

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
МОиПЭВМ

Направление подготовки 09.03.04 «Программная инженерия»

БАКАЛАВРСКАЯ РАБОТА

на тему

**Мобильное приложение для изучения
иностранного языка**

Студент _____ Угроватов Дмитрий Владимирович _____

(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

_____ Макарычев П.П. _____

(подпись)

(фамилия, инициалы)

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

Секретарь ГЭК

_____ Попова Н.А. _____

(подпись)

(фамилия, инициалы)

Пенза, 2020

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

Факультет
Вычислительной техники

Кафедра
МО и ПЭВМ

«Утверждаю»
Заведующий кафедрой
П.П. Макарычев
« 30 » декабря 2019 г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

1. Студент Угроватов Дмитрий Владимирович гр. 16ВП1 факультета ВТ

<Фамилия, имя, отчество полностью>

2. Тема работы Мобильное приложение для изучения иностранного языка

Тема утверждена приказом ПГУ № 1488/о от « 30 » декабря 2019

3. Руководитель работы к.т.н., доцент кафедры «МОиПЭВМ» Князев В.Н.

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

1) *Разработать приложение для мобильных устройств под управлением системы Android, обучающее пользователей иностранному языку.*

2) *Среда выполнения приложения: операционная система Android 9 и выше*

Среда выполнения сервера: операционная система Windows 10

Языковые средства разработки: Dart, UML PHP, SQL,

Инструментальные средства разработки: Flutter,

Android Software Development Kit (SDK), jdk1.8.0_201k1.8.0_201, смартфон Samsung Galaxy S8, MS Visio, ArgoUML, PhpStorm, MySQL

Технологии разработки: RUP, ООП

3) *Основные функции мобильного приложения:*

а) обеспечить пользователям среду для комфортного изучения материала;

б) предоставлять различные задания для обучения;

в) фиксировать результаты выполнения заданий.

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	<i>Анализ предметной области постановка задачи</i>	15%	13.04.2020– 21.04.2020	
2	<i>Проектирование</i>	25%	22.04.2020– 05.05.2020	
3	<i>Реализация</i>	40%	06.05.2020– 30.05.2020	
4	<i>Контроль качества программного обеспечения</i>	5%	31.05.2020– 02.06.2020	
5	<i>Оформление пояснительной записки</i>	15%	03.06.2020– 13.06.2020	

Дата выдачи задания "30" декабря 2019

Руководитель бакалаврской работы _____

(подпись, дата)

Задание к исполнению принял студент _____

(подпись, дата)

Работу к защите допустить

Декан факультета _____

(подпись, дата)

Реферат

Пояснительная записка содержит 165 листов, 55 рисунков, 12 таблиц, 25 использованных источников, 8 приложений.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, АНГЛИЙСКИЙ ЯЗЫК, ANDROID, MYSQL, FLUTTER, DART, RUP, UML, IDEF1, MYSQL, PHP

Объектом разработки является мобильное приложение для изучения иностранного языка.

Цель работы – разработка мобильного приложения для изучения иностранного языка, которое позволяет изучать иностранный язык с помощью текстовых заданий, картинок и заданий с медиа, а также предоставляемой теории.

Разработка серверной части происходила на языке программирования Php. В качестве среды разработки использовалась PhpStrom. В качестве СУБД была использована СУБД MySQL.

Разработка приложения производилась на языке программирования dart с использованием sdk Flutter. В качестве среды разработки использовалась среда программирования Android studio, операционная система Windows 10. Среда исполнения – операционная система Android 9.0.

					ПГУ 09.03.04 – 08BP161.25 ПЗ			
Изм.	Лист	№ докум.	Подп.	Дата	«Мобильное приложение для изучения иностранного языка» Пояснительная записка	Лит.	Лист	Листов
Разраб.		Угроватов Д.В.					4	165
Пров.		Князев В.Н						
Реценз.								
Н. контр.		Такташкин Д.В						
Утв.		Макарычев П.П				гр. 16ВП1		

Содержание

Введение.....	6
1 Анализ предметной области и требований.....	8
1.1 Анализ предметной области и постановка задач.....	8
1.2 Сравнительный анализ программных продуктов-аналогов	8
1.2.1 Анализ онлайн сервиса по изучению языков Duolingo.....	9
1.2.2 Анализ онлайн сервиса по изучению английского языка Lingualeo	10
1.2.3 Анализ онлайн сервиса по изучению языка Puzzle English.....	11
1.2.4 Анализ онлайн сервиса по изучению языков Bussu	12
1.3 Анализ целевой аудитории.....	13
1.4 Анализ требований.....	15
1.4.1 Функциональные требования.....	15
1.4.2 Нефункциональные требования	26
2 Проектирование мобильного приложения	27
2.1 Выбор технологий и инструментальных средств разработки	27
2.2 Проектирование архитектуры программного обеспечения.....	28
2.3 Планирование разработки и оценка бюджета	28
2.4 Проектирование интерфейса мобильного приложения	42
2.5 Проектирование серверной части приложения	49
2.5.1 Логическое проектирование базы данных.....	50
3 Реализация приложения	54
3.1 Реализация серверной части приложения	54
3.1.1 Реализация базы данных.....	54
3.1.2 Разработка серверной части приложения	55
3.2 Реализация мобильного приложения	55
4 Контроль качества программного обеспечения.....	58
4.1 Тестирование серверной части	58
4.2 Тестирование клиентской части мобильного приложения.....	60
4.2.1 Функциональное тестирование.....	60
4.3 Метрики кода.....	61

4.3.1 Метрика Холстеда.....	62
4.3.2 Цикломатическая метрика Мак-Кейба	65
4.3.3 Метрика Чепина	66
4.4 Автоматизированное тестирование.....	66
4.4.1 Unit-тесты.....	67
Заключение	68
Список использованных источников	69
Приложение А. Глоссарий	72
Приложение Б. Сетевой график и критический путь	74
Приложение В. Исходный код серверной части.....	77
Приложение Г. Исходный код приложения	88
Приложение Д. Диаграмма классов приложения	151
Приложение Е. Результаты тестирования серверной части	153
Приложение Ж. Результаты тестирования мобильного приложения.....	160
Приложение З. Граф функции	163

Введение

На данный момент информационные технологии входят в число наиболее быстро развивающихся отраслей. Им уделяется большое внимание не только в России, но и во всем мире. Об этом свидетельствует программа «Цифровая экономика Российской Федерации», основной задачей которой является «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», утвержденная в 2017 году [1].

В связи с увеличением количества владельцев смартфонов, простотой использования интернета и постоянным усовершенствованием технологий, на данный момент возникло огромное обилие интернет сервисов и мобильных приложений. «Стратегия развития информационного общества в Российской Федерации на 2017 – 2030 годы» предусматривает широкомасштабное применение в информационном обществе web-сервисов сети «Интернет», а также различных мобильных устройств и, соответственно, мобильных приложений: «Электронные средства массовой информации, информационные системы, социальные сети, доступ к которым осуществляется с использованием сети «Интернет», стали частью повседневной жизни россиян. ... В России информационное общество характеризуется широким распространением и доступностью мобильных устройств...» [2].

В сфере образования также активно внедряются информационные технологии. Одним из подтверждений этому является Федеральный закон «Об образовании в Российской Федерации», в котором сказано: «При реализации образовательных программ используются различные образовательные технологии, в том числе дистанционные образовательные технологии, электронное обучение» [3].

В настоящее время иностранные языки составляют значимую часть жизни людей. Например, при общении в сети интернет или при чтении литературы на оригинальном языке. Кроме этого, в школах изучению иностранных языков уделяется значительная часть учебного процесса. Однако, существует не так много сервисов, способных на достойном уровне вовлекаться в учебный процесс по изучению языков.

Исходя из перечисленной информации актуальной целью данной квалификационной работы является разработка мобильного приложения для изучения иностранного языка.

1 Анализ предметной области и требований

1.1 Анализ предметной области и постановка задач

Для многих людей, и, в частности, школьников, их мобильные устройства стали частью их повседневного образа жизни. Анализируемое в рамках выпускной квалификационной работы приложение должно позволить упростить изучение иностранных языков как школьникам, так и всем людям в целом. Вышеперечисленные факты, а также то, что в настоящее время информационные технологии внедряются во все сферы жизни, особенно подчеркивают актуальность тематики.

Разрабатываемое мобильное приложение предназначено для того, чтобы пользователи могли выполнять различные задания для улучшения знания языка и изучать теорию по нему. Пользователь может выбрать задания на аудирование, по составлению предложений или идентификацию изображения.

Таким образом, мобильное приложение для изучения иностранного языка должно предоставлять следующие основные функции:

- просмотр теории;
- навигация по меню;
- выполнение текстовых заданий;
- выполнение заданий на идентификацию изображений;
- выполнение заданий с аудио/видео;
- изменение уровня сложности заданий;
- просмотр статистики;
- регистрация новых пользователей;
- авторизация пользователей.

1.2 Сравнительный анализ программных продуктов-аналогов

В соответствии с темой работы, было принято решение более подробно проанализировать средства для изучения иностранного языка и провести сравнительный анализ наиболее популярных представителей этих средств.

1.2.1 Анализ онлайн сервиса по изучению языков Duolingo

Duolingo – это онлайн-сервис для изучения массы иностранных языков как через веб-приложение, так и с помощью мобильных клиентов. Данный сервис был запущен в конце 2011 года и имеет широкую базу пользователей [4, 5]. На рисунке 1 представлен экран выбора заданий в приложении Duolingo.

