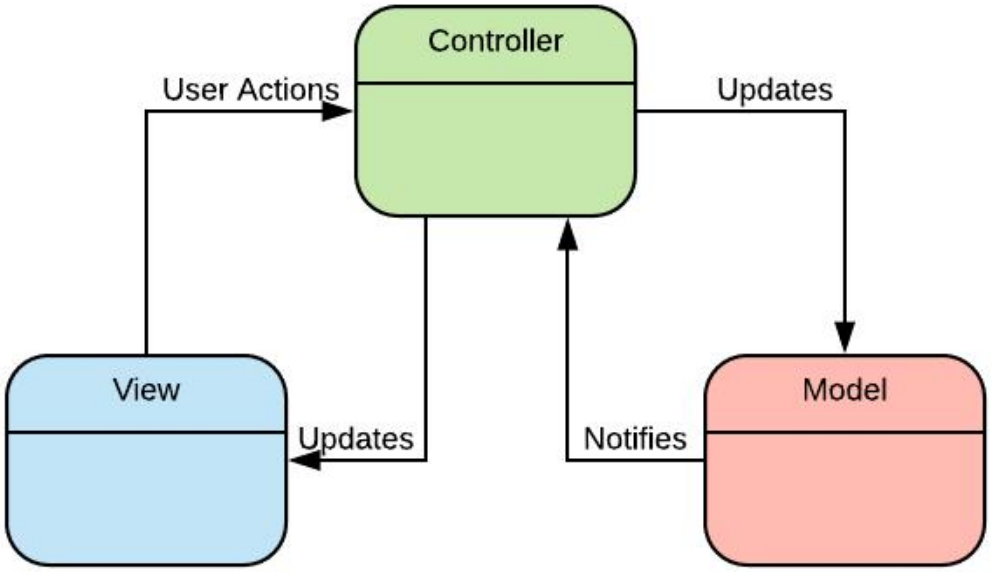


Рисунок 40 – Архитектура MVC

Серверная часть приложения включает в себя классы модели, отвечающие за организацию доступа к данным (т.е. за непосредственное взаимодействие с БД) и классы контроллера, отвечающие за организацию взаимодействия пользователя с клиентом. Именно они обрабатывают HTTP-запросы клиента и формируют ответы на них. К клиенту же отойдёт только уровень представления, отвечающий за взаимодействие с пользователем.

3 Реализация мобильного приложения

3.1 Реализация базы данных

На основе составленной ранее логической модели БД с учётом специфики СУБД Microsoft SQL Server при помощи программы Visual Paradigm была построена физическая модель [22]. Её структура представлена на рисунке 41.

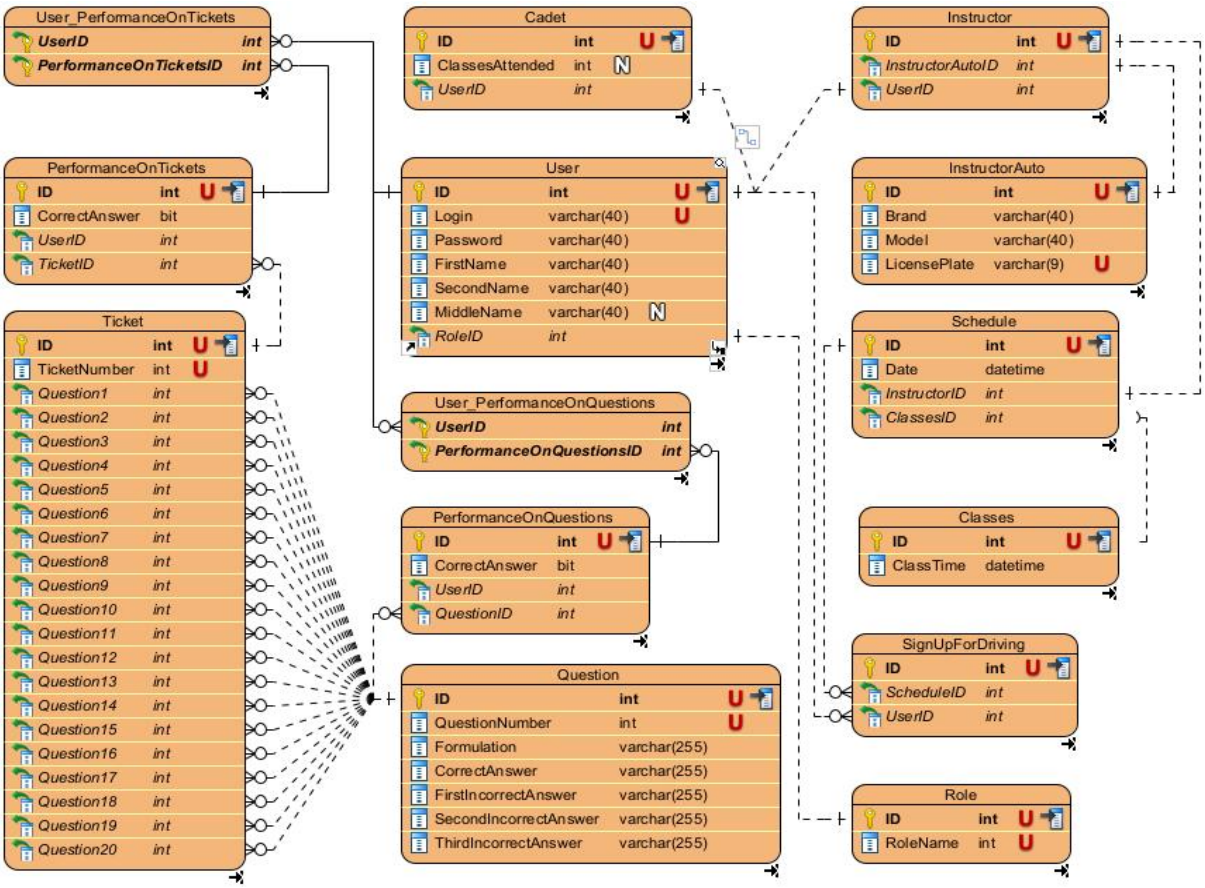


Рисунок 41 – Физическая модель базы данных

На основе составленной физической модели БД был разработан скрипт на языке SQL для создания базы данных [22]. Его код представлен в приложении Б. Развёртывание спроектированной базы данных было выполнено путём запуска данного скрипта с использованием утилиты SQL Server Management Studio, используемой для администрирования баз данных под управлением СУБД Microsoft SQL Server.

3.2 Разработка серверной части

Разработка серверной части приложения производилась в среде программирования Microsoft Visual Studio 2022 на языке C# с применением технологии Web API [23, 24]. Для взаимодействия с базой данных использовалась технология Entity Framework [25]. Диаграмма классов серверной части представлена на рисунке 41.

В соответствии с используемой архитектурой MVC все классы серверной части можно разделить на два слоя: слой модели, предназначенный для взаимодействия с БД, и слой контроллеров, предназначенный для обработки HTTP-запросов к серверу.

Диаграмма классов слоя модели представлена на рисунке 42. Все эти классы были автоматически сгенерированы средой программирования Visual Studio с использованием технологии Entity Framework. Генерация производилась на основе предварительно разработанной до этого базы данных (подход Database First) [26, 27].

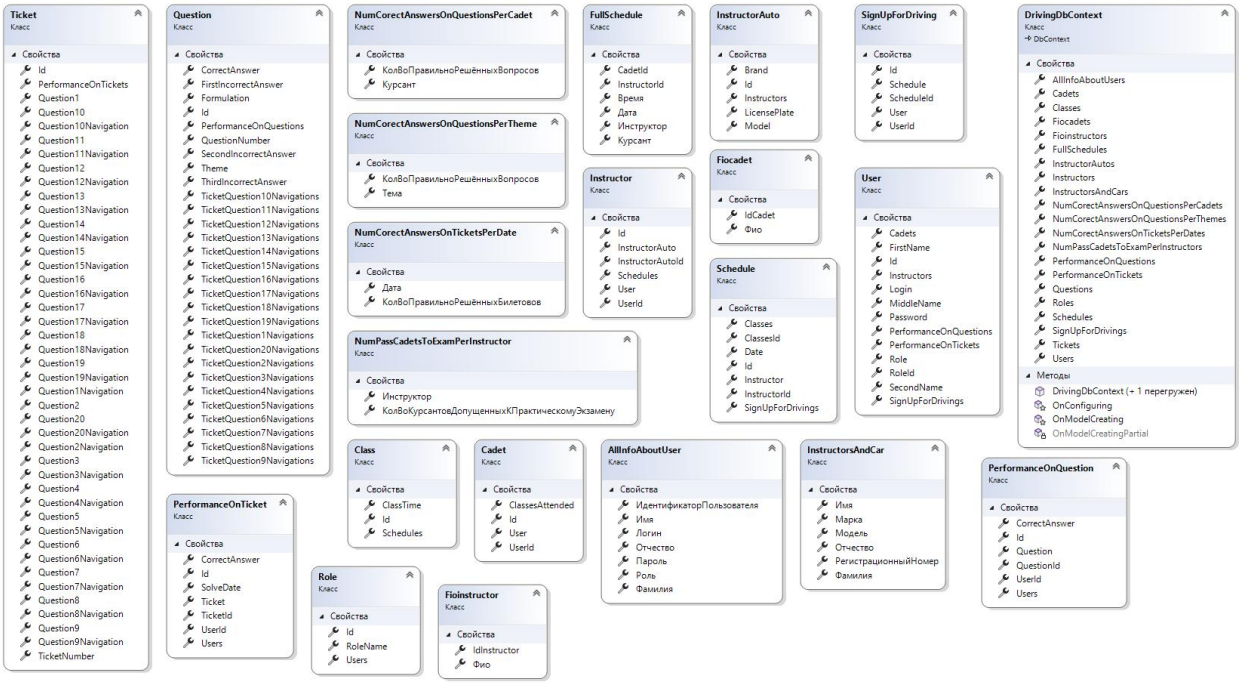


Рисунок 42 – Диаграмма классов серверной части (слой модели)

Как можно заметить, классы Ticket, Question, PerformanceOnTickets,

PerformanceOnQuestons, Class, Role, Cadet, Classes, Instructor, InstructorAuto, Schedule, SignUpForDriving, User были созданы на основании одноименных таблиц базы данных. Классы AllInfoAboutUsers, FIOCadet, FIOInstructors, FullSchedule, InstructorsAndCars, NumCorectAnswersOnQuestionsPerCadet, NumCorectAnswersOnQuestionPerTheme, NumCorectAnswersOnTicketsPerDate, NumPassCadetsToExamPerInstructor же основываются на одноимённых представлениях и позволяют несколько облегчить выборку нужных данных из базы.

Диаграмма классов слоя контроллеров представлена на рисунке 43.

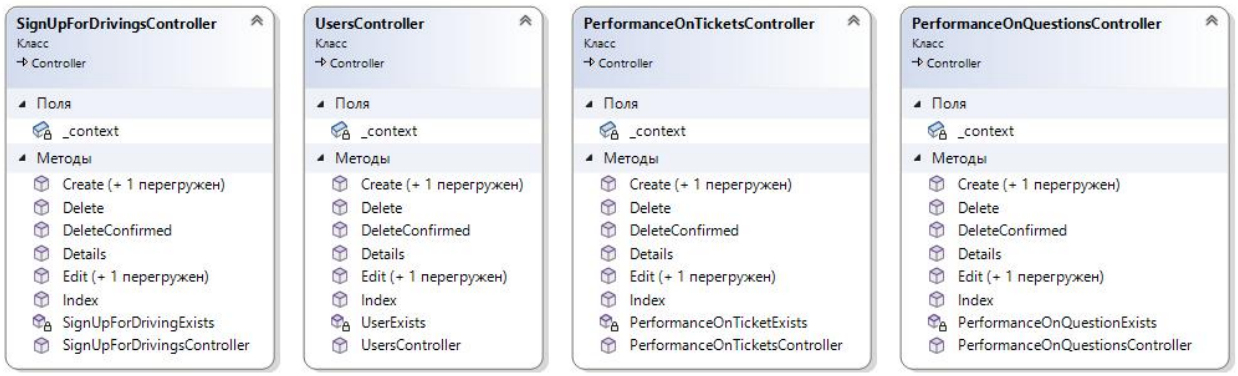


Рисунок 43 – Диаграмма классов серверной части (слой контроллеров)

Более подробное описание классов слоя контроллеров и их назначений представлено в таблице 10.

Таблица 10 – Описание классов слоя контроллеров

Класс	Назначение
SignUpForDrivingController	Отвечает за обработку запросов по созданию и отмене записи на практическое занятие по вождению.
PerformanceOnQuestionsController	Отвечает за обработку запросов на сохранение результатов подготовки по вопросам пользователя.
PerformanceOnTicketsController	Отвечает за обработку запросов на сохранение результатов подготовки по билетам пользователя.
UsersController	Отвечает на обработку запросов на регистрацию нового пользователя и на вход в систему уже зарегистрированного, на смену пароля пользователя, на получение полных списков отзывов и загрузок некоторого пользователя, на получение информации о пользователе и на его блокировку.

Листинг серверной части представлен в приложении В.

3.3 Реализация клиентской части приложения

Разработка мобильного приложения производилась на языке C# с использованием среды программирования Visual Studio 2022. Диаграмма компонентов, отражающая структуру программы, представлена на рисунке 44.

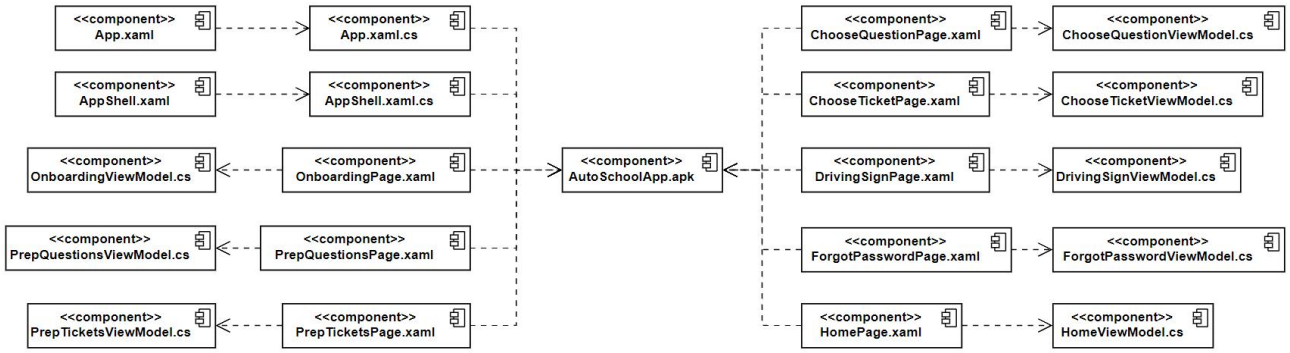


Рисунок 44 – Диаграмма компонентов мобильного приложения

Описание компонентов представлено в таблице 11.

Таблица 11 – Компоненты клиентского приложения

Компонент	Назначение
App.xaml	XAML-файл описания подключаемых ресурсов к приложению
App.xaml.cs	Является начальной точкой загрузки, закрытия приложения
AppShell.xaml	XAML-файл описания структуры интерфейса приложения
AppShell.xaml.cs	Регистрирует страницы приложения, не вошедшие в основную структуру
SignInPage.xaml	XAML-файл описания интерфейса страницы авторизации
SignInViewModel.cs	Управляет логикой и данными страницы авторизации
SignUpPage.xaml	XAML-файл описания интерфейса страницы регистрации
SignUpViewModel.cs	Управляет логикой и данными страницы регистрации
ChooseQuestionPage.xaml	XAML-файл описания интерфейса страницы выбора вопроса
ChooseQuestionViewModel.cs	Управляет логикой и данными страницы выбора вопроса
ChooseTicketPage.xaml	XAML-файл описания интерфейса страницы выбора билета
ChooseTicketViewModel.cs	Управляет логикой и данными страницы выбора билета
DrivingSignPage.xaml	XAML-файл описания интерфейса страницы записи на занятие по вождению
DrivingSignViewModel.cs	Управляет логикой и данными страницы записи на занятие по вождению
ForgotPasswordViewModel.cs	Управляет логикой и данными страницы восстановления пароля

Продолжение таблицы 11

Компонент	Назначение
ForgotPasswordPage.xaml	XAML-файл описания интерфейса страницы восстановления пароля
HomePage.xaml	XAML-файл описания интерфейса домашней страницы
HomeViewModel.cs	Управляет логикой и данными домашней страницы
PrepQuestionsPage.xaml	XAML-файл описания интерфейса страницы подготовки по вопросам
PrepQuestionsViewModel.cs	Управляет логикой и данными страницы подготовки по вопросам
PrepTicketsPage.xaml	XAML-файл описания интерфейса страницы подготовки по билетам
PrepTicketsViewModel.cs	Управляет логикой и данными страницы подготовки по билетам

Диаграмма развертывания, на которой описывается физическое расположение компонентов системы, представлена на рисунке 45.

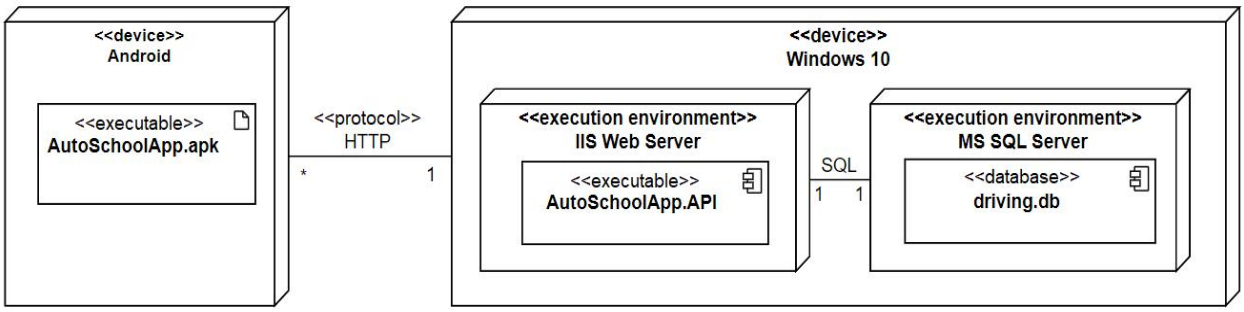


Рисунок 45 – Диаграмма развертывания мобильного приложения

Как можно увидеть на этой диаграмме, клиентское приложение работает на мобильном устройстве под управлением системы Android. API и база данных располагаются на сервере, работающем под управлением ОС Windows. Взаимодействие между серверной и клиентской частями приложения происходит с использованием протокола HTTP.

Листинг клиентской части программы представлен в приложении Г.

4 Тестирование программного продукта

4.1 Тестирование серверной части приложения

Для начала необходимо убедиться в корректности работы серверной части приложения. С этой целью было произведено тестирование серверной части приложения с использованием инструментального средства Postman. Результаты тестирования функционала серверной части представлены в таблице 12. Скриншоты тестов находятся в приложении Д.

Таблица 12 – Результаты тестирования серверной части

№	Описание теста	Запрос		Результат
		Метод	Параметры	
1	Добавление в систему пользователя с логином someuser и паролем password12	POST	login = someuser, password= password12	Тест пройден. Пользователь успешно добавлен в систему. Код ответа HTTP – 201 (рисунок Д.1).
2	Попытка входа в систему пользователя с логином someuser и паролем password12	GET	login = some-user, pass-word= pass-word 12	Тест пройден. Пользователь успешно авторизован. Код ответа HTTP – 200 (рисунок Д.2).
3	Получение сведений о пользователе с идентификатором 1	GET	id = 1	Тест пройден. Информация о пользователе успешно получена. Код ответа HTTP – 200 (рисунок Д.3).
4	Изменение пароля пользователя с идентификатором 8 с password12 на newpwd1234	PUT	id= 8, oldpassword= password12, newpassword= newpwd1234	Тест пройден. Пароль успешно изменен. Код ответа HTTP – 200 (рисунок Д.4).
5	Создание записи на занятие	POST	userId=39, date = 2024-06-06, time =15:30	Тест пройден. Запись успешно создана. Код ответа HTTP – 201 (рисунок Д.5).
6	Отмена записи на занятие	DELE-TE	userId=39, date = 2024-06-06, time =15:30	Тест пройден. Запись успешно отменена. Код ответа HTTP – 200 (рисунок Д.6).
7	Сохранение ответа на теоретический вопрос	POST	userId=39, correct = true, questionId=108	Тест пройден. Ответ успешно сохранен. Код ответа HTTP – 200 (рисунок Д.7).
8	Сохранение ответов на вопросы в билете	POST	userId=39, correct = true, ticketId=19	Тест пройден. Ответы успешно сохранены. Код ответа HTTP – 200 (рисунок Д.8).

Все тесты серверной части были пройдены успешно, что свидетельствует о корректности работы серверной части приложения и позволяет перейти к

тестированию клиентской части разрабатываемого мобильного приложения.

4.2 Тестирование клиентской части приложения

4.2.1 Функциональное тестирование

Для проверки корректности работы клиентской части приложения было проведено функциональное тестирование. Его результаты представлены в таблице 13. Скриншоты тестов находятся в приложении Е.

Таблица 13 – Функциональное тестирование

Вариант использования	Тест и параметры	Результат
Зарегистрироваться в системе	Ввести валидные данные для регистрации. Логин: mylogin. Пароль: qwerty1234.	Тест пройден. Пользователь зарегистрирован в системе (рисунок Е.1).
	Ввести данные с логином существующего в системе пользователя. Логин: mylogin.	Тест пройден. Выведено уведомление о том, что логин уже занят (рисунок Е.2).
	Попытаться задать логин и пароль с недопустимыми символами. Логин: логин. Пароль: пароль.	Тест пройден. Выведено уведомление о вводе недопустимых данных (рисунок Е.3).
	Попытаться задать логин и пароль недопустимой длины. Логин: m. Пароль: qwe1.	Тест пройден. Выведено уведомление о вводе недопустимых данных (рисунок Е.4).
	Произвести попытку задать пароль без букв. Логин: mylogin1 Пароль: qwertyuiop	Тест пройден. Выведено уведомление о вводе недопустимых данных (рисунок Е.5).
Войти в систему	Ввести валидные данные. Логин: mylogin. Пароль: qwerty1234..	Тест пройден. Авторизация выполнена успешно (рисунок Е.6).
	Ввести неверные данные. Логин: mylogin. Пароль: qwerty123456.	Тест пройден. Выведено уведомление о неверном логине или пароле (рисунок Е.7).
	Ввести данные заблокированного пользователя. Логин: blockedLogin. Пароль: qqqww321.	Тест пройден. Выведено сообщение о блокировке (рисунок Е.8).

Продолжение таблицы 13

Вариант использования	Тест и параметры	Результат
Сделать запись на практическое занятие	Записаться на свободное время. Дата: Время:	Тест пройден. Выведено сообщение о том, что запись успешно создана (рисунок Е.9).
	Записаться на забронированное время. Дата: Время:	Тест пройден. Выведено сообщение о том, что выбранное время уже занято (рисунок Е.10).
Готовиться по вопросам	Дать правильный ответ на вопрос. Ответ: 1	Тест пройден. Выбранный ответ подсвечен зеленым (рисунок Е.11).
	Дать правильный ответ на вопрос. Ответ: 2	Тест пройден. Выбранный ответ подсвечен красным, правильны – зеленым, а также выведено пояснение почему выбранный ответ неверный (рисунок Е.12).

Функциональное тестирование показало корректность работы клиентской части мобильного приложения.

4.2.2 Модульное тестирование

Модульные тесты – автоматизированные тесты, позволяющие оценить корректность работы некоторой автономной единицы программного кода (обычно подпрограммы) путем сравнения фактического результата работы с ожидаемым. Основное достоинство модульных тестов состоит в том, что они позволяют быстро выявить и устранить ошибки, возникающие при модификации кода.

В рамках данного проекта было решено покрыть модульными тестами методы класса PerformanceOnQuestionsController Create и Update, проверяющие на корректность предложенные пользователем при регистрации значения логина и пароля соответственно.

Для проведения модульных тестов был разработан тестовый класс PerformanceOnQuestionsControllerTests, код которого представлен в листинге 1.

Листинг 1 – Исходный код модульных тестов

```
using AutoSchoolApp.API.Models;

namespace AutoSchoolApp.API.Controllers.Tests
```

```
{
    [TestClass()]
    public class PerformanceOnQuestionsControllerTests
    {
        private readonly DrivingDbContext _context = new();

        [TestMethod()]
        public void CreateTest()
        {
            var controller = new
PerformanceOnQuestionsController(_context);

            var response = controller.Create(39, true, 108);

            Assert.IsTrue(response.IsCompletedSuccessfully);
        }

        [TestMethod()]
        public void UpdateTest()
        {
            var controller = new
PerformanceOnQuestionsController(_context);

            var response = controller.Update(39, false, 108);

            Assert.IsTrue(response.IsCompletedSuccessfully);
        }
    }
}
```

Результат проведения модульных тестов представлен на рисунке 46.

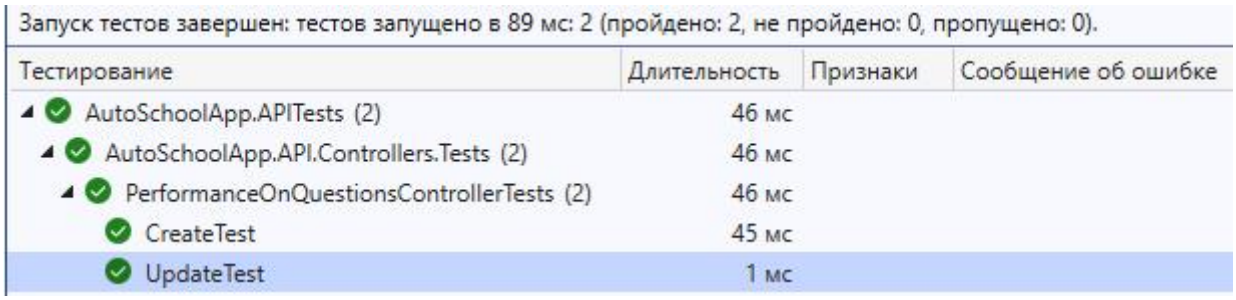


Рисунок 46 – Результаты модульного тестирования

Все тесты были успешно пройдены, что свидетельствует о корректной работе тестируемых методов.

4.3 Метрики кода

Метриками кода называются некоторые количественные величины, позволяющие подсчитать, насколько код программы обладает тем или иным свойством. Существует огромное количество самых разнообразных метрик,

имеющих различное назначение. Наиболее известными являются следующие группы метрик:

– метрики размера программы (SLOC-метрики);

– метрики сложности потока управления программ (метрики Холстеда и цикломатическая сложность).

Рассмотрим метод button1_Click класса PrepTicketsPage, отвечающий за отправку и сохранение ответа пользователя в базе данных. Его код представлен в листинге 2.

Листинг 2 – Метод обработки нажатия на кнопку «Отправить ответ»

```
/// <summary>
/// Метод обработки нажатия на кнопку "Отправить ответ"
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void button1_Click(object sender, EventArgs e)
{
     var sqlExpression = $"select ID from Question where
QuestionNumber='{qn}'";
     var sqlCommand = new SqlCommand(sqlExpression, sc);
     var QuestionId = sqlCommand.ExecuteScalar();
     sqlExpression = $"select*from PerformanceOnQuestions where
UserID=39 and QuestionID='{QuestionId}'";
     sqlCommand = new SqlCommand(sqlExpression, sc);
     var resul = sqlCommand.ExecuteScalar();
     //Проверка – правильно ли решен этот вопрос
     if (resul != null)
     {
          //Проверка – правильно ли решен этот вопрос
          if (radioButton1.Checked)
          {
               sqlExpression = $"update PerformanceOnQuestions set
CorrectAnswer=1 where UserID=UID and QuestionID='{QuestionId}'";
               sqlCommand = new SqlCommand(sqlExpression, sc);
               sqlCommand.ExecuteNonQuery();
               }
          else
          {
               sqlExpression = $"update PerformanceOnQuestions set
CorrectAnswer=0 where UserID=UID and QuestionID='{QuestionId}'";
               sqlCommand = new SqlCommand(sqlExpression, sc);
               sqlCommand.ExecuteNonQuery();
          }
`     }
     else
     {
          if (radioButton1.Checked)
          {
               sqlExpression = $"insert
          PerformanceOnQuestions(CorrectAnswer, UserID,QuestionID)" +
```

```
                $" values (1,'{UID}','{QuestionId}')";
                sqlCommand = new SqlCommand(sqlExpression, sc);
                sqlCommand.ExecuteNonQuery();
            }
            else
            {
                sqlExpression = $"insert
PerformanceOnQuestions(CorrectAnswer,UserID,QuestionID)" +
            $" values (0,'{UID}','{QuestionId}')";
            sqlCommand = new SqlCommand(sqlExpression, sc);
            sqlCommand.ExecuteNonQuery();
            }
      }
      this.Hide();
}
```

Данный метод находится в клиентской части приложения и запускается при нажатии пользователем на кнопку «Отправить ответ». Он проверяет на корректность ответ пользователя, и сохраняет результат в базе данных.

Для данного метода будут вычислены две группы метрик: метрики размера программы и метрики Холстеда.

4.3.1 Метрики размера программы

Данные метрики представляют собой подсчет количества строк программного кода, а также числа строк с комментариями. Вычислим метрики размера программы для приведенного выше программного кода. Результаты представлены в таблице 14.

Таблица 14 – Метрики размера программы

Метрика	Количество
Количество строк кода	42
Строки-комментарии	7

Также вычислим процент комментариев. Он равен (7/42)*100=16,67. Это свидетельствует о достаточном количестве комментариев в коде для его хорошего понимания при отсутствии избыточных комментариев, переполняющих программу.

4.3.2 Метрики Холстеда

Для вычисления метрик Холстеда необходимо измерить четыре характеристики программы [28 – 30]:

– число уникальных операторов;

– число уникальных операндов;

– общее число операторов;

– общее число операндов.

С целью вычисления этих характеристик были составлен словарь операторов (таблица 15) и словарь операндов (таблица 16).

Таблица 15 – Словарь операторов

Операторы	Число вхождений
=	14
!=	1
if	3
else	3
{}	8
()	1
+	2
;	18
var	4
new	6
.	3
sqlCommand	6
ExecuteNonQuery	4
Hide	1
n1 = 16	Nl = 76

Таблица 16 – Словарь операндов

Операнды	Число вхождений
sqlExpression	12
sqlCommand	12
QuestionId	6
sc	6
radioButton1	2
$"select ID from Question where QuestionNumber='{qn}'"	1
$"select*from PerformanceOnQuestions where UserID=39 and QuestionID='{QuestionId}'"	1
$"update PerformanceOnQuestions set CorrectAnswer=1 where UserID=39 and QuestionID='{QuestionId}'"	1
$"update PerformanceOnQuestions set CorrectAnswer=0 where UserID=39 and QuestionID='{QuestionId}'"	1
$"insert PerformanceOnQuestions(CorrectAnswer,UserID,QuestionID)"	1
$" values (1,39,'{QuestionId}')"	1
$"insert PerformanceOnQuestions(CorrectAnswer,UserID,QuestionID)"	1
$" values (0,39,'{QuestionId}')"	1

Продолжение таблицы 16

Операнды	Число вхождений
null	1
n2 = 14	N2 = 47

Таким образом, число уникальных операторов программы $n1 = 14$, число уникальных операндов программы $n2 = 14$, общее число операторов в программе $N1 = 76$, общее число операндов в программе $N2 = 47$.

На основе метрик Холстеда можно вычислить целый ряд характеристик программы. Рассмотрим некоторые из них.

Расчёт словаря программы производится по формуле:

$$n = n1 + n2 = 14 + 4 = 28,$$

где n1 – количество операторов в программе;

n2 – количество операндов программе.

Расчёт длины программы производится по формуле:

$$N = N1 + N2 = 76 + 19 = 90,$$

где N1 – число вхождений всех операторов в программе;

N2 – число вхождений всех операндов программе.

Расчёт объёма программы производится по формуле:

$$V = N * log_2 n = 90 * log_2 28 = 432,$$

где N – длина программы;

n – словарь программы.

Расчёт сложности программы производится по формуле:

$$D = (n1/2) * (N2/n2) = (13 / 2) * (47 / 14) = 21{,}82,$$

где n1 – количество операторов в программе;

n2 – количество операндов программе;

N2 – число вхождений всех операндов программе.

Расчёт оценки интеллектуальных усилий производится по формуле:

$$E = D * V = 21{,}82 * 432 = 9426{,}24,$$

где D – сложность программы;

V – объем программы.

Полученные характеристики показывают, что программный код имеет достаточно высокий уровень качества и удобства для дальнейшей поддержки.

Заключение

В ходе выполнения выпускной квалификационной работы было разработано мобильное программное приложение для поддержки процессов функционирования автошколы.

На этапе анализа требований и предметной области был выполнен анализ программных продуктов аналогичного назначения, в результате которого были выявлены их основные достоинства и недостатки, определены функциональные и нефункциональные требования к создаваемому приложению и произведён анализ целевой аудитории.

Было осуществлено планирование разработки приложения и оценка бюджета. Результаты планирования показали, что общая длительность проекта составляет 57 дней, а предполагаемые затраты на реализацию – 429000 рублей, что свидетельствует об экономической целесообразности данного программного проекта.

На этапе проектирования была составлена логическая модель базы данных и разработана архитектура мобильного приложения. База данных была создана с использованием языка программирования SQL и развёрнута под управлением СУБД Microsoft SQL Server 2019. Серверная часть приложения была разработана на языке C# с использованием среды программирования Microsoft Visual Studio 2022. Клиентская часть была разработана на языке C# с использованием среды программирования Microsoft Visual Studio 2022.

Было выполнено тестирование серверной и клиентской частей разрабатываемого приложения. Оно показало корректность его работы. Также рассчитаны метрики программного кода для оценки его качества.

В дальнейшем планируется опубликовать разработанное приложение в магазине мобильных приложений для системы Android Google Play и развернуть его серверную часть на облачном сервисе, а также создать версию мобильного приложения для операционной системы iOS.

Таким образом, в ходе выполнения выпускной квалификационной работы были успешно выполнены все поставленные задачи.

Список использованных источников

1. Распоряжение Президента Российской Федерации от 9 мая 2017 г. № 203 «Стратегия развития информационного общества в Российской Федерации на 2017 – 2030 годы». [Электронный ресурс]. URL: http://zmedu.hostedu.ru/files/ykaz_7668.pdf (дата обращения: 24.04.2024).

2. Программа «Цифровая экономика Российской Федерации». Утверждена распоряжением Правительства Российской Федерации от 28 июля 2017 г. № 1632–р. [Электронный ресурс]. URL: http://static. government.ru/media/files/9gFM4FHj4PsB79I5v7yLVuPgu4bvR7M0.pdf (дата обращения: 24.04.2024).

3. Официальный сайт приложения «Автошкола Контроль» [Электронный ресурс]. URL: https://dscontrol.ru/about/ (дата обращения 06.05.2024)

4. Официальный сайт приложения «Дром ПДД» [Электронный ресурс]. URL: https://www.drom.ru/pdd/ (дата обращения 06.05.2024)

5. Официальный сайт приложения «Автошкола Реал» [Электронный ресурс]. URL:https://real-avto.com (дата обращения 06.05.2024)

6. В России за год выдано 1,37 млн водительских удостоверений // Комерсантъ [Электронный ресурс]. URL: https://www.kommersant.ru/ doc/6533604 (дата обращения 06.05.2024)

7. Mobile Operating System Market Share Worldwide // StatCounter [Электронный ресурс]. URL: https://gs.statcounter.com/os-market-share/mobile/worldwide (дата обращения: 15.05.2024).

8. Statistics about the distribution of Android versions // Composables [Электронный ресурс]. URL: https://www.composables.com/tools/distribution-chart (дата обращения 29.04.2024)

9. Приказ Министерства образования и науки РФ от 26 декабря 2013 г. № 1408 «Об утверждении примерных программ профессионального обучения водителей транспортных средств соответствующих категорий и подкатегорий»

[Электронный ресурс]. URL: https://normativ.kontur.ru /document?moduleId=1&documentId=303676 (дата обращения: 24.04.2024).

10. Ехлаков, Ю.П. Управление программными проектами. Стандарты, модели: Учебное пособие / Ю.П. Ехлаков. – СПб.: Лань, 2019. – 244 с.

11. Иванов, П.В. Управление проектами: Учебное пособие / П.В. Иванов. – Рн/Д: Феникс, 2018. –102 с.

12. Гонтарева, И.В. Управление проектами / И.В. Гонтарева, Р.М. Нижегородцев, Д.А. Новиков. – М.: КД Либроком, 2018. – 384 с.