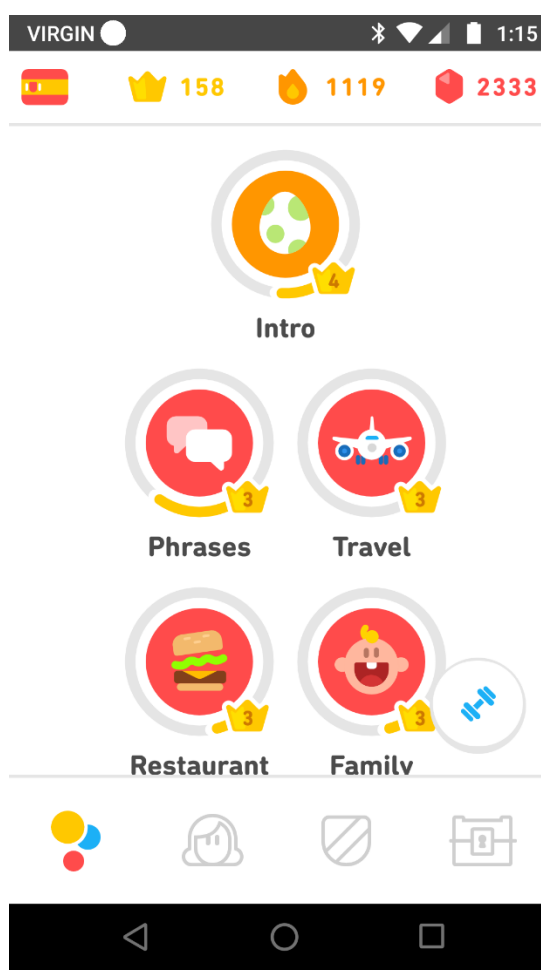


Рисунок 1 – Экран выбора заданий в приложении Duolingo

Он является одним из самых популярных сервисов для изучения иностранного языка. Все задания предоставлены в игровой форме.

Перечислим основные преимущества онлайн сервиса Duolingo:

- широкая база языков;
- имеется как мобильная версия, так и сайт;
- позволяет изучать языки «с нуля».

Из недостатков данного сервиса можно выделить следующие:

- отсутствие раздела теории;
- сервис частично бесплатен, из-за чего встречается много рекламы;
- в бесплатной версии имеется ограничение по количеству доступных для выполнения заданий за один день.

1.2.2 Анализ онлайн сервиса по изучению английского языка **Lingualeo**

Онлайн сервис для изучения английского языка «Lingualeo» был запущен в 2010 году группой российских разработчиков. Он имеет большую базу различного вида заданий, в отличие от первого аналога. Данный сервис также имеет свой мобильный клиент [6]. На рисунке 2 представлен экран с заданием в приложении Lingualeo.



Рисунок 2 – Экран с заданием в приложении Lingualeo

Далее приведены основные преимущества онлайн сервиса Lingualeo:

- достаточно широкий выбор текстов, музыки и видео на английском языке с возможностью получения мгновенного перевода и добавления незнакомых слов на изучение;
- наличие версии для мобильных устройств.

К недостаткам данного сервиса можно отнести:

- раздел теории предоставляет информацию в неудобной форме из-за чего он трудно воспринимается;
- ограничения для пользователей, которые используют бесплатную версию;
- платная версия доступна только по подписке.

1.2.3 Анализ онлайн сервиса по изучению языка Puzzle English

Онлайн сервис Puzzle English предоставляет комплексное изучение английского языка. Сервис предоставляет пять различных типов заданий, направленных для развития определенных навыков: видеопазлы, аудиопазлы, тренажер, направленный на освоение английской грамматики, транслейты. Сервис предоставляет выбор текстов с разным уровнем сложности [7]. На рисунке 3 представлено меню выбора заданий.

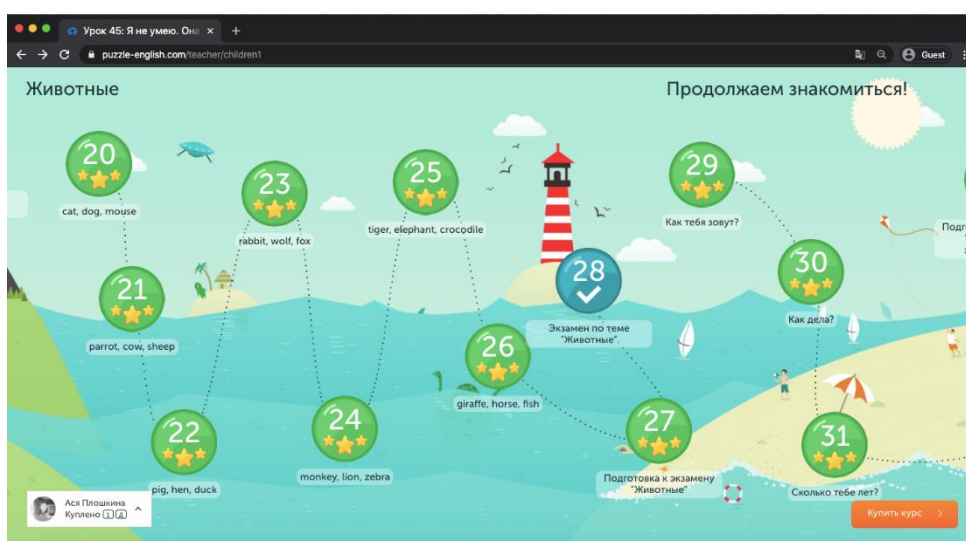


Рисунок 3 – Меню выбора заданий в Puzzle English

Из преимуществ онлайн сервиса Puzzle English выделяют:

- большая база теоретического материала и заданий;
- различные виды курсов.

Среди недостатков онлайн сервиса Puzzle English можно выделить:

- существенное ограничение бесплатной версии;

- высокая стоимость курсов;
- отделение Puzzle Movies в отдельный сервис;
- отсутствие мобильной версии, из-за чего возникают проблемы для доступа к нему с мобильного устройства.

1.2.4 Анализ онлайн сервиса по изучению языков Bussu

Bussu – это виртуальное сообщество для изучения 12 иностранных языков. Здесь собран полный языковой курс, включающий небольшие интерактивные уроки, подходящие для ежедневного и регулярного изучения языка [8]. На рисунке 4 представлен экран выбора заданий.

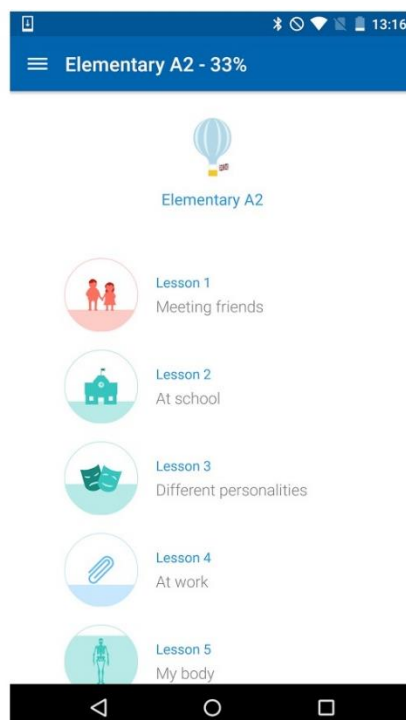


Рисунок 4 – Экран выбора заданий в приложении Bussu

Далее приведем преимущества онлайн сервиса Bussu:

- большой выбор языков для изучения;
- прямое общение с носителями языка.

Из недостатков данного сервиса можно выделить:

- небольшой бесплатный функционал, например, нет доступа к грамматике;

– высокая стоимость уроков.

Таким образом, в результате анализа аналогов было принято решение о создании собственного мобильного приложения для изучения иностранного языка, не имеющего выявленных недостатков.

1.3 Анализ целевой аудитории

Целевой аудиторией для разрабатываемого приложения являются люди школьного возраста и старше.

По данным ВЦИОМ от 30 сентября 2019 года, 63% опрошенных Россиян считают, что изучение иностранного языка необходимо в современных условиях. При интенсивности глобализационных процессов, затрагивающих различные сферы общественной жизни, такой тезис вполне оправдан. Чаще всего о необходимости изучать иностранный язык говорят жители Москвы и Санкт-Петербурга (75%), представители женской аудитории (69%), респонденты с высшим или неполным высшим образованием (66%).

На рисунке 5 представлен результат данного опроса [9].



Рисунок 5 – Результат опроса ВЦИОМ о важности изучении иностранных языков

На рисунке 6 представлены результаты опроса о преимуществах иностранного языка [9]. Из результатов данного опроса видно, что перечень

ключевых преимуществ в знании иностранного языка за последние пять лет изменился несущественно. Однако в целом россияне стали в большей мере эти преимущества осознавать. Так, если в 2014 г. общение с иностранцами как главный «плюс» знания иностранного языка называли 28% россиян, то сегодня таких уже 36%. Также существенно увеличилась доля считающих, что знание иностранного языка увеличивает возможности поездок за границу (16% в 2014 г. и 34% в 2019 г.). На 11% стало больше тех, кто видит ключевое преимущество такого знания для своей работы (22% в 2014 г. и 33% в 2019 г.).

По Вашему мнению, какие преимущества дает знание иностранного языка? *

(закрытый вопрос, до пяти ответов, %, представлены ответы, названные не менее чем 2% респондентов)

**

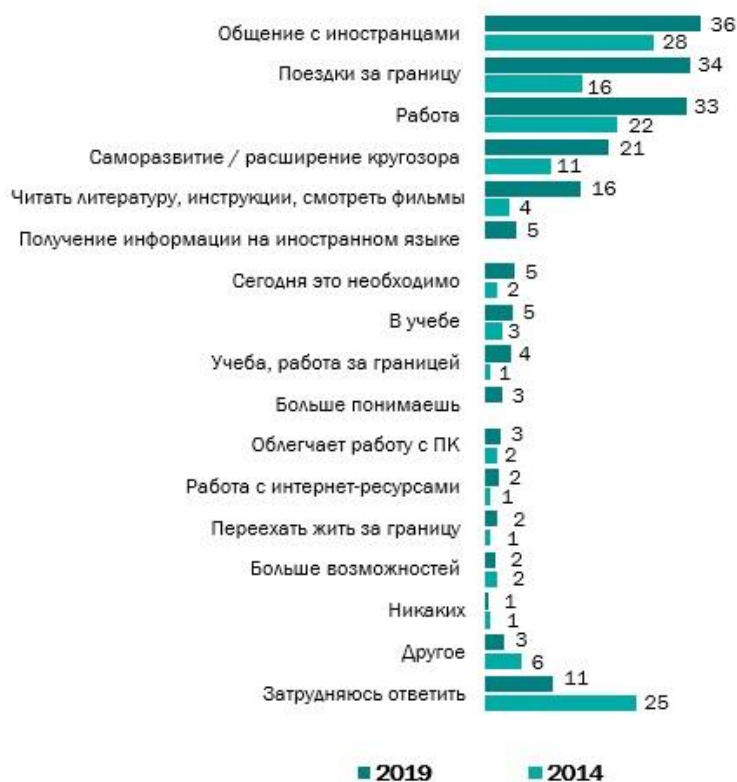


Рисунок 6 – Результат опроса ВЦИОМ о преимуществах знания иностранного языка

На конференции разработчиков Google I/O 2019 компания объявила, что во всем мире более 2,5 миллиардов активных устройств работает на операционной системе Android [10].