13. Ньютон, Р. Управление проектами от А до Я / Р. Ньютон. – М.: Альпина Паблишер, 2018. – 180 с.

14. Вольфсон, Б.Л. Гибкое управление проектами и продуктами / Б.Л. Вольфсон. СПб.: Питер, 2019. – 208 с.

15. Скороход, С.В. Управление проектами средствами Microsoft Project. Курс лекций. / С.В. Скороход. – М.: Интуит НОУ, 2016. – 319 стр.

16. Унифицированный процесс Rational (RUP) // Интернет–технологии.ру [Электронный ресурс]. – URL: https://www.internet-technologies.ru/articles/newbie/unificirovannyy-process-rational-rup.html (дата обращения: 25.05.2024).

17. Арлоу, Д. UML 2 и Унифицированный процесс. Практический объектно–ориентированный анализ и проектирование, 2–е издание / Д. Арлоу, А. Нейштадт – СПб: Символ Плюс, 2007. – 624 с.

18. Концептуальная модель. // Просто о сложном [Электронный ресурс]. URL: https://mlkv.ru/samplepage/материалы–по–базам–данных/ концептуальная–модель/ (дата обращения 29.04.2024)

19. Логическая модель базы данных. // Studopedia [Электронный ресурс]. URL: https://studopedia.ru/6_1895_logicheskaya–model–bazi–dannih.html (дата обращения 06.05.2024)

20. Шаблон Model-View-Controller. [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/aspnet/core/mvc (дата обращения: 05.05.2024).

21. Физическая модель базы данных. // Studopedia [Электронный ресурс]. URL: https://studopedia.ru/6_1898_fizicheskaya-model-bazi-dannih.html (дата обращения 06.05.2024)

22. Е.А. Кольчугина. Базы данных. Методические указания к лабораторным работам. / Е.А. Кольчугина, С В. Самуйлов, С.В. Шибанов. Базы данных. Методические указания к лабораторным работам – Пенза: издательство ПГУ, 2000. – 31 с

23. Введение в Web API // Metanit.com [Электронный ресурс]. URL: https://metanit.com/sharp/mvc/12.1.php (дата обращения: 25.05.2024).

24. Андерсон Р., Ларкин К., Уоссон М. Создание веб–API с помощью ASP.NET Core и Visula Studio для Windows // Microsoft [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/first-web-api?view=aspnetcore-3.1&tabs=visual-studio (дата обращения: 25.05.2024).

25. Руководство по Entity Framework // Metanit.com [Электронный ресурс]. URL: https://metanit.com/sharp/entityframework/2.4.php (дата обращения: 25.05.2021).

26. Database First // Microsoft [Электронный ресурс]. URL: https:// docs.microsoft.com/ru-ru/ef/ef6/modeling/designer/workflows/database-first (дата обращения: 25.05.2021).

27. Entity Framework 6. Использование Database First // Professor Web [Электронный ресурс]. URL: https://professorweb.ru/my/entity-framework/6/level1/1_6.php (дата обращения: 25.05.2024).

28. Программный код и его метрики // Habr.com [Электронный ресурс]. URL: https://habr.com/ru/company/intel/blog/106082/ (дата обращения: 25.05.2024).

29. Аверьянов, А.В. Применение метрик Холстеда для количественного оценивания характеристик программ ЭВМ / А.В. Аверьянов, И.Н. Кошель, В.В. Кузнецов // Известия высших учебных заведений. Приборостроение. – 2019. – №11. – С. 970 – 975.

30. Полякова, Э.Р. Исследование программного обеспечения встраиваемой вычислительной системы метриками Холстеда / Э.Р. Полякова, А.А. Поляков, А.Л.Шустов, Д.Г. Придачкин // Cloud of Science. – 2020. – №4.– С. 950 – 959.

31. Шаблон Model-View-ViewModel. [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/xamarin/xamarin-forms/enterprise-application-patterns/mvvm (дата обращения: 04.05.2024).

Приложение А

(справочное)

Глоссарий

Android – мобильная операционная система для смартфонов, планшетов и других мобильных устройств, разрабатываемая и выпускаемая американской компанией Google и используемая на подавляющем большинстве современных мобильных устройств.

API (Application Programming Interface) – набор готовых классов, функций, процедур, констант и структур, предоставляемых приложением или операционной системой для использования во внешних программных продуктах.

RUP (Rational Unified Process) – методология разработки программного обеспечения, в основе которой лежит итеративная модель разработки.

SQL (Structured Query Language) – декларативный язык программирования, применяемый для создания и модификации данных в реляционной базе данных, работающей под управлением соответствующей системой управления базами данных, а также для управления этими данными.

HTTP (HyperText Transfer Protocol) – протокол прикладного уровня передачи данных, основанный на технологии «клиент–сервер».

UML (Unified Modeling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес–процессов, системного проектирования и отображения организационных структур.

.NET MAUI (.NET Multi–Platform App UI) – это кросс–платформенная платформа для создания собственных мобильных и классических приложений с помощью C# и XAML.

Entity Framework – представляет специальную объектно–ориентированную технологию на базе фреймворка .NET для работы с данными, обладает высоким уровенем абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища.

Приложение Б

(рекомендуемое)

Исходный код скриптов базы данных

Листинг Б.1 – Исходный код скрипта создания таблиц базы данных

```
CREATE TABLE Cadet (
  ID              int IDENTITY NOT NULL,
  ClassesAttended int NULL,
  UserID          int NOT NULL,
  PRIMARY KEY (ID));
CREATE TABLE Classes (
  ID        int IDENTITY NOT NULL,
  ClassTime datetime NOT NULL,
  PRIMARY KEY (ID));
CREATE TABLE Instructor (
  ID               int IDENTITY NOT NULL,
  InstructorAutoID int NOT NULL,
  UserID           int NOT NULL,
  PRIMARY KEY (ID));
CREATE TABLE InstructorAuto (
  ID           int IDENTITY NOT NULL,
  Brand        varchar(40) NOT NULL,
  Model        varchar(40) NOT NULL,
  LicensePlate varchar(9) NOT NULL UNIQUE,
  PRIMARY KEY (ID));
CREATE TABLE PerformanceOnQuestions (
  ID            int IDENTITY NOT NULL,
  CorrectAnswer bit NOT NULL,
  UserID        int NOT NULL,
  QuestionID    int NOT NULL,
  PRIMARY KEY (ID));
CREATE TABLE PerformanceOnTickets (
  ID            int IDENTITY NOT NULL,
  CorrectAnswer bit NOT NULL,
  UserID        int NOT NULL,
  TicketID      int NOT NULL,
  PRIMARY KEY (ID));
CREATE TABLE Question (
  ID                    int IDENTITY NOT NULL,
  QuestionNumber        int NOT NULL UNIQUE,
  Formulation           varchar(255) NOT NULL,
  CorrectAnswer         varchar(255) NOT NULL,
  FirstIncorrectAnswer  varchar(255) NOT NULL,
  SecondIncorrectAnswer varchar(255) NOT NULL,
  ThirdIncorrectAnswer  varchar(255) NOT NULL,
  PRIMARY KEY (ID));
CREATE TABLE Role (
  ID       int IDENTITY NOT NULL,
  RoleName varchar(40) NOT NULL UNIQUE,
  PRIMARY KEY (ID));
CREATE TABLE Schedule (
  ID           int IDENTITY NOT NULL,
  [Date]       datetime NOT NULL,
  InstructorID int NOT NULL,
  ClassesID    int NOT NULL,
  PRIMARY KEY (ID));
CREATE TABLE SignUpForDriving (
  ID         int IDENTITY NOT NULL,
  ScheduleID int NOT NULL,
  UserID     int NOT NULL,
  PRIMARY KEY (ID));
```

```
CREATE TABLE Ticket (
  ID           int IDENTITY NOT NULL,
  TicketNumber int NOT NULL UNIQUE,
  Question1    int NOT NULL,
  Question2    int NOT NULL,
  Question3    int NOT NULL,
  Question4    int NOT NULL,
  Question5    int NOT NULL,
  Question6    int NOT NULL,
  Question7    int NOT NULL,
  Question8    int NOT NULL,
  Question9    int NOT NULL,
  Question10   int NOT NULL,
  Question11   int NOT NULL,
  Question12   int NOT NULL,
  Question13   int NOT NULL,
  Question14   int NOT NULL,
  Question15   int NOT NULL,
  Question16   int NOT NULL,
  Question17   int NOT NULL,
  Question18   int NOT NULL,
  Question19   int NOT NULL,
  Question20   int NOT NULL,
  PRIMARY KEY (ID));
CREATE TABLE [User] (
  ID         int IDENTITY NOT NULL,
  Login      varchar(40) NOT NULL UNIQUE,
  Password   varchar(40) NOT NULL,
  FirstName  varchar(40) NOT NULL,
  SecondName varchar(40) NOT NULL,
  MiddleName varchar(40) NULL,
  RoleID     int NOT NULL,
  PRIMARY KEY (ID));
CREATE TABLE User_PerformanceOnQuestions (
  UserID                   int NOT NULL,
  PerformanceOnQuestionsID int NOT NULL,
  PRIMARY KEY (UserID,
  PerformanceOnQuestionsID));
CREATE TABLE User_PerformanceOnTickets (
  UserID                 int NOT NULL,
  PerformanceOnTicketsID int NOT NULL,
  PRIMARY KEY (UserID,
  PerformanceOnTicketsID));
CREATE UNIQUE INDEX Cadet_ID
  ON Cadet (ID);
CREATE UNIQUE INDEX Classes_ID
  ON Classes (ID);
CREATE UNIQUE INDEX Instructor_ID
  ON Instructor (ID);
CREATE UNIQUE INDEX InstructorAuto_ID
  ON InstructorAuto (ID);
CREATE UNIQUE INDEX PerformanceOnQuestions_ID
  ON PerformanceOnQuestions (ID);
CREATE UNIQUE INDEX PerformanceOnTickets_ID
  ON PerformanceOnTickets (ID);
CREATE UNIQUE INDEX Question_ID
  ON Question (ID);
CREATE UNIQUE INDEX Role_ID
```

```
  ON Role (ID);
CREATE UNIQUE INDEX Schedule_ID
  ON Schedule (ID);
CREATE UNIQUE INDEX SignUpForDriving_ID
  ON SignUpForDriving (ID);
CREATE UNIQUE INDEX Ticket_ID
  ON Ticket (ID);
CREATE UNIQUE INDEX User_ID
  ON [User] (ID);
ALTER TABLE Schedule ADD CONSTRAINT FKSchedule700162 FOREIGN KEY
(InstructorID) REFERENCES Instructor (ID);
ALTER TABLE SignUpForDriving ADD CONSTRAINT FKSignUpForD46785 FOREIGN KEY
(ScheduleID) REFERENCES Schedule (ID);
ALTER TABLE PerformanceOnTickets ADD CONSTRAINT FKPerformanc752572
FOREIGN KEY (TicketID) REFERENCES Ticket (ID);
ALTER TABLE PerformanceOnQuestions ADD CONSTRAINT FKPerformanc789912
FOREIGN KEY (QuestionID) REFERENCES Question (ID);
ALTER TABLE Instructor ADD CONSTRAINT FKInstructor345072 FOREIGN KEY
(InstructorAutoID) REFERENCES InstructorAuto (ID);
ALTER TABLE Cadet ADD CONSTRAINT FKCadet652147 FOREIGN KEY (UserID)
REFERENCES [User] (ID);
ALTER TABLE Instructor ADD CONSTRAINT FKInstructor151216 FOREIGN KEY
(UserID) REFERENCES [User] (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558631 FOREIGN KEY (Question20)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558653 FOREIGN KEY (Question19)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558654 FOREIGN KEY (Question18)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558655 FOREIGN KEY (Question17)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558656 FOREIGN KEY (Question16)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558657 FOREIGN KEY (Question15)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558658 FOREIGN KEY (Question14)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558659 FOREIGN KEY (Question13)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558660 FOREIGN KEY (Question12)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558661 FOREIGN KEY (Question11)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket558662 FOREIGN KEY (Question10)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket384857 FOREIGN KEY (Question9)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket384856 FOREIGN KEY (Question8)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket384855 FOREIGN KEY (Question7)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket384854 FOREIGN KEY (Question6)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket384853 FOREIGN KEY (Question5)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket384852 FOREIGN KEY (Question4)
REFERENCES Question (ID);
```

```
ALTER TABLE Ticket ADD CONSTRAINT FKTicket384851 FOREIGN KEY (Question3)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket384850 FOREIGN KEY (Question2)
REFERENCES Question (ID);
ALTER TABLE Ticket ADD CONSTRAINT FKTicket384849 FOREIGN KEY (Question1)
REFERENCES Question (ID);
ALTER TABLE SignUpForDriving ADD CONSTRAINT FKSignUpForD603085 FOREIGN
KEY (UserID) REFERENCES [User] (ID);
ALTER TABLE [User] ADD CONSTRAINT FKUser349791 FOREIGN KEY (RoleID)
REFERENCES Role (ID);
ALTER TABLE Schedule ADD CONSTRAINT FKSchedule187494 FOREIGN KEY
(ClassesID) REFERENCES Classes (ID);
ALTER TABLE User_PerformanceOnTickets ADD CONSTRAINT FKUser_Perfo184059
FOREIGN KEY (UserID) REFERENCES [User] (ID);
ALTER TABLE User_PerformanceOnTickets ADD CONSTRAINT FKUser_Perfo9023
FOREIGN KEY (PerformanceOnTicketsID) REFERENCES PerformanceOnTickets (ID);
ALTER TABLE User_PerformanceOnQuestions ADD CONSTRAINT FKUser_Perfo790577
FOREIGN KEY (UserID) REFERENCES [User] (ID);
ALTER TABLE User_PerformanceOnQuestions ADD CONSTRAINT FKUser_Perfo101733
FOREIGN KEY (PerformanceOnQuestionsID) REFERENCES PerformanceOnQuestions
(ID);
```

Приложение В

(рекомендуемое)

Исходный код серверной части

Листинг В.1 – Файл DrivingDbContext.cs

```
using System;
using System.Collections.Generic;
using Microsoft.EntityFrameworkCore;

namespace AutoSchoolApp.API.Models;

public partial class DrivingDbContext : DbContext
{
    public DrivingDbContext()
    {
    }

    public DrivingDbContext(DbContextOptions<DrivingDbContext> options)
        : base(options)
    {
    }

    public virtual DbSet<AllInfoAboutUser> AllInfoAboutUsers { get; set; }

    public virtual DbSet<Cadet> Cadets { get; set; }

    public virtual DbSet<Class> Classes { get; set; }

    public virtual DbSet<Fiocadet> Fiocadets { get; set; }

    public virtual DbSet<Fioinstructor> Fioinstructors { get; set; }

    public virtual DbSet<FullSchedule> FullSchedules { get; set; }

    public virtual DbSet<Instructor> Instructors { get; set; }

    public virtual DbSet<InstructorAuto> InstructorAutos { get; set; }

    public virtual DbSet<InstructorsAndCar> InstructorsAndCars { get;
set; }

    public virtual DbSet<NumCorectAnswersOnQuestionsPerCadet>
NumCorectAnswersOnQuestionsPerCadets { get; set; }

    public virtual DbSet<NumCorectAnswersOnQuestionsPerTheme>
NumCorectAnswersOnQuestionsPerThemes { get; set; }

    public virtual DbSet<NumCorectAnswersOnTicketsPerDate>
NumCorectAnswersOnTicketsPerDates { get; set; }

    public virtual DbSet<NumPassCadetsToExamPerInstructor>
NumPassCadetsToExamPerInstructors { get; set; }

    public virtual DbSet<PerformanceOnQuestion> PerformanceOnQuestions
{ get; set; }

    public virtual DbSet<PerformanceOnTicket> PerformanceOnTickets { get;
set; }

    public virtual DbSet<Question> Questions { get; set; }
```

```
    public virtual DbSet<Role> Roles { get; set; }

    public virtual DbSet<Schedule> Schedules { get; set; }

    public virtual DbSet<SignUpForDriving> SignUpForDrivings { get; set; }

    public virtual DbSet<Ticket> Tickets { get; set; }

    public virtual DbSet<User> Users { get; set; }

    protected override void OnConfiguring(DbContextOptionsBuilder
optionsBuilder)
#warning To protect potentially sensitive information in your connection
string, you should move it out of source code. You can avoid scaffolding
the connection string by using the Name= syntax to read it from
configuration - see https://go.microsoft.com/fwlink/?linkid=2131148. For
more guidance on storing connection strings, see
https://go.microsoft.com/fwlink/?LinkId=723263.
        => optionsBuilder.UseSqlServer("Server=DESKTOP-
KBQVOJR;Database=drivingDB;Trusted_Connection=True;TrustServerCertificate
=True;encrypt=false");

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<AllInfoAboutUser>(entity =>
        {
            entity
                .HasNoKey()
                .ToView("allInfoAboutUsers");

            entity.Property(e =>
e.ИдентификаторПользователя).HasColumnName("Идентификатор пользователя");
            entity.Property(e => e.Имя)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.Логин)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.Отчество)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.Пароль)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.Роль)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.Фамилия)
                .HasMaxLength(40)
                .IsUnicode(false);
        });

        modelBuilder.Entity<Cadet>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__Cadet__3214EC27FA845FF5");

            entity.ToTable("Cadet");
```

```
            entity.HasIndex(e => e.Id, "Cadet_ID").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
            entity.Property(e => e.UserId).HasColumnName("UserID");

            entity.HasOne(d => d.User).WithMany(p => p.Cadets)
                .HasForeignKey(d => d.UserId)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKCadet652147");
        });

        modelBuilder.Entity<Class>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__Classes__3214EC27CD35C4A2");

            entity.HasIndex(e => e.Id, "Classes_ID").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
            entity.Property(e => e.ClassTime).HasPrecision(0);
        });

        modelBuilder.Entity<Fiocadet>(entity =>
        {
            entity
                .HasNoKey()
                .ToView("FIOCadet");

            entity.Property(e => e.IdCadet).HasColumnName("Id_Cadet");
            entity.Property(e => e.Фио)
                .HasMaxLength(122)
                .IsUnicode(false)
                .HasColumnName("ФИО");
        });

        modelBuilder.Entity<Fioinstructor>(entity =>
        {
            entity
                .HasNoKey()
                .ToView("FIOInstructors");

            entity.Property(e =>
e.IdInstructor).HasColumnName("Id_Instructor");
            entity.Property(e => e.Фио)
                .HasMaxLength(122)
                .IsUnicode(false)
                .HasColumnName("ФИО");
        });

        modelBuilder.Entity<FullSchedule>(entity =>
        {
            entity
                .HasNoKey()
                .ToView("FullSchedule");

            entity.Property(e => e.CadetId).HasColumnName("CadetID");
```

```
            entity.Property(e =>
e.InstructorId).HasColumnName("InstructorID");
            entity.Property(e => e.Время).HasMaxLength(4000);
            entity.Property(e => e.Дата).HasMaxLength(4000);
            entity.Property(e => e.Инструктор)
                .HasMaxLength(122)
                .IsUnicode(false);
            entity.Property(e => e.Курсант)
                .HasMaxLength(122)
                .IsUnicode(false);
        });

        modelBuilder.Entity<Instructor>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__Instruct__3214EC27E52CAA54");

            entity.ToTable("Instructor");

            entity.HasIndex(e => e.Id, "Instructor_ID").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
            entity.Property(e =>
e.InstructorAutoId).HasColumnName("InstructorAutoID");
            entity.Property(e => e.UserId).HasColumnName("UserID");

            entity.HasOne(d => d.InstructorAuto).WithMany(p =>
p.Instructors)
                .HasForeignKey(d => d.InstructorAutoId)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKInstructor345072");

            entity.HasOne(d => d.User).WithMany(p => p.Instructors)
                .HasForeignKey(d => d.UserId)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKInstructor151216");
        });

        modelBuilder.Entity<InstructorAuto>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__Instruct__3214EC27DC9DDB1D");

            entity.ToTable("InstructorAuto", tb =>
tb.HasTrigger("DelAuto"));

            entity.HasIndex(e => e.Id, "InstructorAuto_ID").IsUnique();

            entity.HasIndex(e => e.LicensePlate,
"UQ__Instruct__026BC15CE21AFB37").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
            entity.Property(e => e.Brand)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.LicensePlate)
                .HasMaxLength(9)
                .IsUnicode(false);
```

```
            entity.Property(e => e.Model)
                .HasMaxLength(40)
                .IsUnicode(false);
        });

        modelBuilder.Entity<InstructorsAndCar>(entity =>
        {
            entity
                .HasNoKey()
                .ToView("InstructorsAndCars");

            entity.Property(e => e.Имя)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.Марка)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.Модель)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.Отчество)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.РегистрационныйНомер)
                .HasMaxLength(9)
                .IsUnicode(false)
                .HasColumnName("Регистрационный номер");
            entity.Property(e => e.Фамилия)
                .HasMaxLength(40)
                .IsUnicode(false);
        });

        modelBuilder.Entity<NumCorectAnswersOnQuestionsPerCadet>(entity
=>
        {
            entity
                .HasNoKey()
                .ToView("NumCorectAnswersOnQuestionsPerCadet");

            entity.Property(e =>
e.КолВоПравильноРешённыхВопросов).HasColumnName("Кол-во правильно
решённых вопросов");
            entity.Property(e => e.Курсант)
                .HasMaxLength(122)
                .IsUnicode(false);
        });

        modelBuilder.Entity<NumCorectAnswersOnQuestionsPerTheme>(entity
=>
        {
            entity
                .HasNoKey()
                .ToView("NumCorectAnswersOnQuestionsPerTheme");

            entity.Property(e =>
e.КолВоПравильноРешённыхВопросов).HasColumnName("Кол-во правильно
решённых вопросов");
            entity.Property(e => e.Тема)
```

```
                .HasMaxLength(255)
                .IsUnicode(false);
        });

        modelBuilder.Entity<NumCorectAnswersOnTicketsPerDate>(entity =>
        {
            entity
                .HasNoKey()
                .ToView("NumCorectAnswersOnTicketsPerDate");

            entity.Property(e =>
e.КолВоПравильноРешённыхБилетовов).HasColumnName("Кол-во правильно
решённых билетовов");
        });

        modelBuilder.Entity<NumPassCadetsToExamPerInstructor>(entity =>
        {
            entity
                .HasNoKey()
                .ToView("NumPassCadetsToExamPerInstructor");

            entity.Property(e => e.Инструктор)
                .HasMaxLength(122)
                .IsUnicode(false);
            entity.Property(e =>
e.КолВоКурсантовДопущенныхКПрактическомуЭкзамену).HasColumnName("Кол-во
курсантов допущенных к практическому экзамену");
        });

        modelBuilder.Entity<PerformanceOnQuestion>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__Performa__3214EC2798929F43");

            entity.HasIndex(e => e.Id,
"PerformanceOnQuestions_ID").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
            entity.Property(e =>
e.QuestionId).HasColumnName("QuestionID");
            entity.Property(e => e.UserId).HasColumnName("UserID");

            entity.HasOne(d => d.Question).WithMany(p =>
p.PerformanceOnQuestions)
                .HasForeignKey(d => d.QuestionId)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKPerformanc789912");
        });

        modelBuilder.Entity<PerformanceOnTicket>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__Performa__3214EC274B7880E5");

            entity.HasIndex(e => e.Id,
"PerformanceOnTickets_ID").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
```

```
            entity.Property(e => e.TicketId).HasColumnName("TicketID");
            entity.Property(e => e.UserId).HasColumnName("UserID");

            entity.HasOne(d => d.Ticket).WithMany(p =>
p.PerformanceOnTickets)
                .HasForeignKey(d => d.TicketId)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKPerformanc752572");
        });

        modelBuilder.Entity<Question>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__Question__3214EC27F6A918D7");

            entity.ToTable("Question");

            entity.HasIndex(e => e.Id, "Question_ID").IsUnique();

            entity.HasIndex(e => e.QuestionNumber,
"UQ__Question__3A469287CA0C1861").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
            entity.Property(e => e.CorrectAnswer)
                .HasMaxLength(512)
                .IsUnicode(false);
            entity.Property(e => e.FirstIncorrectAnswer)
                .HasMaxLength(512)
                .IsUnicode(false);
            entity.Property(e => e.Formulation)
                .HasMaxLength(512)
                .IsUnicode(false);
            entity.Property(e => e.SecondIncorrectAnswer)
                .HasMaxLength(512)
                .IsUnicode(false);
            entity.Property(e => e.Theme)
                .HasMaxLength(255)
                .IsUnicode(false);
            entity.Property(e => e.ThirdIncorrectAnswer)
                .HasMaxLength(512)
                .IsUnicode(false);
        });

        modelBuilder.Entity<Role>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__Role__3214EC27F7AE08BF");

            entity.ToTable("Role", tb => tb.HasTrigger("InsertRole"));

            entity.HasIndex(e => e.Id, "Role_ID").IsUnique();

            entity.HasIndex(e => e.RoleName,
"UQ__Role__8A2B6160E9D19E3F").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
            entity.Property(e => e.RoleName)
                .HasMaxLength(40)
```

```
                .IsUnicode(false);
        });

        modelBuilder.Entity<Schedule>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__Schedule__3214EC27E3440114");

            entity.ToTable("Schedule");

            entity.HasIndex(e => e.Id, "Schedule_ID").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
            entity.Property(e => e.ClassesId).HasColumnName("ClassesID");
            entity.Property(e =>
e.InstructorId).HasColumnName("InstructorID");

            entity.HasOne(d => d.Classes).WithMany(p => p.Schedules)
                .HasForeignKey(d => d.ClassesId)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKSchedule187494");

            entity.HasOne(d => d.Instructor).WithMany(p => p.Schedules)
                .HasForeignKey(d => d.InstructorId)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKSchedule700162");
        });

        modelBuilder.Entity<SignUpForDriving>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__SignUpFo__3214EC27CCAD9775");

            entity.ToTable("SignUpForDriving", tb =>
                {
                    tb.HasTrigger("OnNewSignNotification");
                    tb.HasTrigger("OnRemoveSignNotification");
                    tb.HasTrigger("OnRemoveSignUpForDriving");
                });

            entity.HasIndex(e => e.Id, "SignUpForDriving_ID").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
            entity.Property(e =>
e.ScheduleId).HasColumnName("ScheduleID");
            entity.Property(e => e.UserId).HasColumnName("UserID");

            entity.HasOne(d => d.Schedule).WithMany(p =>
p.SignUpForDrivings)
                .HasForeignKey(d => d.ScheduleId)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKSignUpForD46785");

            entity.HasOne(d => d.User).WithMany(p => p.SignUpForDrivings)
                .HasForeignKey(d => d.UserId)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKSignUpForD603085");
        });
```

```
        modelBuilder.Entity<Ticket>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__Ticket__3214EC272BE2A347");

            entity.ToTable("Ticket");

            entity.HasIndex(e => e.Id, "Ticket_ID").IsUnique();

            entity.HasIndex(e => e.TicketNumber,
"UQ__Ticket__CBED06DACAAC61EE").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");

            entity.HasOne(d => d.Question1Navigation).WithMany(p =>
p.TicketQuestion1Navigations)
                .HasForeignKey(d => d.Question1)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket384849");

            entity.HasOne(d => d.Question10Navigation).WithMany(p =>
p.TicketQuestion10Navigations)
                .HasForeignKey(d => d.Question10)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558662");

            entity.HasOne(d => d.Question11Navigation).WithMany(p =>
p.TicketQuestion11Navigations)
                .HasForeignKey(d => d.Question11)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558661");

            entity.HasOne(d => d.Question12Navigation).WithMany(p =>
p.TicketQuestion12Navigations)
                .HasForeignKey(d => d.Question12)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558660");

            entity.HasOne(d => d.Question13Navigation).WithMany(p =>
p.TicketQuestion13Navigations)
                .HasForeignKey(d => d.Question13)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558659");

            entity.HasOne(d => d.Question14Navigation).WithMany(p =>
p.TicketQuestion14Navigations)
                .HasForeignKey(d => d.Question14)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558658");

            entity.HasOne(d => d.Question15Navigation).WithMany(p =>
p.TicketQuestion15Navigations)
                .HasForeignKey(d => d.Question15)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558657");
```

```
            entity.HasOne(d => d.Question16Navigation).WithMany(p =>
p.TicketQuestion16Navigations)
                .HasForeignKey(d => d.Question16)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558656");

            entity.HasOne(d => d.Question17Navigation).WithMany(p =>
p.TicketQuestion17Navigations)
                .HasForeignKey(d => d.Question17)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558655");

            entity.HasOne(d => d.Question18Navigation).WithMany(p =>
p.TicketQuestion18Navigations)
                .HasForeignKey(d => d.Question18)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558654");

            entity.HasOne(d => d.Question19Navigation).WithMany(p =>
p.TicketQuestion19Navigations)
                .HasForeignKey(d => d.Question19)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558653");

            entity.HasOne(d => d.Question2Navigation).WithMany(p =>
p.TicketQuestion2Navigations)
                .HasForeignKey(d => d.Question2)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket384850");

            entity.HasOne(d => d.Question20Navigation).WithMany(p =>
p.TicketQuestion20Navigations)
                .HasForeignKey(d => d.Question20)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket558631");

            entity.HasOne(d => d.Question3Navigation).WithMany(p =>
p.TicketQuestion3Navigations)
                .HasForeignKey(d => d.Question3)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket384851");

            entity.HasOne(d => d.Question4Navigation).WithMany(p =>
p.TicketQuestion4Navigations)
                .HasForeignKey(d => d.Question4)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket384852");

            entity.HasOne(d => d.Question5Navigation).WithMany(p =>
p.TicketQuestion5Navigations)
                .HasForeignKey(d => d.Question5)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket384853");

            entity.HasOne(d => d.Question6Navigation).WithMany(p =>
p.TicketQuestion6Navigations)
                .HasForeignKey(d => d.Question6)
                .OnDelete(DeleteBehavior.ClientSetNull)
```

```
                .HasConstraintName("FKTicket384854");

            entity.HasOne(d => d.Question7Navigation).WithMany(p =>
p.TicketQuestion7Navigations)
                .HasForeignKey(d => d.Question7)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket384855");

            entity.HasOne(d => d.Question8Navigation).WithMany(p =>
p.TicketQuestion8Navigations)
                .HasForeignKey(d => d.Question8)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket384856");

            entity.HasOne(d => d.Question9Navigation).WithMany(p =>
p.TicketQuestion9Navigations)
                .HasForeignKey(d => d.Question9)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKTicket384857");
        });

        modelBuilder.Entity<User>(entity =>
        {
            entity.HasKey(e =>
e.Id).HasName("PK__User__3214EC278FA85A45");

            entity.ToTable("User", tb => tb.HasTrigger("InsertCadet"));

            entity.HasIndex(e => e.Login,
"UQ__User__5E55825B54880DE1").IsUnique();

            entity.HasIndex(e => e.Id, "User_ID").IsUnique();

            entity.Property(e => e.Id).HasColumnName("ID");
            entity.Property(e => e.FirstName)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.Login)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.MiddleName)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.Password)
                .HasMaxLength(40)
                .IsUnicode(false);
            entity.Property(e => e.RoleId).HasColumnName("RoleID");
            entity.Property(e => e.SecondName)
                .HasMaxLength(40)
                .IsUnicode(false);

            entity.HasOne(d => d.Role).WithMany(p => p.Users)
                .HasForeignKey(d => d.RoleId)
                .OnDelete(DeleteBehavior.ClientSetNull)
                .HasConstraintName("FKUser349791");

            entity.HasMany(d => d.PerformanceOnQuestions).WithMany(p =>
p.Users)
```

```
                .UsingEntity<Dictionary<string, object>>(
                    "UserPerformanceOnQuestion",
                    r => r.HasOne<PerformanceOnQuestion>().WithMany()
                        .HasForeignKey("PerformanceOnQuestionsId")
                        .OnDelete(DeleteBehavior.ClientSetNull)
                        .HasConstraintName("FKUser_Perfo101733"),
                    l => l.HasOne<User>().WithMany()
                        .HasForeignKey("UserId")
                        .OnDelete(DeleteBehavior.ClientSetNull)
                        .HasConstraintName("FKUser_Perfo790577"),
                    j =>
                    {
                        j.HasKey("UserId",
"PerformanceOnQuestionsId").HasName("PK__User_Per__E8B5F59698582D25");
                        j.ToTable("User_PerformanceOnQuestions");

j.IndexerProperty<int>("UserId").HasColumnName("UserID");

j.IndexerProperty<int>("PerformanceOnQuestionsId").HasColumnName("Perform
anceOnQuestionsID");
                    });

            entity.HasMany(d => d.PerformanceOnTickets).WithMany(p =>
p.Users)
                .UsingEntity<Dictionary<string, object>>(
                    "UserPerformanceOnTicket",
                    r => r.HasOne<PerformanceOnTicket>().WithMany()
                        .HasForeignKey("PerformanceOnTicketsId")
                        .OnDelete(DeleteBehavior.ClientSetNull)
                        .HasConstraintName("FKUser_Perfo9023"),
                    l => l.HasOne<User>().WithMany()
                        .HasForeignKey("UserId")
                        .OnDelete(DeleteBehavior.ClientSetNull)
                        .HasConstraintName("FKUser_Perfo184059"),
                    j =>
                    {
                        j.HasKey("UserId",
"PerformanceOnTicketsId").HasName("PK__User_Per__C3E2EBBC3B952200");
                        j.ToTable("User_PerformanceOnTickets");

j.IndexerProperty<int>("UserId").HasColumnName("UserID");

j.IndexerProperty<int>("PerformanceOnTicketsId").HasColumnName("Performan
ceOnTicketsID");
                    });
        });

        OnModelCreatingPartial(modelBuilder);
    }

    partial void OnModelCreatingPartial(ModelBuilder modelBuilder);
}
```

Листинг В.2 – Файл AllInfoAboutUser.cs

```
using System;
using System.Collections.Generic;
namespace AutoSchoolApp.API.Models;
```

```
public partial class AllInfoAboutUser
{    public int ИдентификаторПользователя { get; set; }
    public string Логин { get; set; } = null!;
    public string Пароль { get; set; } = null!;
    public string Фамилия { get; set; } = null!;
    public string Имя { get; set; } = null!;
    public string? Отчество { get; set; }
    public string Роль { get; set; } = null!;
}
```

Листинг В.3 – Файл Cadet.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class Cadet
{
    public int Id { get; set; }

    public int? ClassesAttended { get; set; }

    public int UserId { get; set; }

    public virtual User User { get; set; } = null!;
}
```

Листинг В.4 – Файл Class.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class Class
{
    public int Id { get; set; }

    public TimeOnly? ClassTime { get; set; }

    public virtual ICollection<Schedule> Schedules { get; set; } = new
List<Schedule>();
}
```

Листинг В.5 – Файл Fiocadet.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class Fiocadet
{
    public string? Фио { get; set; }

    public int IdCadet { get; set; }
}
```

Листинг В.6 – Файл Fioinstructor.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class Fioinstructor
{
    public string? Фио { get; set; }

    public int IdInstructor { get; set; }
}
```

Листинг В.7 – Файл FullSchedule.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class FullSchedule
{
    public string? Дата { get; set; }

    public int InstructorId { get; set; }

    public string? Инструктор { get; set; }

    public int CadetId { get; set; }

    public string? Курсант { get; set; }

    public string? Время { get; set; }
}
```

Листинг В.8 – Файл Instructor.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class Instructor
{
    public int Id { get; set; }

    public int InstructorAutoId { get; set; }

    public int UserId { get; set; }

    public virtual InstructorAuto InstructorAuto { get; set; } = null!;

    public virtual ICollection<Schedule> Schedules { get; set; } = new
List<Schedule>();

    public virtual User User { get; set; } = null!;
}
```

Листинг В.9 – Файл InstructorAuto.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class InstructorAuto
{
    public int Id { get; set; }

    public string Brand { get; set; } = null!;

    public string Model { get; set; } = null!;

    public string LicensePlate { get; set; } = null!;

    public virtual ICollection<Instructor> Instructors { get; set; } =
new List<Instructor>();
}
```

Листинг В.10 – Файл InstructorsAndCar.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class InstructorsAndCar
{
    public string Фамилия { get; set; } = null!;

    public string Имя { get; set; } = null!;

    public string? Отчество { get; set; }

    public string Марка { get; set; } = null!;

    public string Модель { get; set; } = null!;

    public string РегистрационныйНомер { get; set; } = null!;
}
```

Листинг В.11 – Файл NumCorectAnswersOnQuestionsPerCadet.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class NumCorectAnswersOnQuestionsPerCadet
{
    public string? Курсант { get; set; }

    public int? КолВоПравильноРешённыхВопросов { get; set; }
}
```

Листинг В.12 – Файл NumCorectAnswersOnQuestionsPerTheme.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class NumCorectAnswersOnQuestionsPerTheme
{
    public string? Тема { get; set; }

    public int? КолВоПравильноРешённыхВопросов { get; set; }
}
```