Активными называют те устройства, которые ежемесячно отмечаются в магазине приложений Google Play Store. К ним не относятся устройства от неизвестных китайских производителей, от Amazon с Fire OS, и, в целом, все смартфоны и планшеты со сторонними прошивками.

В том же году во время конференц-звонка, посвященного финансовым результатам Apple, старший вице-президент и финансовый директор компании Лука Маэстри (Luca Maestri) рассказал о том, что во всем мире используется более 900 миллионов активных iPhone [11].

Из всего этого можно сделать вывод, что мобильные устройства на базе операционной системы Android являются самыми популярными. Поэтому мобильное приложение будет разрабатываться для людей школьного возраста и старше, владеющих мобильными устройствами с данной ОС.

1.4 Анализ требований

Анализ требований – один из важнейших этапов разработки программного обеспечения, включающий в себя сбор и систематизация требований к будущему приложению [12].

Можно выделить функциональные и нефункциональные требования. Функциональные требования описывают функции, которые должно выполнять приложение, поведение приложения при взаимодействии с ним пользователя. Нефункциональные требования описывают свойства и ограничения приложения, которые не относятся к его поведению [13].

1.4.1 Функциональные требования

В результате анализа предметной области и продуктов-аналогов были выделены следующие основные функциональные требования к приложению:

- просмотр теории;
- навигация по меню;
- выполнение текстовых заданий;
- выполнение заданий на идентификацию изображений;

- выполнение заданий с аудио/видео;
- изменение уровня сложности заданий;
- просмотр статистики;
- регистрация пользователей;

Диаграмма вариантов использования мобильного приложения представлена на рисунке 7.



Рисунок 7 – Диаграмма вариантов использования мобильного приложения

Диаграмма вариантов использования отображает основные функциональные требования к приложению. На ней четко видно возможности пользователя. Спецификации прецедентов позволяют лучше понять последовательность действий, выполняемых актерами при использовании конкретного функционала приложения. Далее представлены спецификации прецедентов. На рисунке 8 представлена спецификация прецедента «Выполнить задание с картинкой».

Прецедент: Выполнить задание с картинкой
ID: 1
Краткое описание: Пользователь выполняет задание на идентификацию изображения
Главные актеры: Пользователь
Второстепенные актеры: Сервер
Предусловия: Нет
Основной поток: <ol style="list-style-type: none"> 1. Пользователь переходит на вкладку «Задания» 2. Пользователь нажимает кнопку заданий с картинками 3. Система отправляет запрос на получение данных о заданиях серверу 4. Сервер возвращает список актуальных заданий 5. Пользователь выбирает задание 6. Система выводит задание пользователю 7. Пользователь выполняет задание
Постусловия: <ol style="list-style-type: none"> 1. Система выводит на экран результат выполнения задания 2. Система отправляет результат выполнения задания на сервер 3. Сервер заносит полученные данные в базу данных
Альтернативные потоки: Нет

Рисунок 8 – Спецификация прецедента «Выполнить задание с картинкой»

На рисунках 9, 10 представлены спецификации прецедентов «Выполнить задание с текстом» и «Выполнить задание с аудио/видео» соответственно.

Прецедент: Выполнить задание с текстом
ID: 2
Краткое описание: Пользователь выполняет задание на составление предложений
Главные актеры: Пользователь
Второстепенные актеры: Сервер
Предусловия: Нет
Основной поток: <ol style="list-style-type: none"> 1. Пользователь переходит на вкладку «Задания» 2. Пользователь нажимает кнопку заданий с текстом 3. Система отправляет запрос на получение данных о заданиях серверу 4. Сервер возвращает список актуальных заданий 5. Пользователь выбирает задание 6. Система выводит задание пользователю 7. Пользователь выполняет задание
Постусловия: <ol style="list-style-type: none"> 1. Система выводит на экран результат выполнения задания 2. Система отправляет результат выполнения задания на сервер 3. Сервер заносит полученные данные в базу данных
Альтернативные потоки: Нет

Рисунок 9 – Спецификация прецедента «Выполнить задание с текстом»

Прецедент: Выполнить задание с аудио/видео
ID: 3
Краткое описание: Пользователь выполняет задание на аудирование
Главные актеры: Пользователь
Второстепенные актеры: Сервер
Предусловия: Нет
Основной поток: <ol style="list-style-type: none"> 1. Пользователь переходит на вкладку «Задания» 2. Пользователь нажимает кнопку заданий с медиа 3. Система отправляет запрос на получение данных о заданиях серверу 4. Сервер возвращает список актуальных заданий 5. Пользователь выбирает задание 6. Система выводит задание пользователю 7. Пользователь выполняет задание
Постусловия: <ol style="list-style-type: none"> 1. Система выводит на экран результат выполнения задания 2. Система отправляет результат выполнения задания на сервер 3. Сервер заносит полученные данные в базу данных
Альтернативные потоки: Нет

Рисунок 10 – Спецификация прецедента «Выполнить задание с аудио/видео»

Спецификации прецедентов, представленные на рисунках 8 – 10, описывают то, какие действия должен совершить пользователь, чтобы выполнить задание. Алгоритм действий зависит от типа задания, которое выбирает пользователь. Система всегда ведет себя одинаково. В запрос на

получение данных о заданиях входят информация об уровне сложности и информация о типе задания.

На рисунке 11 представлена спецификация прецедента «Просмотреть статистику».

Прецедент: Просмотреть статистику
ID: 4
Краткое описание: Пользователь просматривает результаты по выполнению заданий
Главные актеры: Пользователь, сервер
Второстепенные актеры: Нет
Предусловия: Нет
Основной поток: <ol style="list-style-type: none"> 1. Пользователь выбирает тип заданий 2. Система отправляет запрос на получение данных о результатах уже выполненных заданий серверу 3. Сервер проверяет в базе данных наличие информации о выполненных заданиях соответствующего уровня сложности и типа пользователем 4. Сервер возвращает полученные данные 5. Система отображает список заданий и результаты их выполнения пользователем
Постусловия: <ol style="list-style-type: none"> 1. Пользователь просматривает статистику по выполненным заданиям
Альтернативные потоки: Нет

Рисунок 11 – Спецификация прецедента «Просмотреть статистику»

Спецификация прецедента, изображенная на рисунке 12, описывает алгоритм действий при просмотре статистики. Система выводит статистику

как по уже выполненным заданиям, так и по невыполненным. Таким образом можно отследить, какие задания еще не были выполнены.

Прецедент: Выбрать уровень сложности
ID: 5
Краткое описание: Пользователь выбирает уровень сложности заданий
Главные актеры: Пользователь
Второстепенные актеры: Нет
Предусловия: Нет
Основной поток: <ol style="list-style-type: none"> 1. Пользователь переходит на вкладку «Настройки» 2. Пользователь нажимает кнопку выбора уровня сложности 3. Система выводит на экран существующие уровни сложности 4. Пользователь выбирает уровень сложности
Постусловия: Нет
Альтернативные потоки: Нет

Рисунок 12 – Спецификация прецедента «Выбрать уровень сложности»

На рисунке 13 представлена спецификация прецедента «Просмотреть теорию». Пользователь просматривает теорию по одному из представленных разделов, которая приходит ответом от сервера.

Прецедент: Просмотреть теорию
ID: 6
Краткое описание: Пользователь просматривает теорию о языке
Главные актеры: Пользователь
Второстепенные актеры: Нет
Предусловия: Нет
Основной поток: <ol style="list-style-type: none"> 1. Пользователь переходит на вкладку «Теория» 2. Пользователь нажимает кнопку выбора раздела теории 3. Сервер ищет актуальную информацию по соответствующему разделу теории 4. Сервер возвращает найденную информацию системе 5. Система выводит на экран актуальную информацию по выбранному разделу
Постусловия: 1. Пользователь просматривает теорию
Альтернативные потоки: Нет

Рисунок 13 – Спецификация прецедента «Просмотреть теорию»

На рисунке 14 представлена спецификация прецедента «Зарегистрировать пользователя».

Прецедент: Зарегистрировать пользователя
ID: 7
Краткое описание: Происходит регистрация пользователя
Главные актеры: Пользователь
Второстепенные актеры: Сервер
Предусловия: Пользователь впервые запускает приложение
Основной поток: <ol style="list-style-type: none"> 1. Пользователь нажимает кнопку входа 2. Система запрашивает данные у пользователя 3. Пользователь дает разрешение на использование своих данных 4. Система отправляет серверу запрос на регистрацию пользователя 5. Сервер вносит данные о пользователе в базу данных
Постусловия: <ol style="list-style-type: none"> 1. Система выводит на экран главное меню приложения
Альтернативные потоки: Нет

Рисунок 14 – Спецификация прецедента «Зарегистрировать пользователя»

Представленное описание прецедентов позволяет продемонстрировать работу системы не только при правильных действиях пользователя, но и описывает работу системы при возникновении ошибок.

Для прецедента «Выполнить задание с картинкой» были составлены диаграммы деятельности и диаграммы последовательности. Диаграмма деятельности представлена на рисунке 15.