Листинг В.13 – Файл NumPassCadetsToExamPerInstructor.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class NumPassCadetsToExamPerInstructor
{
    public string? Инструктор { get; set; }

    public int? КолВоКурсантовДопущенныхКПрактическомуЭкзамену { get;
set; }
}
```

Листинг В.14 – Файл PerformanceOnQuestion.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class PerformanceOnQuestion
{
    public int Id { get; set; }

    public bool CorrectAnswer { get; set; }

    public int UserId { get; set; }

    public int QuestionId { get; set; }

    public virtual Question Question { get; set; } = null!;

    public virtual ICollection<User> Users { get; set; } = new
List<User>();
}
```

Листинг В.15 – Файл PerformanceOnTicket.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;
```

```
public partial class PerformanceOnTicket
{
    public int Id { get; set; }

    public bool CorrectAnswer { get; set; }

    public int UserId { get; set; }

    public int TicketId { get; set; }

    public DateOnly? SolveDate { get; set; }

    public virtual Ticket Ticket { get; set; } = null!;

    public virtual ICollection<User> Users { get; set; } = new
List<User>();
}
```

Листинг В.16 – Файл Question.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class Question
{
    public int Id { get; set; }

    public int QuestionNumber { get; set; }

    public string? Formulation { get; set; }

    public string? CorrectAnswer { get; set; }

    public string? FirstIncorrectAnswer { get; set; }

    public string? SecondIncorrectAnswer { get; set; }

    public string? ThirdIncorrectAnswer { get; set; }

    public string? Theme { get; set; }

    public virtual ICollection<PerformanceOnQuestion>
PerformanceOnQuestions { get; set; } = new List<PerformanceOnQuestion>();

    public virtual ICollection<Ticket> TicketQuestion10Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion11Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion12Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion13Navigations { get;
set; } = new List<Ticket>();
```

```
    public virtual ICollection<Ticket> TicketQuestion14Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion15Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion16Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion17Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion18Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion19Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion1Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion20Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion2Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion3Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion4Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion5Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion6Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion7Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion8Navigations { get;
set; } = new List<Ticket>();

    public virtual ICollection<Ticket> TicketQuestion9Navigations { get;
set; } = new List<Ticket>();
}
```

Листинг В.17 – Файл Role.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class Role
{
```

```
    public int Id { get; set; }

    public string RoleName { get; set; } = null!;

    public virtual ICollection<User> Users { get; set; } = new
List<User>();
}
```

Листинг В.18 – Файл Schedule.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class Schedule
{
    public int Id { get; set; }

    public DateOnly? Date { get; set; }

    public int InstructorId { get; set; }

    public int ClassesId { get; set; }

    public virtual Class Classes { get; set; } = null!;

    public virtual Instructor Instructor { get; set; } = null!;

    public virtual ICollection<SignUpForDriving> SignUpForDrivings { get;
set; } = new List<SignUpForDriving>();
}
```

Листинг В.19 – Файл SignUpForDriving.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class SignUpForDriving
{
    public int Id { get; set; }

    public int ScheduleId { get; set; }

    public int UserId { get; set; }

    public virtual Schedule Schedule { get; set; } = null!;

    public virtual User User { get; set; } = null!;
}
```

Листинг В.20 – Файл Ticket.cs

```
using System;
using System.Collections.Generic;
```

```
namespace AutoSchoolApp.API.Models;

public partial class Ticket
{
    public int Id { get; set; }

    public int TicketNumber { get; set; }

    public int Question1 { get; set; }

    public int Question2 { get; set; }

    public int Question3 { get; set; }

    public int Question4 { get; set; }

    public int Question5 { get; set; }

    public int Question6 { get; set; }

    public int Question7 { get; set; }

    public int Question8 { get; set; }

    public int Question9 { get; set; }

    public int Question10 { get; set; }

    public int Question11 { get; set; }

    public int Question12 { get; set; }

    public int Question13 { get; set; }

    public int Question14 { get; set; }

    public int Question15 { get; set; }

    public int Question16 { get; set; }

    public int Question17 { get; set; }

    public int Question18 { get; set; }

    public int Question19 { get; set; }

    public int Question20 { get; set; }

    public virtual ICollection<PerformanceOnTicket> PerformanceOnTickets
{ get; set; } = new List<PerformanceOnTicket>();

    public virtual Question Question10Navigation { get; set; } = null!;

    public virtual Question Question11Navigation { get; set; } = null!;

    public virtual Question Question12Navigation { get; set; } = null!;

    public virtual Question Question13Navigation { get; set; } = null!;
```

```
    public virtual Question Question14Navigation { get; set; } = null!;

    public virtual Question Question15Navigation { get; set; } = null!;

    public virtual Question Question16Navigation { get; set; } = null!;

    public virtual Question Question17Navigation { get; set; } = null!;

    public virtual Question Question18Navigation { get; set; } = null!;

    public virtual Question Question19Navigation { get; set; } = null!;

    public virtual Question Question1Navigation { get; set; } = null!;

    public virtual Question Question20Navigation { get; set; } = null!;

    public virtual Question Question2Navigation { get; set; } = null!;

    public virtual Question Question3Navigation { get; set; } = null!;

    public virtual Question Question4Navigation { get; set; } = null!;

    public virtual Question Question5Navigation { get; set; } = null!;

    public virtual Question Question6Navigation { get; set; } = null!;

    public virtual Question Question7Navigation { get; set; } = null!;

    public virtual Question Question8Navigation { get; set; } = null!;

    public virtual Question Question9Navigation { get; set; } = null!;
}
```

Листинг В.21 – Файл User.cs

```
using System;
using System.Collections.Generic;

namespace AutoSchoolApp.API.Models;

public partial class User
{
    public int Id { get; set; }

    public string Login { get; set; } = null!;

    public string Password { get; set; } = null!;

    public string FirstName { get; set; } = null!;

    public string SecondName { get; set; } = null!;

    public string? MiddleName { get; set; }

    public int RoleId { get; set; }
```

```
    public virtual ICollection<Cadet> Cadets { get; set; } = new
List<Cadet>();

    public virtual ICollection<Instructor> Instructors { get; set; } =
new List<Instructor>();

    public virtual Role Role { get; set; } = null!;

    public virtual ICollection<SignUpForDriving> SignUpForDrivings { get;
set; } = new List<SignUpForDriving>();

    public virtual ICollection<PerformanceOnQuestion>
PerformanceOnQuestions { get; set; } = new List<PerformanceOnQuestion>();

    public virtual ICollection<PerformanceOnTicket> PerformanceOnTickets
{ get; set; } = new List<PerformanceOnTicket>();
}
```

Листинг В.22 – Файл Program.cs

```
using AutoSchoolApp.API.Models;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Options;

var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
builder.Services.AddDbContext<DrivingDbContext>(options =>
{    options.UseSqlServer(builder.Configuration.GetConnectionString("Driv
ingConnection") ?? throw new InvalidOperationException("Database
connection string 'DrivingConnection' is not found"));
});
var app = builder.Build();
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}
app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();
```

Листинг В.23 – Файл UsersController.cs

```
using AutoSchoolApp.API.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace AutoSchoolApp.API.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class UsersController : Controller
    {
        private readonly DrivingDbContext _context;
```

```
        public UsersController(DrivingDbContext context)
        {
            _context = context;
        }

        [HttpGet("GetAllUsers")]
        public async Task<ActionResult<IEnumerable<User>>> Get()
        {
            return await _context.Users.ToListAsync();
        }

        [HttpGet("SearchUserById{id}")]
        public async Task<ActionResult<User>> Get(int id)
        {
            var users = await _context.Users.ToListAsync();

            if (users == null)
            {
                return NotFound();
            }

            var user = users.Find(x => x.Id == id);

            if (user == null)
            {
                return NotFound();
            }
            return Ok(user);
        }

        [HttpGet("LogInUser{login}/{password}")]
        public async Task<ActionResult<User>> Get(string login, string
password)
        {
            var users = await _context.Users.ToListAsync();

            if (users == null)
            {
                return NotFound();
            }

            var user = users.Find(x => x.Login == login && x.Password ==
password);

            if (user == null)
            {
                return NotFound();
            }
            return Ok(user);
        }

        [HttpPut("ChangePassword")]
        public async Task<ActionResult<User>> Update(int id, string
password)
        {
            var users = await _context.Users.ToListAsync();
            var user = users.Find(x => x.Id == id);
```

```
            if (user == null)
                return BadRequest();

            if (!_context.Users.Any(x => x.Id == user.Id))
                return NotFound();

            user.Password = password;

            return Ok(user);
        }

        [HttpPost("AddUser")]
        public async Task<ActionResult<User>> Create(string login, string
password, string firstName, string secondName, string middleName, int
roleId)
        {
            User user = new()
            {
                Login = login,
                Password = password,
                FirstName = firstName,
                MiddleName = middleName,
                SecondName = secondName,
                RoleId = roleId
            };

            if (Helper.IsExists(ref user, _context.Users.ToList()))
                return Ok(user);

            _context.Users.Add(user);
            await _context.SaveChangesAsync();
            return Ok(user);
        }

        [HttpDelete("RemoveUser{id}")]
        public async Task<ActionResult<User>> Delete(int id)
        {
            var user = _context.Users.FirstOrDefault(x => x.Id == id);

            if (user == null)
            {
                return NotFound();
            }

            _context.Users.Remove(user);
            await _context.SaveChangesAsync();
            return Ok(_context.Users.ToListAsync());
        }
    }
}
```

Листинг В.24 – Файл SignUpForDrivingsController.cs

```
using AutoSchoolApp.API.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace AutoSchoolApp.API.Controllers
{
```

```
    [Route("api/[controller]")]
    [ApiController]
    public class SignUpForDrivingsController : Controller
    {
        private readonly DrivingDbContext _context;

        public SignUpForDrivingsController(DrivingDbContext context)
        {
            _context = context;
        }

        [HttpPost("MakeSignUpForDriving")]
        public async Task<ActionResult<SignUpForDriving>> Create(DateOnly
date, TimeOnly time, int cadetId, int instructorId)
        {
            var classes = await _context.Classes.ToListAsync();
            int classId = classes.Find(x => x.ClassTime == time).Id;

            Schedule schedule = new()
            {
                Date = date,
                ClassesId = classId,
                InstructorId = instructorId
            };

            SignUpForDriving signUpForDriving = new()
            {
                ScheduleId = schedule.Id,
                UserId = cadetId
            };

            if (Helper.IsExists(ref signUpForDriving,
_context.SignUpForDrivings.ToList()))
                return Ok(signUpForDriving);

            _context.SignUpForDrivings.Add(signUpForDriving);
            await _context.SaveChangesAsync();
            return Ok(signUpForDriving);
        }

        [HttpDelete("RemoveSignUpForDriving")]
        public async Task<ActionResult<SignUpForDriving>> Delete(DateOnly
date, TimeOnly time, int cadetId, int instructorId)
        {
            var delClass = _context.Classes.FirstOrDefault(x =>
x.ClassTime == time);
            var schedule = _context.Schedules.FirstOrDefault(x => x.Date
== date && x.ClassesId == delClass.Id && x.InstructorId == instructorId);

            if (schedule == null)
            {
                return NotFound();
            }

            var signUpForDriving =
_context.SignUpForDrivings.FirstOrDefault(x => x.Schedule.Id ==
schedule.Id && x.UserId == cadetId);
```

```
            if (signUpForDriving == null)
            {
                return NotFound();
            }

            _context.SignUpForDrivings.Remove(signUpForDriving);
            await _context.SaveChangesAsync();
            return Ok();
        }
    }
}
```

Листинг В.25 – Файл PerformanceOnQuestionsController.cs

```
using AutoSchoolApp.API.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace AutoSchoolApp.API.Controllers
{
    public class PerformanceOnQuestionsController : Controller
    {
        private readonly DrivingDbContext _context;

        public PerformanceOnQuestionsController(DrivingDbContext context)
        {
            _context = context;
        }

        [HttpPost("SaveAnswer")]
        public async Task<ActionResult<PerformanceOnQuestion>> Create(int
cadetId, bool correctAnswer, int questionId)
        {
            PerformanceOnQuestion performanceOnQuestion = new()
            {
                UserId = cadetId,
                CorrectAnswer = correctAnswer,
                QuestionId = questionId
            };

            if (Helper.IsExists(ref performanceOnQuestion,
_context.PerformanceOnQuestions.ToList()))
            {
                await Update(cadetId, correctAnswer, questionId);
                return Ok();
            }

            _context.PerformanceOnQuestions.Add(performanceOnQuestion);
            await _context.SaveChangesAsync();
            return Ok(performanceOnQuestion);
        }

        [HttpPut("UpdateAnswer")]
        public async Task<ActionResult<PerformanceOnQuestion>> Update(int
cadetId, bool correctAnswer, int questionId)
        {
```

```
            var Performances = await
_context.PerformanceOnQuestions.ToListAsync();
            var Performance = Performances.Find(x => x.UserId == cadetId
&& x.QuestionId == questionId);

            if (Performance == null)
                return BadRequest();

            if (!_context.PerformanceOnQuestions.Any(x => x.UserId ==
cadetId && x.QuestionId == questionId))
                return NotFound();

            Performance.CorrectAnswer = correctAnswer;
            await _context.SaveChangesAsync();
            return Ok(Performance);
        }
    }
}
```

Листинг В.26 – Файл PerformanceOnTicketsController.cs

```
using AutoSchoolApp.API.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace AutoSchoolApp.API.Controllers
{
    public class PerformanceOnTicketsController : Controller
    {
        private readonly DrivingDbContext _context;

        public PerformanceOnTicketsController(DrivingDbContext context)
        {
            _context = context;
        }

        [HttpPost("SaveAnswer")]
        public async Task<ActionResult<PerformanceOnTicket>> Create(int
cadetId, bool correctAnswer, int ticketId)
        {
            PerformanceOnTicket performanceOnTicket = new()
            {
                UserId = cadetId,
                CorrectAnswer = correctAnswer,
                TicketId = ticketId
            };

            if (Helper.IsExists(ref performanceOnTicket,
_context.PerformanceOnTickets.ToList()))
            {
                await Update(cadetId, correctAnswer, ticketId);
                return Ok();
            }

            _context.PerformanceOnTickets.Add(performanceOnTicket);
            await _context.SaveChangesAsync();
```

```
            return Ok(performanceOnTicket);
        }

        [HttpPut("UpdateAnswer")]
        public async Task<ActionResult<PerformanceOnTicket>> Update(int
cadetId, bool correctAnswer, int ticketId)
        {
            var Performances = await
_context.PerformanceOnTickets.ToListAsync();
            var Performance = Performances.Find(x => x.UserId == cadetId
&& x.TicketId == ticketId);

            if (Performance == null)
                return BadRequest();

            if (!_context.PerformanceOnTickets.Any(x => x.UserId ==
cadetId && x.TicketId == ticketId))
                return NotFound();

            Performance.CorrectAnswer = correctAnswer;
            await _context.SaveChangesAsync();
            return Ok(Performance);
        }
    }
}
```

Приложение Г

(рекомендуемое)

Исходный код мобильного приложения

Листинг Г.1 – Файл MauiProgram.cs

```
using AutoSchoolApp.Pages;
using CommunityToolkit.Maui;
using Microsoft.Extensions.Logging;

namespace AutoSchoolApp
{
    public static class MauiProgram
    {
        public static MauiApp CreateMauiApp()
        {
            var builder = MauiApp.CreateBuilder();
            builder
                .UseMauiApp<App>()
                .ConfigureFonts(fonts =>
                {
                    fonts.AddFont("OpenSans-Regular.ttf",
"OpenSansRegular");
                    fonts.AddFont("OpenSans-Semibold.ttf",
"OpenSansSemibold");
                })
                .UseMauiCommunityToolkit();

#if DEBUG
            builder.Logging.AddDebug();
#endif
            builder.Services.AddTransient<DrivingSignPage>();
            builder.Services.AddTransient<ForgotPasswordPage>();
            builder.Services.AddTransient<HomePage>();
            builder.Services.AddTransient<OnboardingPage>();
            builder.Services.AddTransient<PrepQuestionsPage>();
            builder.Services.AddTransient<PrepTicketsPage>();
            builder.Services.AddTransient<SignUpPage>();
            builder.Services.AddTransient<SignInPage>();


            return builder.Build();
        }
    }
}
```

Листинг Г.2 – Файл App.xaml.cs

```
namespace AutoSchoolApp
{
    public partial class App : Application
    {
        public App()
        {
            InitializeComponent();

            MainPage = new AppShell();
        }
    }
}
```

Листинг Г.3 – Файл AppShell.xaml

```
<?xml version="1.0" encoding="UTF-8" ?>
<Shell
    x:Class="AutoSchoolApp.AppShell"
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:AutoSchoolApp"
    xmlns:pages="clr-namespace:AutoSchoolApp.Pages"
    Title="AutoSchoolApp"
    Shell.FlyoutBehavior="Disabled">

    <ShellContent  ContentTemplate="{DataTemplate
pages:OnboardingPage}"/>
    <ShellContent  Title="HomePage"
                   Route="HomePage"
                   ContentTemplate="{DataTemplate pages:HomePage}"/>

</Shell>
```

Листинг Г.4 – Файл AppShell.xaml.cs

```
namespace AutoSchoolApp
{
    public partial class AppShell : Shell
    {
        public AppShell()
        {
            InitializeComponent();
        }
    }
}
```

Листинг Г.5 – Файл ChooseQuestionPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AutoSchoolApp.Pages.ChooseQuestionPage"
             Title="ChooseQuestionPage">

    <ScrollView>
        <VerticalStackLayout Margin="15" Spacing="5">
            <Label Text="Подготовка по вопросам"
                   VerticalOptions="Center"
                   HorizontalOptions="Center"
                   FontAttributes="Bold"
                   FontSize="20"
                   Margin="0,20,0,60"/>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Общие положения" TextColor="#330044"
FontSize="15"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
```

```
                    <Label Text="25 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Дорожные знаки" TextColor="#330044"
FontSize="15"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="126 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_1"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Дорожная разметка" TextColor="#330044"
FontSize="15"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="40 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_2"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Сигналы светофора и регулировщика"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="39 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
```

```
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_3"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Начало движения, маневрирование"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="113 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_4"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Скорость движения" TextColor="#330044"
FontSize="15"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="19 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_5"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Обгон, опережение, встречный разъезд"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="36 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_6"/>
                </Frame.GestureRecognizers>
            </Frame>
```

```
            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Остановка и стоянка" TextColor="#330044"
FontSize="15"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="47 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_7"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Проезд перекрестков" TextColor="#330044"
FontSize="15"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="113 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_8"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Пользование внешними световыми приборами
и звуковыми сигналами" TextColor="#330044" FontSize="15"
FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="23 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_9"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Неисправности и условия допуска
транспортных средств к эксплуатации" TextColor="#330044" FontSize="15"
```

```
FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand"  WidthRequest="240"
Margin="90,0,0,0"/>
                    <Label Text="26 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_10"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Безопасность движения и техника
управления автомобилем" TextColor="#330044" FontSize="15"
FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="59 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_11"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Оказание доврачебной медицинской помощи"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="20 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_12"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Общие обязанности водителей"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
```

```
                    <Label Text="15 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_13"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Расположение транспортных средств на
проезжей части" TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="24 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_14"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Приоритет маршрутных транспортных
средств" TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="5 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_15"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Буксировка механических транспортных
средств" TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="8 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
```

```
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_16"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Применение специальных сигналов"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="9 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_17"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Движение по автомагистралям"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="13 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_18"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Учебная езда и дополнительные требования
к движению велосипедов" TextColor="#330044" FontSize="15"
FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="4 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_19"/>
                </Frame.GestureRecognizers>
```

```
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Движение в жилых зонах"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="7 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_20"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Движение через железнодорожные пути"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="11 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_21"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="5" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Пешеходные переходы и места остановок
маршрутноых транспортных средств" TextColor="#330044" FontSize="15"
FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="5 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_22"/>
                </Frame.GestureRecognizers>
            </Frame>
```

```
            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Перевозка людей и грузов"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="6 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_23"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Ответственность водителя"
TextColor="#330044" FontSize="15"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="15 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_24"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="8" HorizontalOptions="FillAndExpand" CornerRadius="30"
HeightRequest="110">
                <StackLayout Orientation="Horizontal">
                    <Label Text="Применение аварийной сигнализации и
знака аварийной остановки" TextColor="#330044" FontSize="15"
FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" WidthRequest="240"  Margin="90,0,0,0"/>
                    <Label Text="9 вопросов" TextColor="#330044"
FontSize="14"  FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="EndAndExpand" Margin="0,0,5,0"/>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_25"/>
                </Frame.GestureRecognizers>
            </Frame>
        </VerticalStackLayout>
    </ScrollView>
</ContentPage>
```

Листинг Г.6 – Файл ChooseQuestionPage.xaml.cs

```
namespace AutoSchoolApp.Pages;

public partial class ChooseQuestionPage : ContentPage
{
    public ChooseQuestionPage()
      {
            InitializeComponent();
      }

    private void TapGestureRecognizer_Tapped(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(1));
    }

    private void TapGestureRecognizer_Tapped_1(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(2));
    }

    private void TapGestureRecognizer_Tapped_2(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(3));
    }

    private void TapGestureRecognizer_Tapped_3(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(4));
    }

    private void TapGestureRecognizer_Tapped_4(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(5));
    }

    private void TapGestureRecognizer_Tapped_5(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(6));
    }

    private void TapGestureRecognizer_Tapped_6(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(7));
    }

    private void TapGestureRecognizer_Tapped_7(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(8));
    }
```

```
    private void TapGestureRecognizer_Tapped_8(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(9));
    }

    private void TapGestureRecognizer_Tapped_9(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(10));
    }

    private void TapGestureRecognizer_Tapped_10(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(11));
    }

    private void TapGestureRecognizer_Tapped_11(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(12));
    }

    private void TapGestureRecognizer_Tapped_12(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(13));
    }

    private void TapGestureRecognizer_Tapped_13(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(14));
    }

    private void TapGestureRecognizer_Tapped_14(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(15));
    }

    private void TapGestureRecognizer_Tapped_15(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(16));
    }

    private void TapGestureRecognizer_Tapped_16(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(17));
    }

    private void TapGestureRecognizer_Tapped_17(object sender,
TappedEventArgs e)
    {
```

```
        Navigation.PushAsync(new PrepQuestionsPage(18));
    }

    private void TapGestureRecognizer_Tapped_18(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(19));
    }

    private void TapGestureRecognizer_Tapped_19(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(20));
    }

    private void TapGestureRecognizer_Tapped_20(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(21));
    }

    private void TapGestureRecognizer_Tapped_21(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(22));
    }

    private void TapGestureRecognizer_Tapped_22(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(23));
    }

    private void TapGestureRecognizer_Tapped_23(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(24));
    }

    private void TapGestureRecognizer_Tapped_24(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(25));
    }

    private void TapGestureRecognizer_Tapped_25(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepQuestionsPage(26));
    }
}
```

Листинг Г.7 – Файл ChooseTicketPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AutoSchoolApp.Pages.ChooseTicketPage"
```

```
             Title="Подготовка по билетам">

    <ScrollView>
        <VerticalStackLayout Margin="20" Spacing="8 ">
            <Label Text="Выберите билет"
                   VerticalOptions="Center"
                   HorizontalOptions="Center"
                   FontAttributes="Bold"
                   FontSize="20"
                   Margin="0,20,0,60"/>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                        <Label Text="1" TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                        <Label Text="2" TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
```

```
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_1"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                        <Label Text="3" TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_2"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
```

```
                        <Label Text="4" TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_3"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                        <Label Text="5" TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_4"/>
```

```
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                    <Label Text="6" TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                </Frame>
                <VerticalStackLayout Padding="7">
                    <HorizontalStackLayout>
                        <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                    <HorizontalStackLayout>
                        <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                </VerticalStackLayout>
            </StackLayout>
            <Frame.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_5"/>
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                    <Label Text="7" TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                </Frame>
                <VerticalStackLayout Padding="7">
                    <HorizontalStackLayout>
                        <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                    <HorizontalStackLayout>
                        <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
```

```
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_6"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                        <Label Text="8" TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_7"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                        <Label Text="9" TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
```

```
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_8"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="10" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_9"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
```

```
                        <Label Text="11" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_10"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="12" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_11"/>
```

```
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                    <Label Text="13" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                </Frame>
                <VerticalStackLayout Padding="7">
                    <HorizontalStackLayout>
                        <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                    <HorizontalStackLayout>
                        <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                </VerticalStackLayout>
            </StackLayout>
            <Frame.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_12"/>
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                    <Label Text="14" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                </Frame>
                <VerticalStackLayout Padding="7">
                    <HorizontalStackLayout>
                        <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                    <HorizontalStackLayout>
                        <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
```

```
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_13"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="15" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_14"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="16" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
```

```
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_15"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="17" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_16"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
```

```
                        <Label Text="18" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_17"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="19" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_18"/>
```

```
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                    <Label Text="20" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                </Frame>
                <VerticalStackLayout Padding="7">
                    <HorizontalStackLayout>
                        <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                    <HorizontalStackLayout>
                        <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                </VerticalStackLayout>
            </StackLayout>
            <Frame.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_19"/>
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                    <Label Text="21" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                </Frame>
                <VerticalStackLayout Padding="7">
                    <HorizontalStackLayout>
                        <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                    <HorizontalStackLayout>
                        <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
```

```
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_20"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="22" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_21"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="23" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
```

```
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_22"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="24" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_23"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
```

```
                        <Label Text="25" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_24"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="26" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_25"/>
```

```
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                    <Label Text="27" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                </Frame>
                <VerticalStackLayout Padding="7">
                    <HorizontalStackLayout>
                        <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                    <HorizontalStackLayout>
                        <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                </VerticalStackLayout>
            </StackLayout>
            <Frame.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_26"/>
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                    <Label Text="28" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                </Frame>
                <VerticalStackLayout Padding="7">
                    <HorizontalStackLayout>
                        <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                    <HorizontalStackLayout>
                        <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
```

```
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_27"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="29" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_28"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="30" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
```

```
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_29"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="31" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_30"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
```

```
                        <Label Text="32" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_31"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="33" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_32"/>
```

```
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                    <Label Text="34" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                </Frame>
                <VerticalStackLayout Padding="7">
                    <HorizontalStackLayout>
                        <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                    <HorizontalStackLayout>
                        <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                </VerticalStackLayout>
            </StackLayout>
            <Frame.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_33"/>
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                    <Label Text="35" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                </Frame>
                <VerticalStackLayout Padding="7">
                    <HorizontalStackLayout>
                        <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                    </HorizontalStackLayout>
                    <HorizontalStackLayout>
                        <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
```

```
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_36"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="38" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_37"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
```

```
                        <Label Text="39" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_38"/>
                </Frame.GestureRecognizers>
            </Frame>

            <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
                <StackLayout Orientation="Horizontal">
                    <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="13,10,10,10" Margin="3">
                        <Label Text="40" TextColor="#330044"
FontSize="18" FontAttributes="Bold"/>
                    </Frame>
                    <VerticalStackLayout Padding="7">
                        <HorizontalStackLayout>
                            <Label Text="Решено дней назад: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="{Binding DaysAgo}"
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                        <HorizontalStackLayout>
                            <Label Text="Количество ошибок: "
TextColor="#330044" FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                            <Label Text="0" TextColor="#330044"
FontSize="14" IsVisible="false" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
                        </HorizontalStackLayout>
                    </VerticalStackLayout>
                </StackLayout>
                <Frame.GestureRecognizers>
                    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_39"/>
```

```
                </Frame.GestureRecognizers>
            </Frame>
        </VerticalStackLayout>
    </ScrollView>
</ContentPage>
```

Листинг Г.8 – Файл ChooseTicketPage.xaml.cs

```
namespace AutoSchoolApp.Pages;

public partial class ChooseTicketPage : ContentPage
{
     public ChooseTicketPage()
     {
          InitializeComponent();
     }

    private void TapGestureRecognizer_Tapped(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(1));
    }

    private void TapGestureRecognizer_Tapped_1(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(2));
    }

    private void TapGestureRecognizer_Tapped_2(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(3));
    }

    private void TapGestureRecognizer_Tapped_3(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(4));
    }

    private void TapGestureRecognizer_Tapped_4(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(5));
    }

    private void TapGestureRecognizer_Tapped_5(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(6));
    }

    private void TapGestureRecognizer_Tapped_6(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(7));
    }
```

```
    private void TapGestureRecognizer_Tapped_7(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(8));
    }

    private void TapGestureRecognizer_Tapped_8(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(9));
    }

    private void TapGestureRecognizer_Tapped_9(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(10));
    }

    private void TapGestureRecognizer_Tapped_10(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(11));
    }

    private void TapGestureRecognizer_Tapped_11(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(12));
    }

    private void TapGestureRecognizer_Tapped_12(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(13));
    }

    private void TapGestureRecognizer_Tapped_13(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(14));
    }

    private void TapGestureRecognizer_Tapped_14(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(15));
    }

    private void TapGestureRecognizer_Tapped_15(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(16));
    }

    private void TapGestureRecognizer_Tapped_16(object sender,
TappedEventArgs e)
    {
```

```
        Navigation.PushAsync(new PrepTicketsPage(17));
    }

    private void TapGestureRecognizer_Tapped_17(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(18));
    }

    private void TapGestureRecognizer_Tapped_18(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(19));
    }

    private void TapGestureRecognizer_Tapped_19(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(20));
    }

    private void TapGestureRecognizer_Tapped_20(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(21));
    }

    private void TapGestureRecognizer_Tapped_21(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(22));
    }

    private void TapGestureRecognizer_Tapped_22(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(23));
    }

    private void TapGestureRecognizer_Tapped_23(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(24));
    }

    private void TapGestureRecognizer_Tapped_24(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(25));
    }

    private void TapGestureRecognizer_Tapped_25(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(26));
    }
```

```
    private void TapGestureRecognizer_Tapped_26(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(27));
    }

    private void TapGestureRecognizer_Tapped_27(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(28));
    }

    private void TapGestureRecognizer_Tapped_28(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(29));
    }

    private void TapGestureRecognizer_Tapped_29(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(30));
    }

    private void TapGestureRecognizer_Tapped_30(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(31));
    }

    private void TapGestureRecognizer_Tapped_31(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(32));
    }

    private void TapGestureRecognizer_Tapped_32(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(33));
    }

    private void TapGestureRecognizer_Tapped_33(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(34));
    }

    private void TapGestureRecognizer_Tapped_34(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(35));
    }

    private void TapGestureRecognizer_Tapped_35(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(36));
```

```
    }

    private void TapGestureRecognizer_Tapped_36(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(37));
    }

    private void TapGestureRecognizer_Tapped_37(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(38));
    }

    private void TapGestureRecognizer_Tapped_38(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(39));
    }

    private void TapGestureRecognizer_Tapped_39(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new PrepTicketsPage(40));
    }
}
```

Листинг Г.9 – Файл DrivingSignPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:viewmodel="clr-namespace:AutoSchoolApp.ViewModels"
             x:DataType="viewmodel:DrivingSignViewModel"
             x:Class="AutoSchoolApp.Pages.DrivingSignPage"
             Title="DrivingSignPage">
    <VerticalStackLayout Margin="20" Spacing="0">
        <StackLayout>
            <Frame BackgroundColor="#aaffff" HeightRequest="160"
WidthRequest="160" CornerRadius="80" Padding="20" Margin="0,60,0,0"
HorizontalOptions="Center">
                <Image Source="event.png" Aspect="AspectFill"
Margin="0"/>
            </Frame>
        </StackLayout>

        <Label
            Text="Запись на занятие"
            VerticalOptions="Center"
            HorizontalOptions="Center"
            FontAttributes="Bold"
            FontSize="20"
            Margin="0,20,0,60"/>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30">
            <StackLayout Orientation="Horizontal" Padding="20,0,20,0">
                <Label x:Name="datelabel" Text="Выберите дату"
```

```
                    VerticalOptions="Center" HorizontalOptions="Center"
FontSize="18"/>
                <DatePicker  x:Name="datePicker"  Format="d"
DateSelected="DatePicker_DateSelected"  HorizontalOptions="EndAndExpand"
FontSize="18" />
            </StackLayout>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Margin="0,8,0,8"  Padding="0" HorizontalOptions="FillAndExpand"
CornerRadius="30" >
            <StackLayout Orientation="Horizontal" Padding="20,0,20,0">
                <Label x:Name="timelabel" Text="Выберите время"
                        VerticalOptions="Center"
HorizontalOptions="Center" FontSize="18"/>
                <TimePicker x:Name="timePicker" Time="17:00:00"
PropertyChanged="timePicker_PropertyChanged"
HorizontalOptions="EndAndExpand" FontSize="18" />
            </StackLayout>
        </Frame>

        <Button Text="Записаться" BackgroundColor="#5a19a0"
TextColor="#aaffff" FontAttributes="Bold" FontSize="Body"
                CornerRadius="30" WidthRequest="200" Margin="0,60,0,20"
x:Name="RegButton" Clicked="RegButton_Clicked"  />

        <Label
            Text="Назад"
            VerticalOptions="Center"
            HorizontalOptions="Center"
            FontSize="Subtitle">
            <Label.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped"/>
            </Label.GestureRecognizers>
        </Label>

    </VerticalStackLayout>
</ContentPage>
```

Листинг Г.10 – Файл DrivingSignPage.xaml.cs

```
//using Google.Android.Material.DatePicker;

namespace AutoSchoolApp.Pages;

public partial class DrivingSignPage : ContentPage
{
    DateTime selectedDate = new();

     public DrivingSignPage()
     {
        InitializeComponent();
        datePicker.MinimumDate = DateTime.Now;
        datePicker.MaximumDate = DateTime.Now.AddDays(7);
    }
```

```
    private async void TapGestureRecognizer_Tapped(object sender,
TappedEventArgs e)
    {
       await Navigation.PopToRootAsync();
    }

    private async void RegButton_Clicked(object sender, EventArgs e)
    {
        await Navigation.PopToRootAsync();
    }

    private void timePicker_PropertyChanged(object sender,
System.ComponentModel.PropertyChangedEventArgs e)
    {

    }

    private void DatePicker_DateSelected(object sender,
DateChangedEventArgs e)
    {
        selectedDate = e.NewDate.Date;
    }
}
```

Листинг Г.11 – Файл ForgotPasswordPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AutoSchoolApp.Pages.ForgotPasswordPage"
             Title="ForgotPasswordPage"
             Shell.NavBarIsVisible="False">

    <VerticalStackLayout Margin="20" Spacing="0">
        <StackLayout>
            <Frame BackgroundColor="#aaffff" HeightRequest="160"
WidthRequest="160" CornerRadius="80" Padding="20" Margin="0,60,0,0"
HorizontalOptions="Center">
                <Image Source="mail.png" Aspect="AspectFill" Margin="0"/>

            </Frame>
        </StackLayout>

        <Label
            Text="Вход без пароля"
            VerticalOptions="Center"
            HorizontalOptions="Center"
            FontAttributes="Bold"
            FontSize="20"
            Margin="0,20,0,60"
           />

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30">
            <StackLayout Orientation="Horizontal">
                <Frame BackgroundColor="MediumPurple" HeightRequest="50"
WidthRequest="50" CornerRadius="30" Padding="4" Margin="3">
```

```
                    <Image Source="maildog.png" Aspect="AspectFill"
Margin="0"/>
                </Frame>
                <Entry Text="{Binding UserName}" Placeholder="E-mail"
PlaceholderColor="#aa66ff" TextColor="#aaffff"
                       FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" Margin="0,0,20,0"/>
            </StackLayout>
        </Frame>

        <Button Text="Отправить код" BackgroundColor="#5a19a0"
TextColor="#aaffff" FontAttributes="Bold" FontSize="Body"
                CornerRadius="30" WidthRequest="200" Margin="0,60,0,20"
x:Name="ResetButton" Clicked="ResetButton_Clicked"  />

        <Label
            Text="Назад"
            VerticalOptions="Center"
            HorizontalOptions="Center"
            FontSize="Subtitle">
            <Label.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped"/>
            </Label.GestureRecognizers>
        </Label>

    </VerticalStackLayout>
</ContentPage>
```

Листинг Г.12 – Файл ForgotPasswordPage.xaml.cs