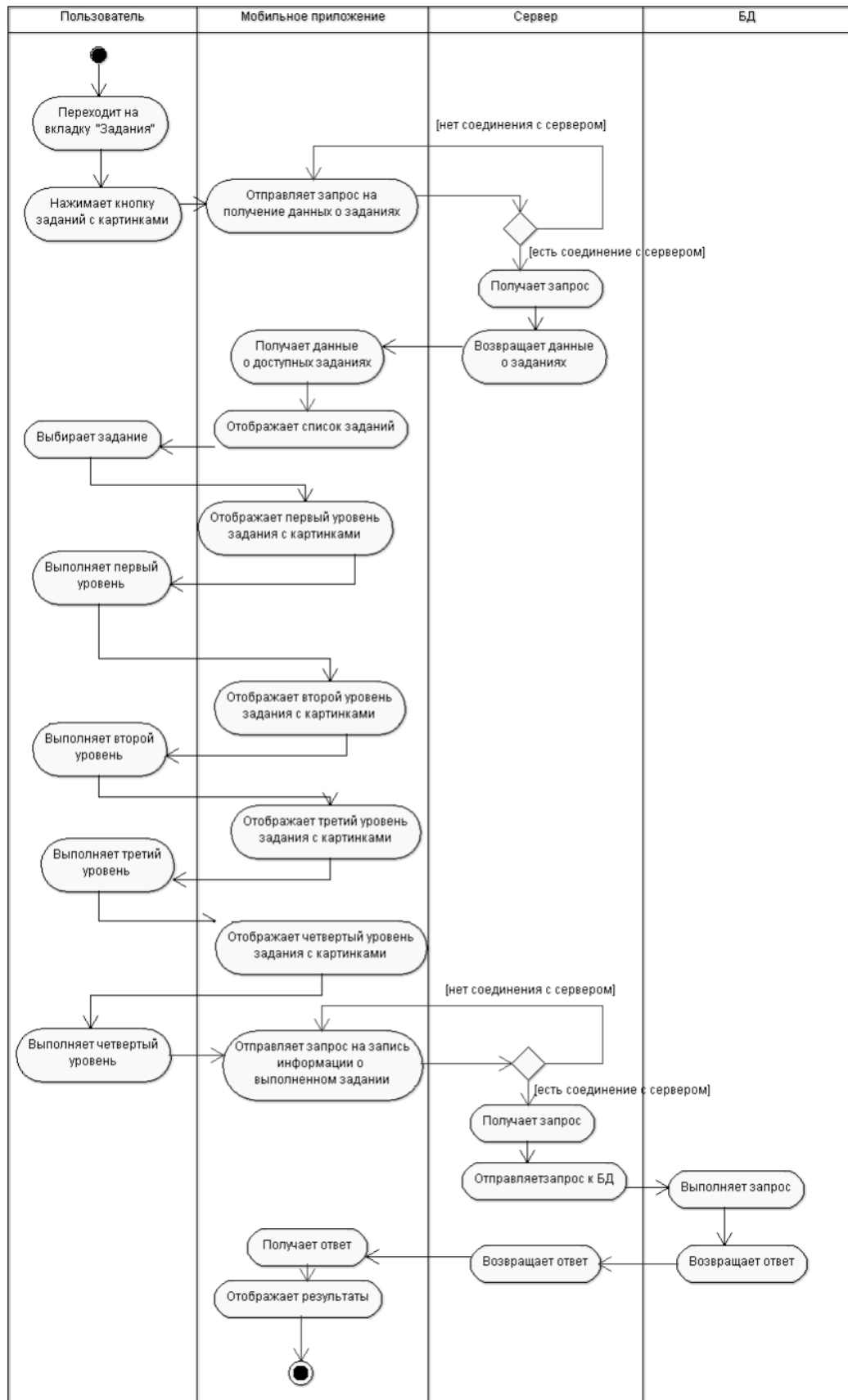


Рисунок 15 – Диаграмма деятельности для прецедента «Выполнить задание с картинкой»

Диаграмма последовательности для прецедента «Выполнить задание с картинкой» представлена на рисунке 16.

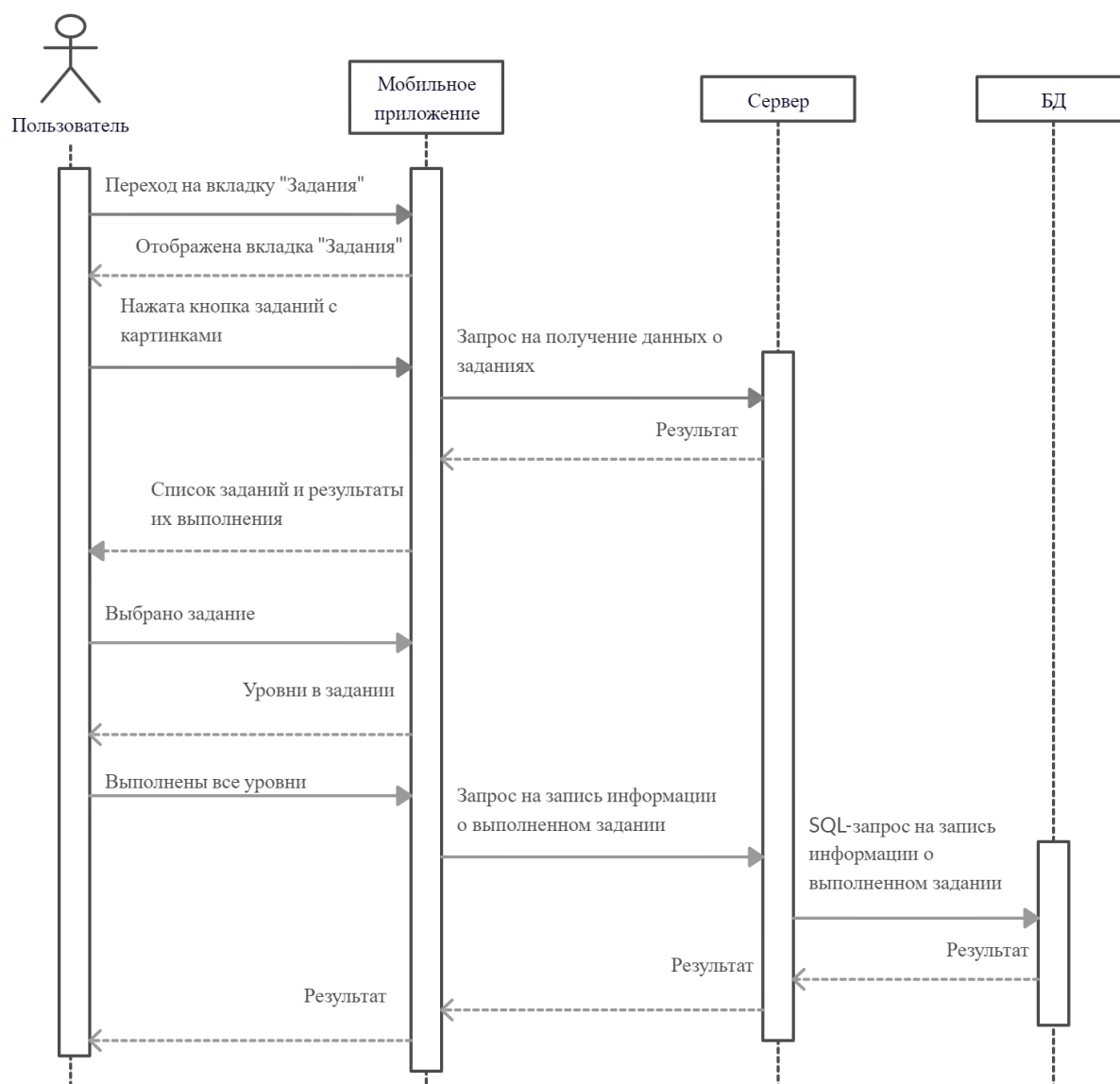


Рисунок 16 – Диаграмма последовательности для прецедента «Выполнить задание с картинкой»

В результате анализа функциональных требований были определены и уточнены основные функции разрабатываемого мобильного приложения, были составлены следующие диаграммы: диаграмма вариантов использования, диаграмма деятельности и последовательности и спецификации прецедентов.

1.4.2 Нефункциональные требования

Разработка будет состоять из следующих этапов: разработка клиентской части, разработка серверной части, проектирование базы данных. Соответственно, составим требования для каждой из них отдельно.

Приведем требования к клиентской части:

- приложение должно быть удобно для использования, следовательно, должно обладать понятным интерфейсом;
- в приложении не должно возникать исключительных ситуаций, приводящих к аварийному завершению работы;
- разработка приложения должна осуществляться в среде Android Studio при помощи набора средств разработки Flutter на языке Dart;
- приложение должно функционировать на мобильных устройствах с операционной системой Android 9 и выше.

Из требований к серверной части выделим следующее:

- обмен должен осуществляться по протоколу TCP.
- разработку серверной части вести в среде PhpStorm на языке Php;
- развертывание сервера должно происходить при помощи Apache.

Выделим требования к базе данных:

- в качестве целевой СУБД использовать MySQL;
- запросы к базе данных должны осуществляться на языке SQL.

Для написания серверной части будет использоваться среда разработки PhpStorm. Эта среда поддерживает PHP последней версии, SQL, а также обладает возможностью подсветки синтаксиса и автоматического рефакторинга. Также PhpStorm поддерживает систему плагинов для расширения своих возможностей [14].

Разработка будет проводится в операционной системе Windows, а тестирование приложения будет происходить на телефоне Samsung Galaxy S8. В результате анализа были составлены функциональные и нефункциональные требования к клиентской части, серверной части и базе данных.

2 Проектирование мобильного приложения

2.1 Выбор технологий и инструментальных средств разработки

Для реализации приложения была выбрана методология Rational Unified Process (RUP). Эта методология является одной из самых популярных на данный момент. Для разработки приложения была выбрана именно эта методология, потому что в ней используется итеративная модель разработки, которая позволяет быстро обнаруживать и устранять риски на ранних стадиях разработки [15, 16]. Каждая итерация составляет от 2 до 6 недель, по окончании которых получается промежуточная, но функциональная версия конечного продукта [15].

На этапе планирования разработки приложения и оценки бюджета было использовано программное средство Microsoft Project, позволяющее эффективно управлять проектом на различных этапах его реализации. Оно дает возможность выполнить структуризацию проекта путем разделения его на этапы, задачи и подзадачи, выявить критические задачи (задачи, длительность которых существенно влияет на длительность реализации всего проекта), получить сетевой график и календарный план проекта, осуществить назначение ресурсов задачам проекта, эффективно контролировать загрузку ресурсов.

Серверная часть должна быть разработана на языке PHP. С ее помощью будет осуществляться взаимодействие с базой данных с помощью отправки SQL запросов, а также браться актуальные задания.

Данные будут храниться в БД, управляемой с помощью СУБД MySQL. Это свободная реляционная система управления базами данных.