```
namespace AutoSchoolApp.Pages;

public partial class ForgotPasswordPage : ContentPage
{
    //public ForgotPasswordPage(ForgotPasswordModel vm)
    //{
    //    InitializeComponent();
    //    BindingContext = vm;
    //}

    public ForgotPasswordPage()
    {
        InitializeComponent();
    }

    private void TapGestureRecognizer_Tapped(object sender,
TappedEventArgs e)
    {
        Navigation.PopToRootAsync();
    }

    private void ResetButton_Clicked(object sender, EventArgs e)
    {
        Navigation.PushAsync(new ResetPage());
    }
}
```

Листинг Г.13 – Файл HomePage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AutoSchoolApp.Pages.HomePage"
             Title="HomePage">

    <Shell.TitleView>
        <StackLayout Orientation="Horizontal" HorizontalOptions="Center">
            <Label Text="автошкола АВ 2024" TextColor="#aaffff"
FontSize="Medium" Margin="10"/>
            <!--<Image Source="avatar.png" HeightRequest="40"
WidthRequest="40" Aspect="AspectFill" Margin="10">
                <Image.GestureRecognizers>
                    <TapGestureRecognizer Tapped="OnAvatarTapped"/>
                </Image.GestureRecognizers>
            </Image>-->
        </StackLayout>
    </Shell.TitleView>

    <VerticalStackLayout Margin="20" Spacing="0">
        <StackLayout>
            <Frame BackgroundColor="#aaffff" HeightRequest="160"
WidthRequest="160" CornerRadius="80" Padding="20" Margin="0,60,0,0"
HorizontalOptions="Center">
                <Image Source="quiz.png" Aspect="AspectFill" Margin="0"/>
            </Frame>
        </StackLayout>

        <Label
            Text="Учитесь вместе с нами"
            VerticalOptions="Center"
            HorizontalOptions="Center"
            FontAttributes="Bold"
            FontSize="20"
            Margin="0,20,0,60"/>

        <StackLayout Orientation="Vertical"
HorizontalOptions="CenterAndExpand">
            <Button Text="Записаться на практическое занятие"
BackgroundColor="#5a19a0" TextColor="#aaffff" FontAttributes="Bold"
CornerRadius="30" WidthRequest="350" Margin="0,25,0,0" FontSize="Body"
Clicked="OnPracticalTrainingClicked"/>
            <Button Text="Готовиться по вопросам"
BackgroundColor="#5a19a0" TextColor="#aaffff" FontAttributes="Bold"
CornerRadius="30" WidthRequest="350" Margin="0,25,0,0" FontSize="Body"
Clicked="OnAnswerQuestionsClicked"/>
            <Button Text="Готовиться по билетам"
BackgroundColor="#5a19a0" TextColor="#aaffff" FontAttributes="Bold"
CornerRadius="30" WidthRequest="350" Margin="0,25,0,0" FontSize="Body"
Clicked="OnSolveTicketsClicked"/>
        </StackLayout>
    </VerticalStackLayout>
</ContentPage>
```

Листинг Г.14 – Файл HomePage.xaml.cs

```
namespace AutoSchoolApp.Pages;

public partial class HomePage : ContentPage
{
    public HomePage()
    {
        InitializeComponent();
    }

    //private void OnAvatarTapped(object sender, EventArgs e)
    //{
    //    // Handle avatar tapped event, navigate to personal account
page
    //}

    private void OnPracticalTrainingClicked(object sender, EventArgs e)
    {
        Navigation.PushAsync(new DrivingSignPage());
    }

    private void OnAnswerQuestionsClicked(object sender, EventArgs e)
    {
        Navigation.PushAsync(new ChooseQuestionPage());
    }

    private void OnSolveTicketsClicked(object sender, EventArgs e)
    {
        Navigation.PushAsync(new ChooseTicketPage());
    }
}
```

Листинг Г.15 – Файл OnboardingPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AutoSchoolApp.Pages.OnboardingPage"
             Title="OnboardingPage">
    <VerticalStackLayout Spacing="25">
        <Label
            Text="Welcome to .NET MAUI!"
            VerticalOptions="Center"
            HorizontalOptions="Center" />
        <Button Text="Войти"
                Clicked="Button_Clicked"
                HorizontalOptions="Center"
                VerticalOptions="Center">
        </Button>
        <Button Text="Зарегистрироваться"
                Clicked="Button_Clicked_1"
                HorizontalOptions="Center"
                VerticalOptions="Center">
        </Button>
    </VerticalStackLayout>
</ContentPage>
```

Листинг Г.16 – Файл OnboardingPage.xaml.cs

```
namespace AutoSchoolApp.Pages;

public partial class OnboardingPage : ContentPage
{
     public OnboardingPage()
     {
          InitializeComponent();
     }

    private async void Button_Clicked(object sender, EventArgs e)
    {
        await Navigation.PushAsync(new SignInPage());
        //await Shell.Current.GoToAsync($"//{nameof(HomePage)}");
    }

    private async void Button_Clicked_1(object sender, EventArgs e)
    {
        await Navigation.PushAsync(new SignUpPage());
    }
}
```

Листинг Г.17 – Файл PrepQuestionsPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AutoSchoolApp.Pages.PrepQuestionsPage"
             Title="PrepQuestionsPage">
    <VerticalStackLayout Margin="20" Spacing="8 ">

        <StackLayout IsVisible="False">
            <Frame HeightRequest="500"  Padding="0" Margin="0,60,0,0"
HorizontalOptions="Center">
                <Image Source="{Binding QuestionPicture}"
Aspect="AspectFill" Margin="0"/>
            </Frame>
        </StackLayout>

        <Label Text="{Binding QuestionFormulation}"
                   VerticalOptions="Center"
                   HorizontalOptions="Center"
                   FontAttributes="Bold"
                   FontSize="18"
                   Margin="0,20,0,60"/>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                  <Label Text="1." TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                </Frame>
                <Label Text="{Binding CorrectAnswer}" TextColor="#aaffff"
FontSize="14" FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand"/>
            </StackLayout>
            <Frame.GestureRecognizers>
```

```
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped"/>
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30">
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                    <Label Text="2." TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                </Frame>
                <Label Text="{Binding FirstIncorrectAnswer}"
TextColor="#aaffff" FontSize="14"  FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
            </StackLayout>
            <Frame.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_1"/>
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30">
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                    <Label Text="3." TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                </Frame>
                <Label Text="{Binding SecondIncorrectAnswer}"
TextColor="#aaffff"  FontSize="14" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
            </StackLayout>
            <Frame.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_2"/>
            </Frame.GestureRecognizers>
        </Frame>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30"
IsVisible="False">
            <StackLayout Orientation="Horizontal">
                <Frame HeightRequest="50" WidthRequest="50"
CornerRadius="30" Padding="18,12,12,15" Margin="3">
                    <Label Text="4." TextColor="#330044" FontSize="18"
FontAttributes="Bold"/>
                </Frame>
                <Label Text="{Binding ThirdIncorrectAnswer}"
TextColor="#aaffff" FontSize="14" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"/>
            </StackLayout>
            <Frame.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_3"/>
            </Frame.GestureRecognizers>
```

```
        </Frame>
    </VerticalStackLayout>
</ContentPage>
```

Листинг Г.18 – Файл PrepQuestionsPage.xaml.cs

```
namespace AutoSchoolApp.Pages;

public partial class PrepQuestionsPage : ContentPage
{
    private int questionID;

    public PrepQuestionsPage()
      {
            InitializeComponent();
      }
    public PrepQuestionsPage(int qID)
    {
        questionID = qID;
        InitializeComponent();
    }

    private void TapGestureRecognizer_Tapped(object sender,
TappedEventArgs e)
    {
        SendCorrectAnswer();
    }

    private void TapGestureRecognizer_Tapped_1(object sender,
TappedEventArgs e)
    {
        SendIncorrectAnswer(1);
    }

    private void TapGestureRecognizer_Tapped_2(object sender,
TappedEventArgs e)
    {
        SendIncorrectAnswer(2);
    }

    private void TapGestureRecognizer_Tapped_3(object sender,
TappedEventArgs e)
    {
        SendIncorrectAnswer(3);
    }
}
```

Листинг Г.19 – Файл PrepTicketsPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AutoSchoolApp.Pages.PrepTicketsPage"
             Title="PrepTicketsPage">
    <VerticalStackLayout>
        <Label
            Text="Welcome to .NET MAUI!"
            VerticalOptions="Center"
```

```
            HorizontalOptions="Center" />
    </VerticalStackLayout>
</ContentPage>
```

Листинг Г.20 – Файл PrepTicketsPage.xaml.cs

```
namespace AutoSchoolApp.Pages;

public partial class PrepTicketsPage : ContentPage
{
     public PrepTicketsPage()
     {
          InitializeComponent();
     }
}
```

Листинг Г.21 – Файл SignInPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AutoSchoolApp.Pages.SignInPage"
             Title="SignInPage"
              Shell.NavBarIsVisible="False">

    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition  Height="220"/>
            <RowDefinition  Height="*"/>
            <RowDefinition  Height="60"/>
        </Grid.RowDefinitions>

        <Path Data="m0,-1l800.58444,-1l8.41558,166.84079c-
583.96105,127.44372 -392.37664,-73.11174 -810.00000,26.7937410,-
193.63453z" Fill ="#5a19a0"/>
        <Path Data="m299.5,248" Opacity="NaN" Stroke="#000"
Fill="#332D40"/>

        <StackLayout>
            <Frame BackgroundColor="#aaffff" HeightRequest="160"
WidthRequest="160" CornerRadius="80" Padding="0" Margin="0,60,0,0"
HorizontalOptions="Center">
                <Image Source="avatar.png" Aspect="AspectFill"
Margin="0"/>

            </Frame>
        </StackLayout>

        <Grid RowSpacing="5" Grid.Row="1" Margin="20,20,20,0">
            <Grid.RowDefinitions>
                <RowDefinition Height="*"/>
                <RowDefinition Height="50"/>
                <RowDefinition Height="50"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="40"/>
                <RowDefinition Height="40"/>
                <RowDefinition Height="*"/>
            </Grid.RowDefinitions>
```

```
            <Label Grid.Row="1" Text="Добро пожаловать!"
HorizontalOptions="Center" FontSize="Title" FontAttributes="Bold"
Padding="0" Margin="0"/>
            <Label Grid.Row="2" Text="Войдите чтобы продолжить"
HorizontalOptions="Center" FontSize="Subtitle"/>

            <StackLayout Grid.Row="3" Orientation="Vertical"
HorizontalOptions="CenterAndExpand">
                <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30">
                    <StackLayout Orientation="Horizontal">
                        <Frame BackgroundColor="MediumPurple"
HeightRequest="50" WidthRequest="50" CornerRadius="30" Padding="4"
Margin="3">
                            <Image Source="iconuser.png"
Aspect="AspectFill" Margin="0"/>
                        </Frame>
                        <Entry Text="{Binding UserName}"
Placeholder="Username" PlaceholderColor="#aa66ff" TextColor="#aaffff"
FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" Margin="0,0,20,0"/>
                    </StackLayout>
                </Frame>
                <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Margin="0,8,0,0"  Padding="0" HorizontalOptions="FillAndExpand"
CornerRadius="30">
                    <StackLayout Orientation="Horizontal">
                        <Frame BackgroundColor="MediumPurple"
HeightRequest="50" WidthRequest="50" CornerRadius="30" Padding="6,2,6,8"
Margin="3">
                            <Image Source="lock.png" Aspect="AspectFill"
Margin="0"/>
                        </Frame>
                        <Entry Text="{Binding Password}"
Placeholder="Password" PlaceholderColor="#aa66ff" TextColor="#aaffff"
FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" Margin="0,0,20,0"/>
                    </StackLayout>
                </Frame>

                <StackLayout Orientation="Horizontal" Margin="0,8,0,0"
Padding="0">
                    <CheckBox IsChecked="False"/>
                    <Label Text="Запомнить меня" Padding="0,0,20,0"
FontSize="Small" VerticalTextAlignment="Center"
HorizontalTextAlignment="Center"/>

                    <Label Text="Забыли пароль?" TextColor="#aa66ff"
FontAttributes="Bold" VerticalTextAlignment="Center"
HorizontalTextAlignment="Center" HorizontalOptions="EndAndExpand">
                        <Label.GestureRecognizers>
                            <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped"  />
                        </Label.GestureRecognizers>
                    </Label>
                </StackLayout>
```

```
                <Button Text="Войти" BackgroundColor="#5a19a0"
TextColor="#aaffff" FontAttributes="Bold" CornerRadius="30"
WidthRequest="200" Margin="0,15,0,0"
                        Clicked="Button_Clicked"  FontSize="Body"/>

                <StackLayout Orientation="Horizontal"
HorizontalOptions="CenterAndExpand" Margin="0,120,0,0" Padding="0">
                    <Label Text="Нет аккаунта?" Padding="0,0,20,0"
FontSize="Small" VerticalTextAlignment="Center"
HorizontalTextAlignment="Center"/>
                    <Label Text="Регистрация" TextColor="#aa66ff"
FontSize="Small" FontAttributes="Bold" VerticalTextAlignment="Center"
HorizontalTextAlignment="Center" HorizontalOptions="EndAndExpand">
                        <Label.GestureRecognizers>
                            <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped_1"  />
                        </Label.GestureRecognizers>
                    </Label>
                </StackLayout>
            </StackLayout>
        </Grid>
    </Grid>
</ContentPage>
```

Листинг Г.22 – Файл SignInPage.xaml.cs

```
namespace AutoSchoolApp.Pages;

public partial class SignInPage : ContentPage
{
     public SignInPage()
     {
          InitializeComponent();
     }

    private void TapGestureRecognizer_Tapped(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new ForgotPasswordPage());
    }

    private void TapGestureRecognizer_Tapped_1(object sender,
TappedEventArgs e)
    {
        Navigation.PushAsync(new SignUpPage());
    }

    private void Button_Clicked(object sender, EventArgs e)
    {
        Navigation.PushAsync(new HomePage());
    }
}
```

Листинг Г.23 – Файл SignUpPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
```

```
             x:Class="AutoSchoolApp.Pages.SignUpPage"
             Title="SignUpPage"
             Shell.NavBarIsVisible="False">

    <VerticalStackLayout Margin="20" Spacing="0">
        <StackLayout>
            <Frame BackgroundColor="#aaffff" HeightRequest="160"
WidthRequest="160" CornerRadius="80" Padding="20" Margin="0,60,0,0"
HorizontalOptions="Center">
                <Image Source="add.png" Aspect="AspectFill" Margin="0"/>
            </Frame>
        </StackLayout>

        <Label
            Text="Создание аккаунта"
            VerticalOptions="Center"
            HorizontalOptions="Center"
            FontAttributes="Bold"
            FontSize="20"
            Margin="0,20,0,60"/>

        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame BackgroundColor="MediumPurple" HeightRequest="50"
WidthRequest="50" CornerRadius="30" Padding="4" Margin="3">
                    <Image Source="iconuser.png" Aspect="AspectFill"
Margin="0"/>
                </Frame>
                <Entry Text="{Binding UserName}" Placeholder="Username"
PlaceholderColor="#aa66ff" TextColor="#aaffff" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
Margin="0,0,20,0"/>
            </StackLayout>
        </Frame>
        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Margin="0,8,0,8"  Padding="0" HorizontalOptions="FillAndExpand"
CornerRadius="30" >
            <StackLayout Orientation="Horizontal">
                <Frame BackgroundColor="MediumPurple" HeightRequest="50"
WidthRequest="50" CornerRadius="30" Padding="6,2,6,8" Margin="3">
                    <Image Source="lock.png" Aspect="AspectFill"
Margin="0"/>
                </Frame>
                <Entry Text="{Binding Password}" Placeholder="Password"
PlaceholderColor="#aa66ff" TextColor="#aaffff" FontAttributes="Bold"
VerticalOptions="Center" HorizontalOptions="FillAndExpand"
Margin="0,0,20,0"/>
            </StackLayout>
        </Frame>
        <Frame BackgroundColor="#5a19a0" BorderColor="#330044"
Padding="0" HorizontalOptions="FillAndExpand" CornerRadius="30">
            <StackLayout Orientation="Horizontal">
                <Frame BackgroundColor="MediumPurple" HeightRequest="50"
WidthRequest="50" CornerRadius="30" Padding="4" Margin="3">
                    <Image Source="maildog.png" Aspect="AspectFill"
Margin="0"/>
                </Frame>
```

```
                <Entry Text="{Binding Email}" Placeholder="E-mail"
PlaceholderColor="#aa66ff" TextColor="#aaffff"
                       FontAttributes="Bold" VerticalOptions="Center"
HorizontalOptions="FillAndExpand" Margin="0,0,20,0"/>
            </StackLayout>
        </Frame>

        <Button Text="Регистрация" BackgroundColor="#5a19a0"
TextColor="#aaffff" FontAttributes="Bold" FontSize="Body"
                CornerRadius="30" WidthRequest="200" Margin="0,60,0,20"
x:Name="RegButton" Clicked="RegButton_Clicked"  />

        <Label
            Text="Назад"
            VerticalOptions="Center"
            HorizontalOptions="Center"
            FontSize="Subtitle">
            <Label.GestureRecognizers>
                <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped"/>
            </Label.GestureRecognizers>
        </Label>

    </VerticalStackLayout>
</ContentPage>
```

Листинг Г.24 – Файл SignUpPage.xaml.cs

```
namespace AutoSchoolApp.Pages;

public partial class SignUpPage : ContentPage
{
     public SignUpPage()
     {
          InitializeComponent();
     }

    private void RegButton_Clicked(object sender, EventArgs e)
    {
      AddUser();
    }

    private void TapGestureRecognizer_Tapped(object sender,
TappedEventArgs e)
    {
        Navigation.PopToRootAsync();
    }
}
```

Приложение Д

(рекомендуемое)

Результаты тестирования серверной части

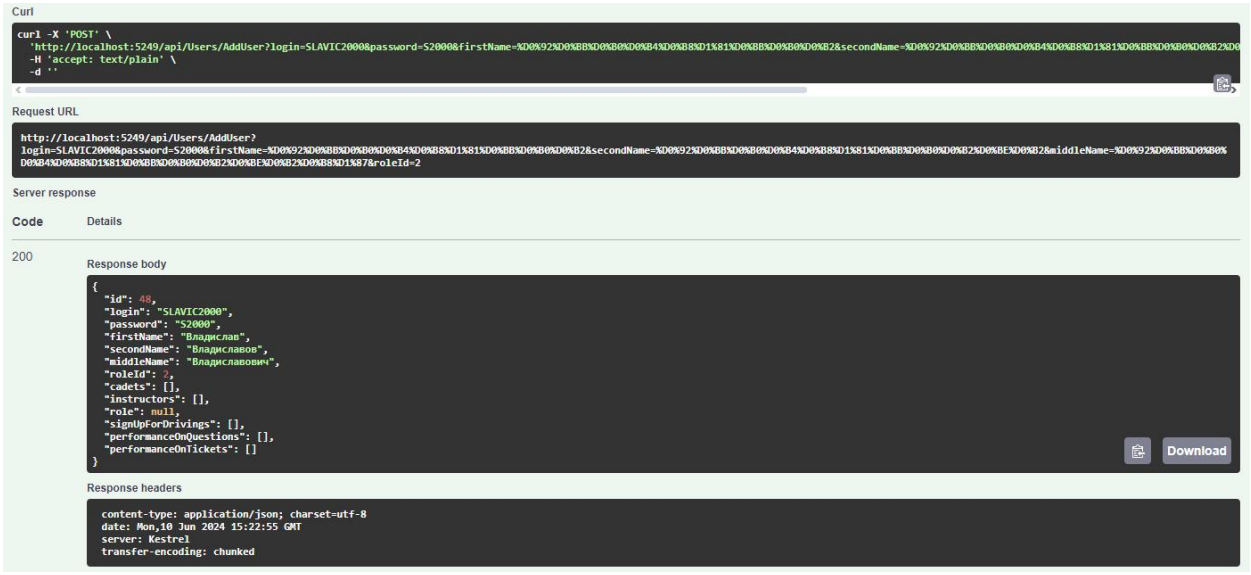


Рисунок Д.1 – Добавление нового пользователя

Рисунок Д.2 – Вход существующего пользователя в систему

Рисунок Д.3 – Получение информации о пользователе

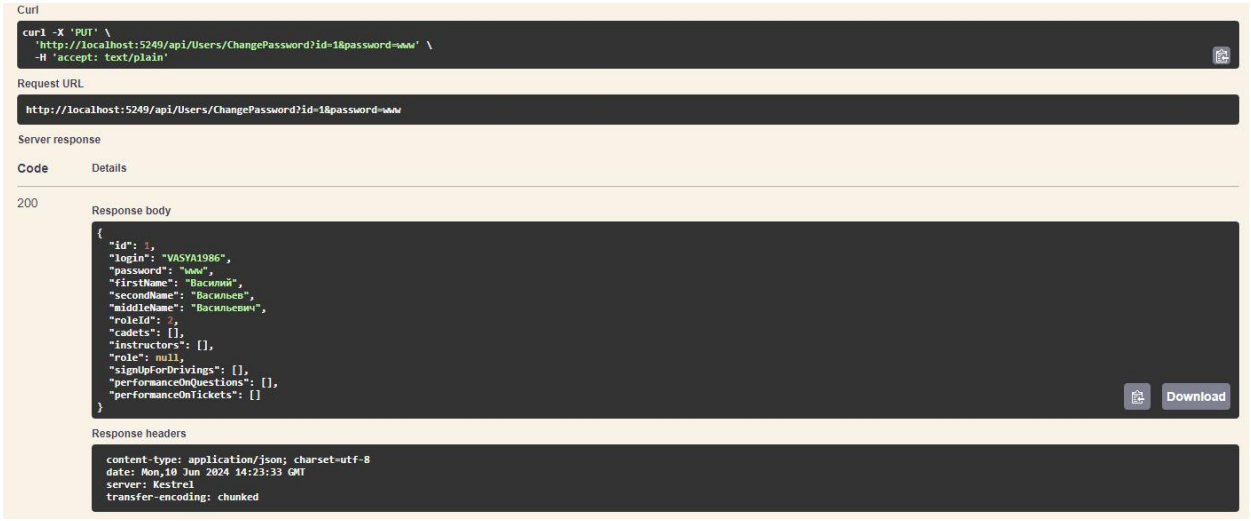


Рисунок Д.4 – Изменение пароля пользователя

Рисунок Д.5 – Запись курсанта на занятие

Рисунок Д.6 – Отмена записи на занятие

Рисунок Д.7 – Сохранение ответа на отдельный теоретический вопрос

Рисунок Д.8 – Сохранение ответов на вопросы в билетег

Приложение Е

(рекомендуемое)

Результаты тестирования клиентской части

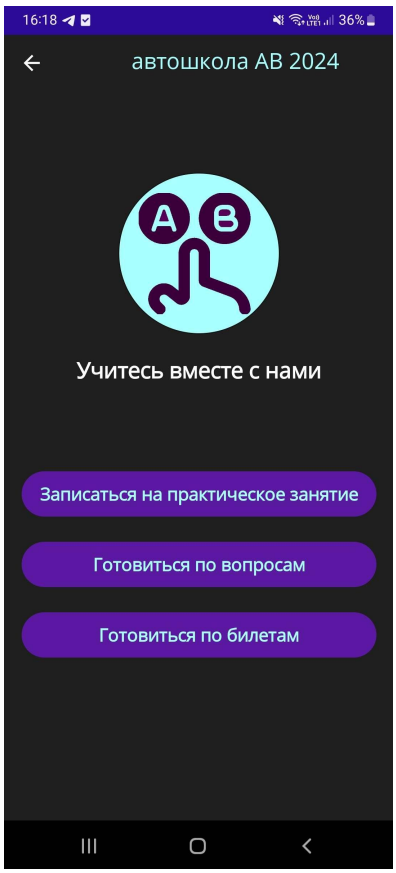


Рисунок Е.1 – Успешная регистрация в системе

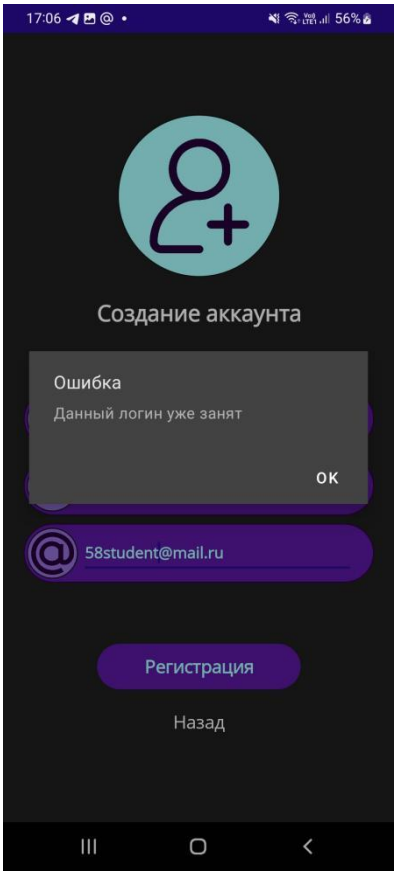


Рисунок Е.2 – Логин занят

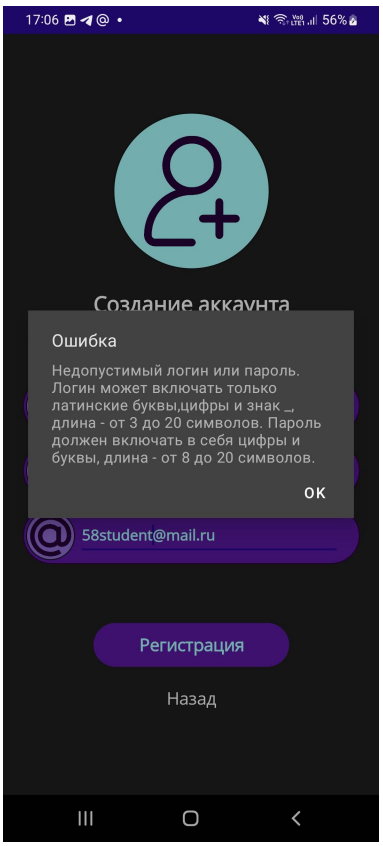


Рисунок Е.3 – Недопустимые символы в логине и пароле

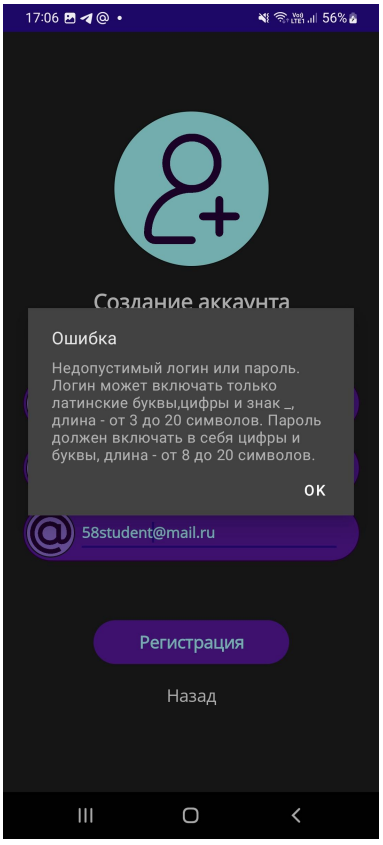


Рисунок Е.4 – Логин и пароль недопустимой длины

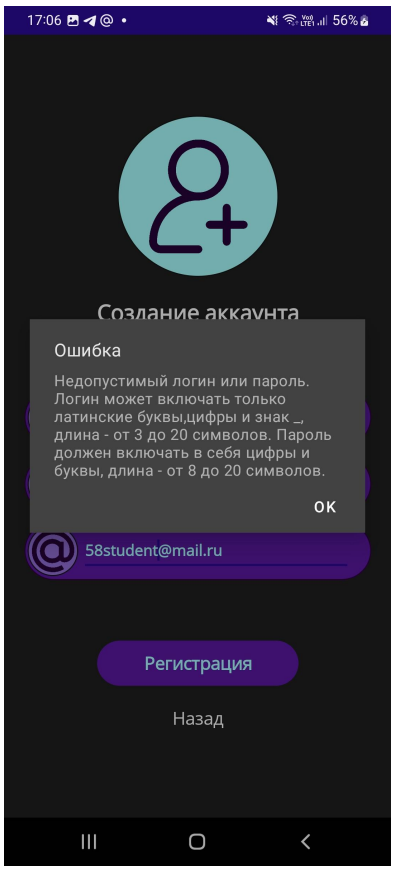


Рисунок Е.5 – Пароль без букв

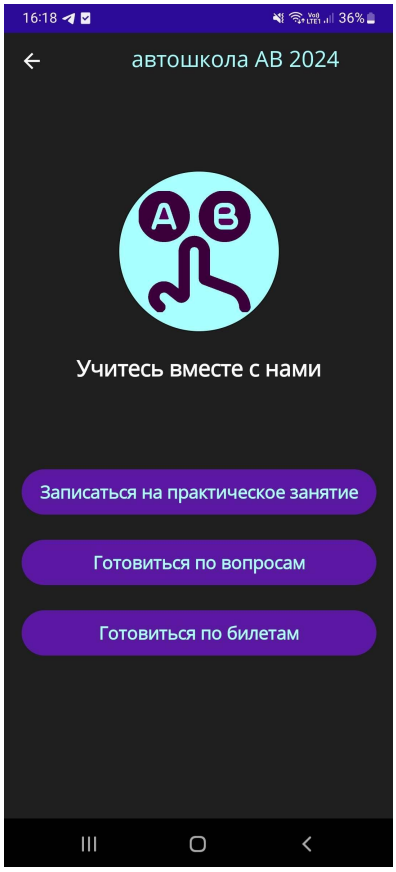


Рисунок Е.6 – Успешная авторизация в системе

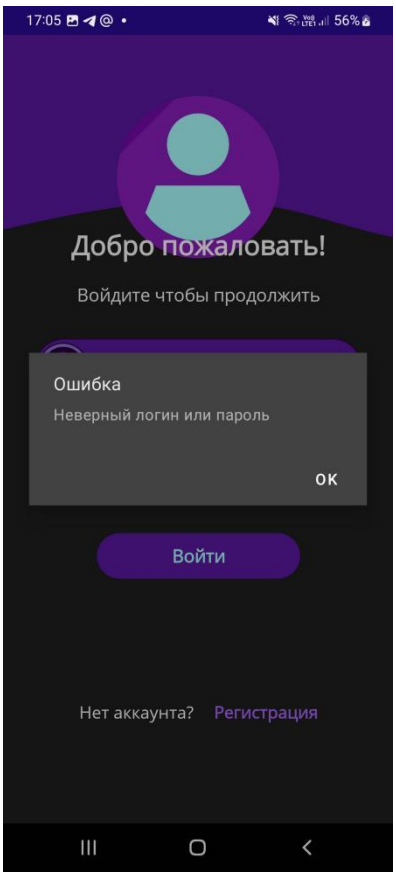


Рисунок Е.7 – Неверный логин или пароль

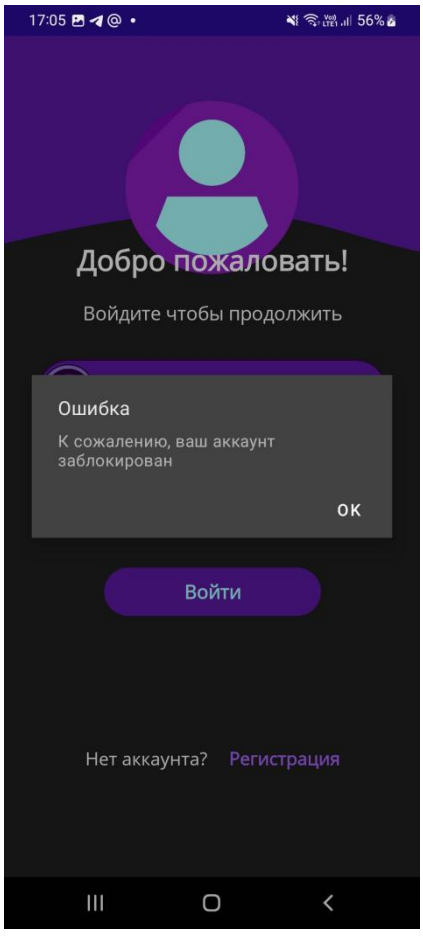


Рисунок Е.8 – Аккаунт заблокирован

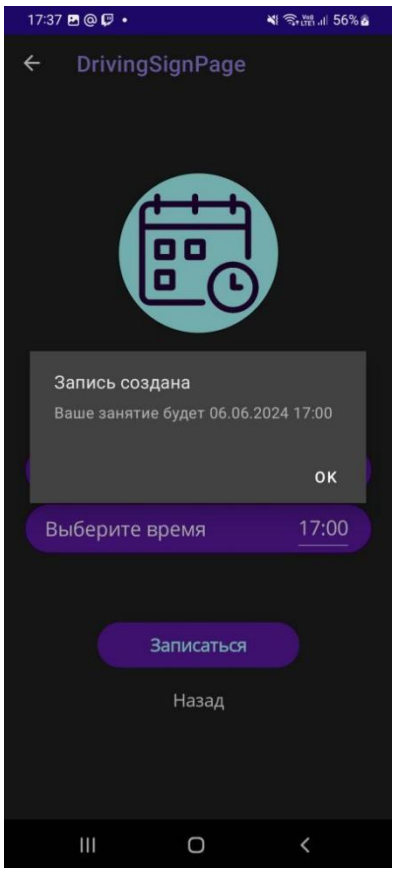


Рисунок Е.9 – Запись на свободное время

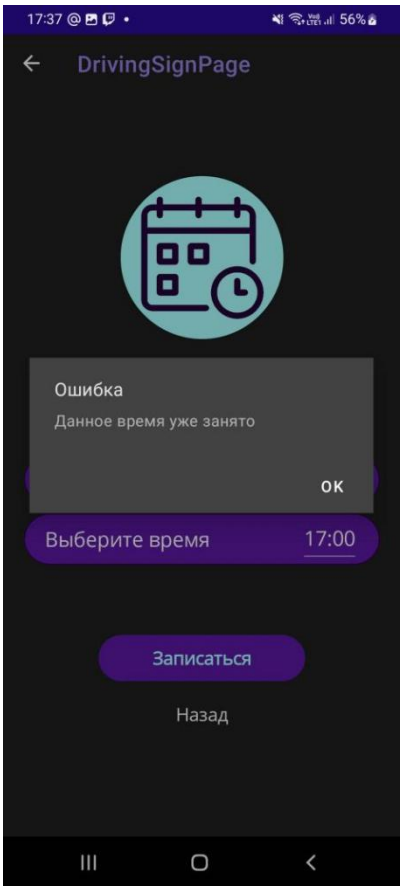


Рисунок Е.10 – Запись на занятое время

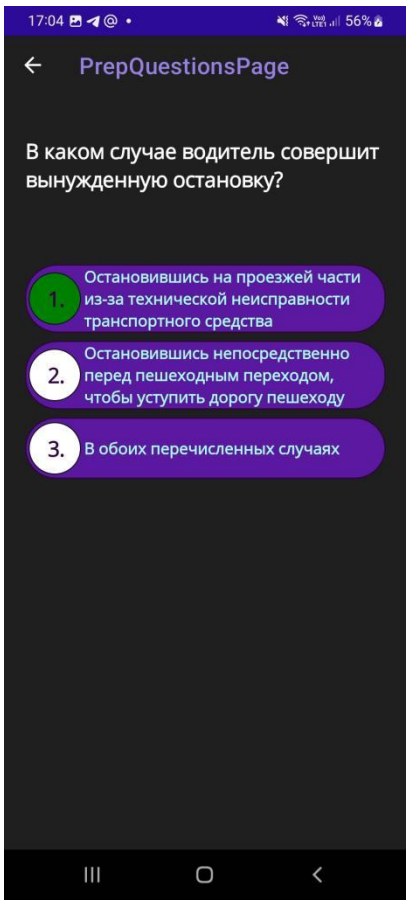


Рисунок Е.11 – Верный ответ на вопрос

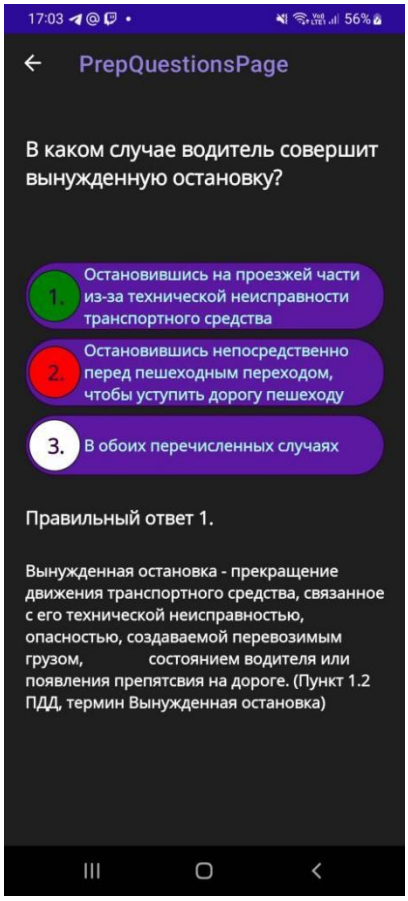


Рисунок Е.12– Неверный ответ на вопрос