Разработка мобильного приложения должна производиться с помощью платформы Flutter. В качестве языка программирования будет использоваться Dart, который компилируется в бинарный код, за счет чего достигается высокая скорость выполнения операций, сравнимая с Objective-C, Swift, Java и Kotlin. Одной из главных особенностей данной платформы является

встроенный hot-reload. Hot-reload позволяет загружать измененные участки кода и перестраивать дерево виджетов приложения, что занимает гораздо меньше времени, чем обычная сборка приложения [17].

Также особенностью разработки Flutter-приложений является использование виджетов при создании пользовательского интерфейса. Виджеты являются аналогами классов.

Тестирование мобильного приложения производилось на смартфоне Samsung Galaxy S8.

2.2 Проектирование архитектуры программного обеспечения

Для реализации приложения была выбрана двухуровневая архитектура, поскольку данные пользователей, задания, результаты их выполнения и теория должны храниться на сервере, который предоставляет доступ к ним. Клиентская часть – мобильное приложение, которое взаимодействует с сервером и предоставляет удобный интерфейс для отображения данных и выполнения заданий.

Преимуществами данной архитектуры является то, что все данные хранятся на сервере, что позволяет использовать приложение с разных устройств и не терять данные. Также это позволяет изменять список заданий и теории без запросов на обновление версии приложения. Еще это позволяет избежать дублирования кода при использовании нескольких видов представления данных: мобильного приложения, веб интерфейса, десктопного приложения.

2.3 Планирование разработки и оценка бюджета

Для оценки трудоемкости, а также стоимости проекта необходимо провести его грамотное планирование. Для этого был определен список задач, решение которых необходимо для создания приложения, и перечень ресурсов, с помощью которых эти задачи будут решаться.

В таблице 1 представлены задачи, представляющие из себя этапы создания проекта.

Таблица 1 – Перечень задач проекта

№	Название задачи	Длительность	Предшественники
1	Начало проекта	-	-
2	Анализ требований	-	-
3	Поиск литературы (учебников английского языка и учебных пособий)	7 дней	1
4	Составление списка заданий для тренировки и для теста	4 дня	3
5	Анализ списка заданий	2 дня	4
6	Консультация с учителями английского языка	1 день	5
7	Принятие окончательного решения	1 день	6
8	Анализ требований завершен	-	7
9	Выбор средств реализации проекта	-	-
10	Анализ возможных программных средств	4 дня	1
11	Выбор программного средства для реализации	2 дня	10
12	Выбор средств реализации завершен	-	11
13	Реализация проекта разработки мобильного приложения	-	-
14	Проектирование пользовательского интерфейса	-	-
15	Проектирование интерфейса меню	2 дня	8;12
16	Проектирование интерфейса заданий	3 дня	8;12

Продолжение таблицы 1

№	Название задачи	Длительность	Предшественники
17	Проектирование интерфейса для изучения материалов	2 дня	8;12
18	Проектирование интерфейса завершено	-	15;16;17
19	Программирование сопутствующих и обслуживающих классов	20 дней	18
20	Реализация проекта разработки мобильного приложения завершена	-	19
21	Отладка	-	-
22	Тестирование заданий для тренировки и тестов	5 дней	20
23	Тестирование работы сопутствующих классов	5 дней	20
24	Исправление ошибок	5 дней	22;23
25	Отладка завершена	-	24
26	Внедрение программного средства в общий проект	-	-
27	Внедрение программного средства	4 дня	25
28	Поиск и исправление ошибок при внедрении	5 дней	27
29	Внедрение завершено	-	28
30	Конец проекта	-	29

По данной таблице можно сделать вывод, что для реализации программного проекта нужно выполнить 30 задач.

В таблице 2 представлен список затрачиваемых ресурсов, распределение которых приведено в таблице 3.

Таблица 2 – Список ресурсов проекта

Название ресурса	Тип	Стандартная ставка	Ставка сверхурочных
Программист1	Трудовой	500 Р\ч	1200 Р\час
Программист2	Трудовой	500 Р\ч	1200 Р\час
Программист3	Трудовой	500 Р\ч	1200 Р\час
Интернет	Затраты	3Р/Гб	-
Компьютер	Материальны	40000 Р	-
Android-устройство	Материальный	6000 Р	-

Таблица 3 – Назначения ресурсов

№	Название задачи	Ресурс	Единицы
1	Начало проекта		
2	Анализ требований		
3	Поиск литературы (учебников английского языка и учебных пособий)	Программист1 Компьютер Интернет	100% 1 шт
4	Составление списка заданий для тренировки и для теста	Программист1 Компьютер Интернет	100% 1 шт
5	Анализ списка заданий	Программист1 Программист3 Компьютер Интернет	50% 100% 1 шт
6	Консультация с учителями английского языка	Программист3	100%
7	Принятие окончательного решения	Программист1 Программист2 Программист3	100% 100% 100%

Продолжение таблицы 3

№	Название задачи	Ресурс	Единицы
8	Анализ требований завершен		
9	Выбор средств реализации проекта		
10	Анализ возможных программных средств	Программист2 Компьютер Интернет	100% 1 шт
11	Выбор программного средства для реализации	Программист2 Компьютер Интернет	100% 1 шт
12	Выбор средств реализации завершен		
13	Реализация проекта		
14	Проектирование пользовательского интерфейса		
15	Проектирование интерфейса меню	Программист1 Компьютер Интернет	100% 1 шт
16	Проектирование интерфейса заданий	Программист2 Компьютер Интернет	100% 1 шт
17	Проектирование интерфейса для изучения материалов	Программист3 Компьютер Интернет	100% 1 шт
18	Проектирование интерфейса завершено		
19	Программирование сопутствующих и обслуживающих классов	Программист1 Программист2 Программист3 Компьютер Интернет	100% 100% 100% 3 шт

Окончание таблицы 3

№	Название задачи	Ресурс	Единицы
20	Реализация проекта завершена		
21	Отладка		
22	Тестирование заданий для тренировки и тестов	Программист2 Компьютер Android-устройство	100% 1 шт 1 шт
23	Тестирование работы сопутствующих классов	Программист3 Компьютер Android-устройство	100% 1 шт 1 шт
24	Исправление ошибок	Программист2 Программист3 Компьютер	100% 100% 2 шт
25	Отладка завершена		
26	Внедрение программного средства в общий проект		
27	Внедрение программного средства	Программист1 Компьютер Интернет	100% 1 шт
28	Поиск и исправление ошибок при внедрении	Программист1 Компьютер Интернет	100% 1 шт
29	Внедрение завершено		
30	Конец проекта		

По данным таблицам видно, что в решении задач будет задействовано 3 программиста, которым для этого необходимы компьютеры, доступ к интернету и Android-устройству для тестирования приложения.

Сперва вручную просчитаем время, которое необходимо для реализации проекта, после чего определим критический путь. Для этого построим сетевой график проекта. Он представлен на рисунке Б.1.

Для вычисления критического пути необходимо для каждой задачи определить раннее и позднее время начала работ. Раннее время определяется по формуле:

$$T_{p(i)} = \max_{j \in G} (T_{p(j)} + t_j),$$

где t_i - время выполнения работы i , G – множество работ, непосредственно предшествующих работе i , H – множество работ, непосредственно следующих за работой i .

Позднее время начала вычисляется по формуле:

$$T_{п(i)} = \min_{j \in H} (T_{п(j)} - t_j).$$

Подробное вычисление раннего времени начала работ представлено в таблице 4.

Таблица 4 – Вычисление раннего времени начала работ

Работа	Предшественники			Вычисление	Раннее время начала
	№	Раннее время начала	Длительность		
1	-	-	-	-	0
3	1	0	0	0+0	0
4	3	0	7	0+7	7
5	4	7	4	7+4	11
6	5	11	2	11+2	13
7	6	13	1	13+1	14
8	7	14	1	14+1	15
10	1	0	0	0+0	0
11	10	0	4	0+4	4
12	11	4	2	4+2	6
15	8	15	0	Max(15+0;6+0)	15
	12	6	0		
16	8	15	0	Max(15+0;6+0)	15
	12	6	0		

Продолжение таблицы 4

Работа	Предшественники			Вычисление	Раннее время начала
	№	Раннее время начала	Длительность		
17	8	15	0	Max(15+0;6+0)	15
	12	6	0		
18	15	15	2	Max(15+2;15+3;15+2)	18
	16	15	3		
	17	15	2		
19	18	18	0	18+0	18
20	19	18	20	18+20	38
22	20	38	0	38+0	38
23	20	38	0	38+0	38
24	22	38	5	Max(38+5;38+5)	43
	23	38	5		
25	24	43	5	43+5	48
27	25	48	0	48+0	48
28	26	48	4	48+4	52
29	28	52	5	52+5	57
30	29	57	0	57+0	57

Исходя из данных вычислений, следует, что выполнение проекта должно занять 57 рабочих дней. Теперь необходимо вычислить время позднего начала работ. Оно представлено в таблице 5.

Таблица 5 – Вычисление позднего времени начала работ

Работа	Последователи		Длительность	Вычисление	Позднее время начала
	№	Позднее время начала			
30	-	-	-	-	57
29	28	57	0	57-0	57
28	29	57	5	57-5	52

Продолжение таблицы 5

Работа	Последователи		Длительность	Вычисление	Позднее время начала
	№	Позднее время начала			
27	28	52	4	52-4	48
25	27	48	0	48-0	48
24	25	48	5	48-5	43
23	24	43	5	43-5	38
22	24	43	5	43-5	38
20	22	38	0	Min(38-0;38-0)	38
	23	38	0		
19	20	38	20	38-20	18
18	19	18	0	18-0	18
17	18	18	2	18-2	16
16	18	18	3	18-3	15
15	18	18	2	18-2	16
12	15	16	0	Min(16-0;15-0;16-0)	15
	16	15			
	17	16			
11	12	15	2	15-2	13
10	11	13	4	13-4	9
8	15	16	0	Min(16-0;15-0;16-0)	15
	16	15			
	17	16			
7	8	15	1	15-1	14
6	7	14	1	14-1	13
5	6	13	2	13-2	11
4	5	11	4	11-4	7
3	4	7	7	7-7	0
1	3	0	0	Min(0-0;10-0)	0
	10	10			

Далее определим критический путь для нашего проекта, вычисление которого представлено в таблице 6.

Таблица 6 – Вычисление критического пути

Работа №	Раннее время начала	Позднее время начала	Резерв времени
1	0	0	0
3	0	0	0
4	7	7	0
5	11	11	0
6	13	13	0
7	14	14	0
8	15	15	0
10	0	9	9
11	4	13	9
12	6	15	9
15	15	16	1
16	15	15	0
17	15	16	1
18	18	18	0
19	18	18	0
20	38	38	0
22	38	38	0
23	38	38	0
24	43	43	0
25	48	48	0
27	48	48	0
28	52	52	0
29	57	57	0
30	57	57	0

Из этой таблицы мы можем определить, что критический путь проходит через работы, с номерами 1, 3, 4, 5, 6, 7, 8, 16, 18, 19, 20, 22 (23), 24, 25, 27, 28,

29, 30, так как данные работы имеют нулевой резерв времени. Его можно наглядно проследить на рисунке Б.2.

На рисунке 17 изображено введенное в Project содержание таблицы 1 с вычисленными датами начала и завершения каждой задачи. За начало проекта была принята дата 03.02.2020.

Название задачи ▼	Длитель ▼	Начало ▼	Окончание ▼
Начало проекта	0 дней	Пн 03.02.20	Пн 03.02.20
▲ Анализ требований	15 дней	Пн 03.02.20	Пн 24.02.20
Поиск литературы (учебников английского языка и учебных пособий)	7 дней	Пн 03.02.20	Вт 11.02.20
Составление списка заданий для тренировки и для теста	4 дней	Ср 12.02.20	Пн 17.02.20
Анализ списка заданий	2 дней	Вт 18.02.20	Ср 19.02.20
Консультация с учителями английского языка	1 день	Чт 20.02.20	Чт 20.02.20
Принятие окончательного решения	1 день	Пт 21.02.20	Пт 21.02.20
Анализ требований завершен	0 дней	Пн 24.02.20	Пн 24.02.20
▲ Выбор средств реализации проекта	6 дней	Пн 24.02.20	Вт 03.03.20
Анализ возможных программных средств	4 дней	Пн 24.02.20	Чт 27.02.20
Выбор программного средства для реализации	2 дней	Пт 28.02.20	Пн 02.03.20
Выбор средств реализации завершен	0 дней	Вт 03.03.20	Вт 03.03.20
▲ Реализация проекта	23 дней	Пн 24.02.20	Чт 26.03.20
▲ Проектирование пользовательского интерфейса	3 дней	Пн 24.02.20	Чт 27.02.20
Проектирование интерфейса меню	2 дней	Пн 24.02.20	Вт 25.02.20
Проектирование интерфейса заданий	3 дней	Пн 24.02.20	Ср 26.02.20
Проектирование интерфейса для изучения материалов	2 дней	Пн 24.02.20	Вт 25.02.20
Проектирование интерфейса завершено	0 дней	Чт 27.02.20	Чт 27.02.20
Программирование сопутствующих и обслуживающих классов	20 дней	Чт 27.02.20	Ср 25.03.20
Реализация проекта завершена	0 дней	Чт 26.03.20	Чт 26.03.20
▲ Отладка	10 дней	Чт 26.03.20	Чт 09.04.20
Тестирование заданий для тренировки и тестов	5 дней	Чт 26.03.20	Ср 01.04.20
Тестирование работы сопутствующих классов	5 дней	Чт 26.03.20	Ср 01.04.20
Исправление ошибок	5 дней	Чт 02.04.20	Ср 08.04.20
Отладка завершена	0 дней	Чт 09.04.20	Чт 09.04.20
▲ Внедрение программного средства в общий проект	9 дней	Чт 09.04.20	Ср 22.04.20
Внедрение программного средства	4 дней	Чт 09.04.20	Вт 14.04.20
Поиск и исправление ошибок при внедрении	5 дней	Ср 15.04.20	Вт 21.04.20
Внедрение завершено	0 дней	Ср 22.04.20	Ср 22.04.20
Конец проекта	0 дней	Ср 22.04.20	Ср 22.04.20

Рисунок 17 – Задачи проекта в среде Microsoft Project 2019

На рисунке 18 представлена построенная программой диаграмма Ганта.

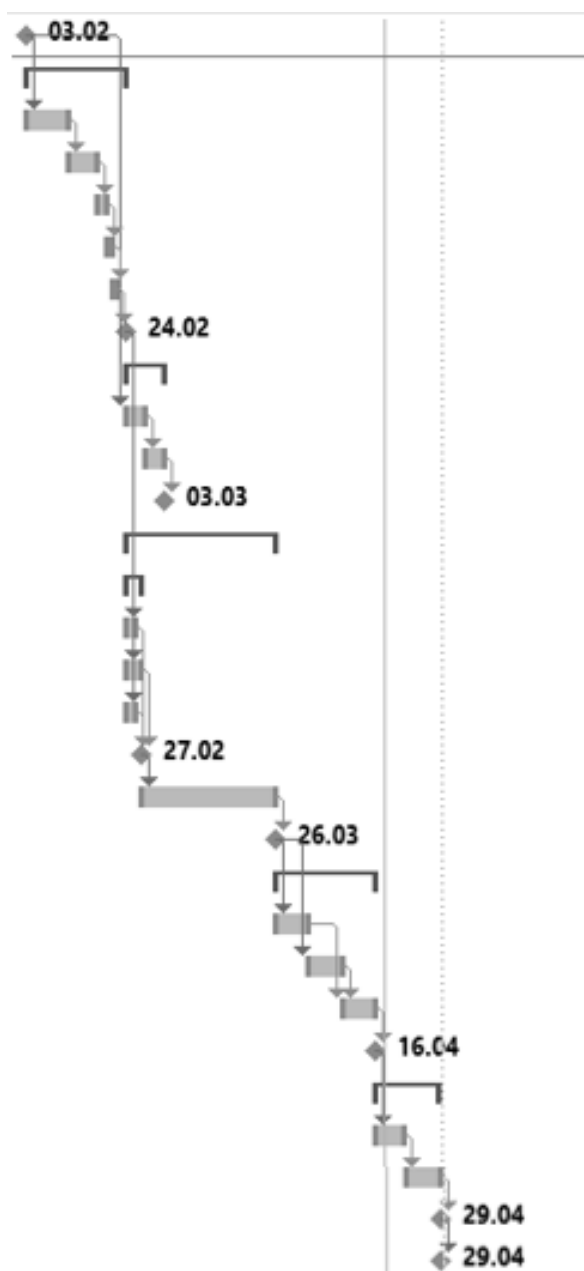


Рисунок 18 – Диаграмма Ганта

По данным рисункам видно, что критический путь, вычисленный программой, проходит по задачам 1, 3, 4, 5, 6, 7, 8, 16, 18, 19, 20, 22 (23), 24, 25, 27, 28, 29, 30. Таким образом, построенный программой график и определенный ею критический путь совпадают с теми, что были вычислены вручную.

На рисунке 19 изображен обзор проекта, на котором можно увидеть подсчитанную с помощью Microsoft Project 2019 длительность реализации проекта – 57 дней, и предполагаемые затраты – 606000 руб.

	Начало	Окончание
Текущее	Пн 03.02.20	Ср 22.04.20
Базовое	НД	НД
Фактическое	НД	НД
Отклонение	0д	0д

	Длительность	Трудозатраты	Затраты
Текущие	57д	960ч	606 000,00 Р
Базовые	0д	0ч	0,00 Р
Фактические	0д	0ч	0,00 Р
Оставшиеся	57д	960ч	606 000,00 Р

Рисунок 19 – Обзор проекта

Далее приведем графики ресурсов. На рисунке 20 представлен график для материального ресурса «Компьютер».

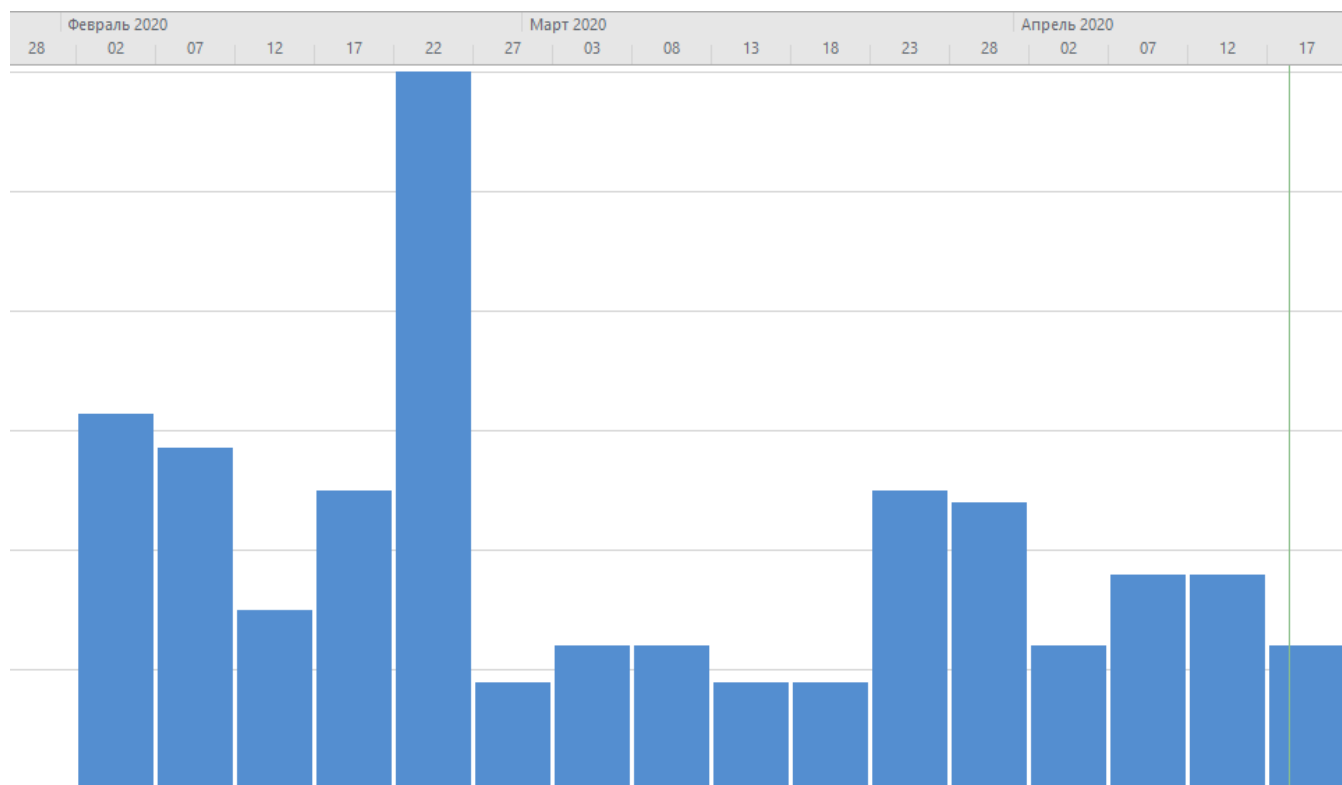


Рисунок 20 –График ресурсов для ресурса «Компьютер»

На рисунках 21, 22 представлены графики для трудовых ресурсов «Программист1» и «Программист2» соответственно. Если на графике присутствуют красные столбцы, то данный ресурс перерабатывает.

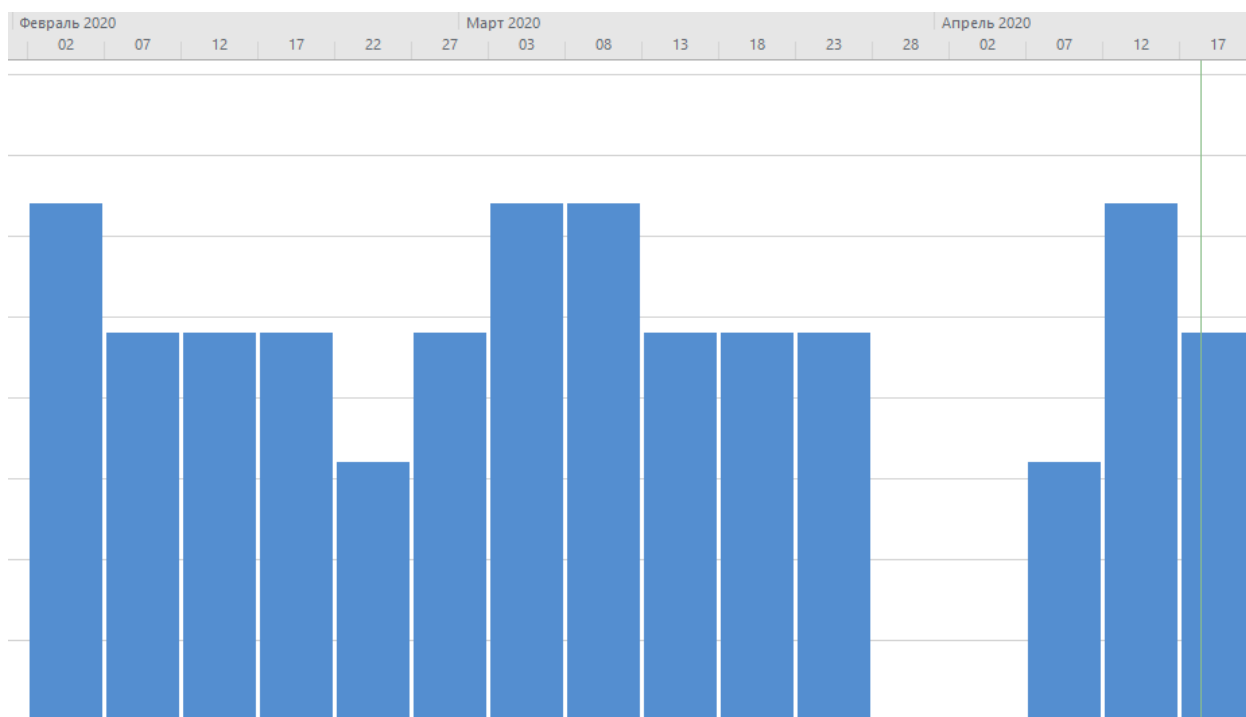


Рисунок 21 – График ресурсов для ресурса «Программист1»

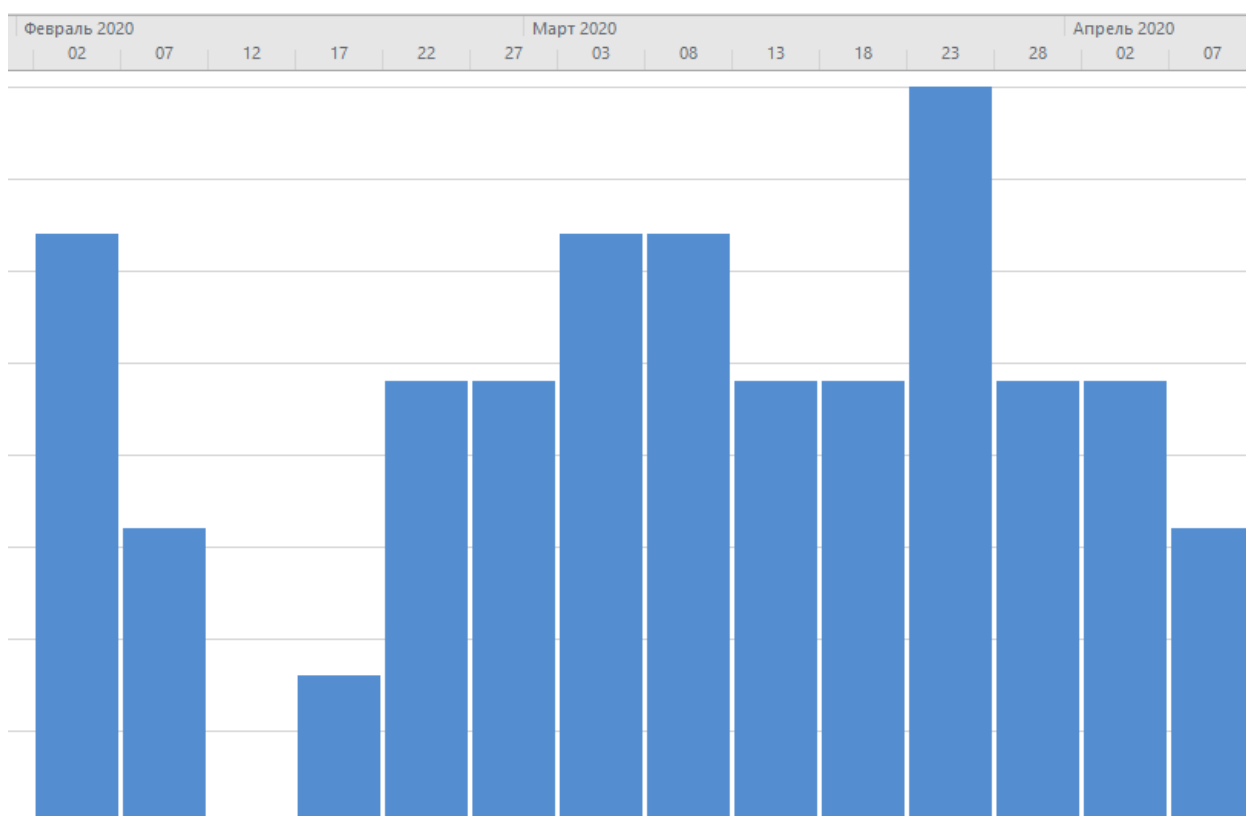


Рисунок 22 – График ресурсов для ресурса «Программист2»

На рисунке 23 представлен график для трудового ресурса «Программист3».

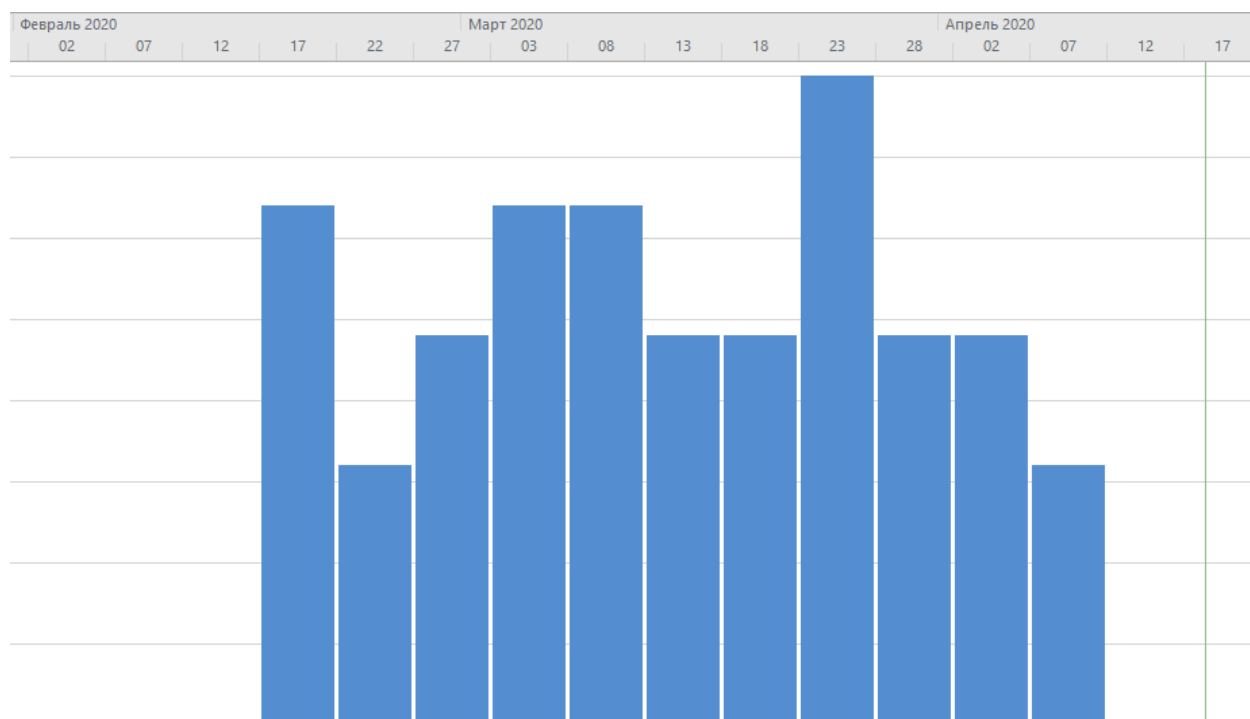


Рисунок 23 – График ресурсов для ресурса «Программист3»

Из этих рисунков видно, что нагрузка на ресурсы не превышает нормальную.

2.4 Проектирование интерфейса мобильного приложения

На основе анализа предметной области и функциональных требований были выделены следующие экраны:

- экран входа в аккаунт;
- экран выбора типа задания;
- экран выбора раздела теории;
- экран настроек;
- экран выбора номера задания;
- экран с заданием;
- экран, сообщающий результат выполнения задания.

Главное меню составляют 3 экрана: выбора типа задания, выбора раздела теории и настроек. По ним пользователь может перемещаться с помощью области навигации в нижней части экрана. Разработанные прототипы экрана настроек представлен на рисунке 24.

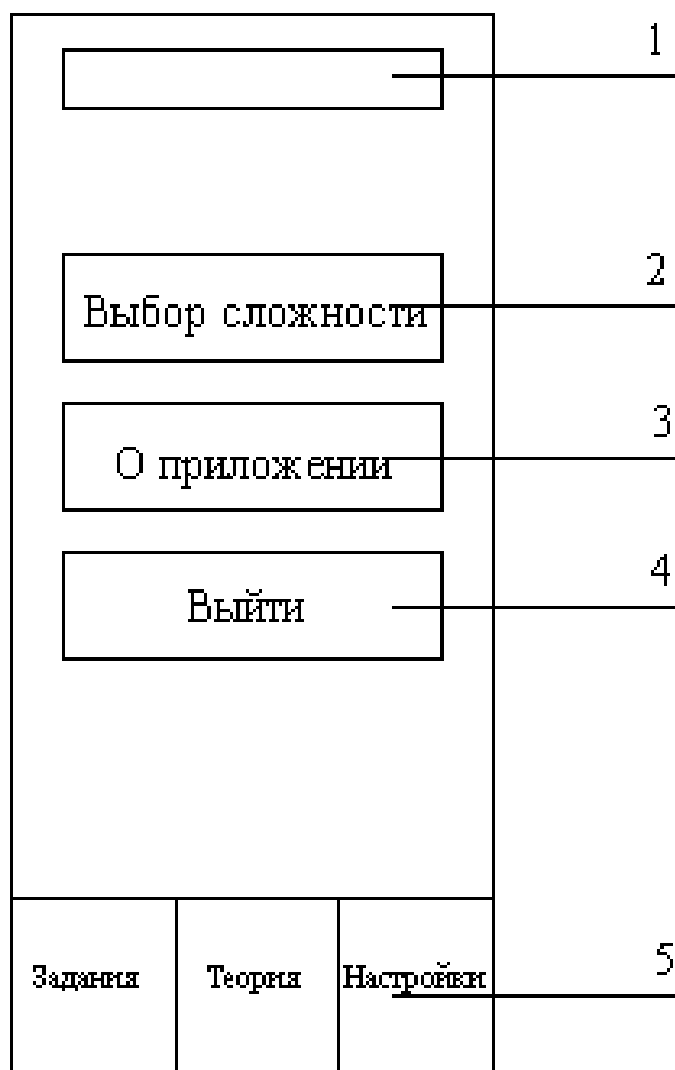


Рисунок 24 – Прототип экрана настроек

Цифрами на рисунке 24 отмечены:

- 1 – поле вывода информации об аккаунте пользователя;
- 2 – кнопка выбора уровня сложности;
- 3 – кнопка информации о приложении;
- 4 – кнопка выхода из аккаунта;
- 5 – область навигации по главному меню.

На рисунке 25 представлен прототип экрана выбора типа задания.

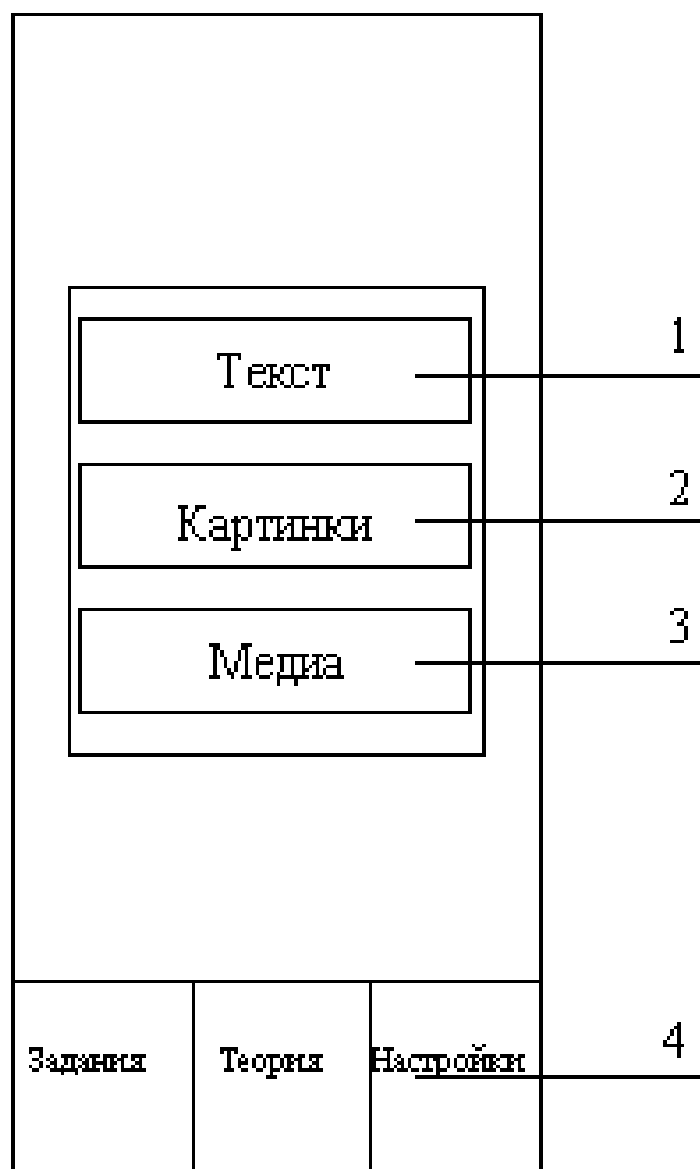


Рисунок 25 – Прототип экрана выбора типа задания

На рисунке 25 отмечено:

- 1 – кнопка выбора текстовых заданий;
- 2 – кнопка выбора заданий на идентификацию изображения;
- 3 – кнопка выбора заданий с аудио/видео;
- 4 – область навигации по главному меню.

На рисунке 26 представлен прототип экрана выбора раздела теории.

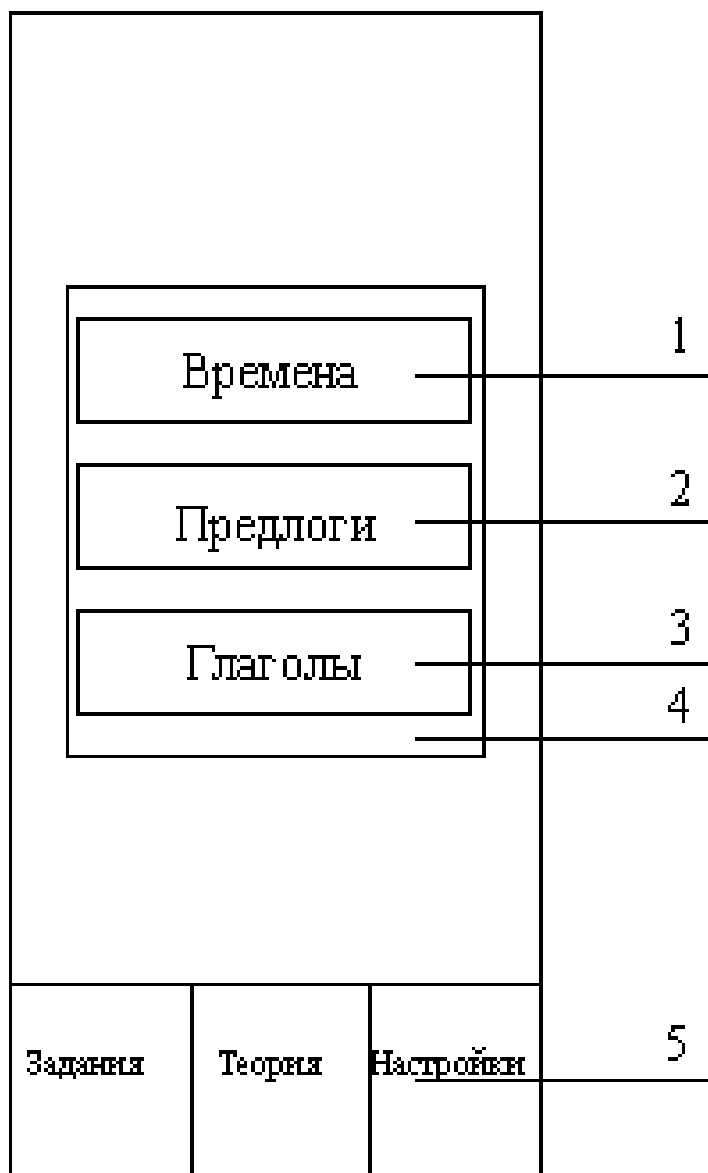


Рисунок 26 – Прототип экрана выбора раздела теории

Рассмотрим основные элементы данного экрана:

- 1 – кнопка выбора раздела теории «Времена»;
- 2 – кнопка выбора раздела теории «Предлоги»;
- 3 – кнопка выбора раздела теории «Глаголы»;
- 4 – область, содержащая список кнопок, выбирающих разделы теории;
- 5 – область навигации по главному меню.

После выбора типа перед пользователем появляется экран выбора номера задания. На рисунке 27 изображен его прототип.

После выбора номера появляется экран с самим заданием. Он состоит из 4-х уровней. Прототип данного экрана изображен на рисунке 28. Перед

переходом к следующему уровню пользователю выводится правильный ответ в данном.

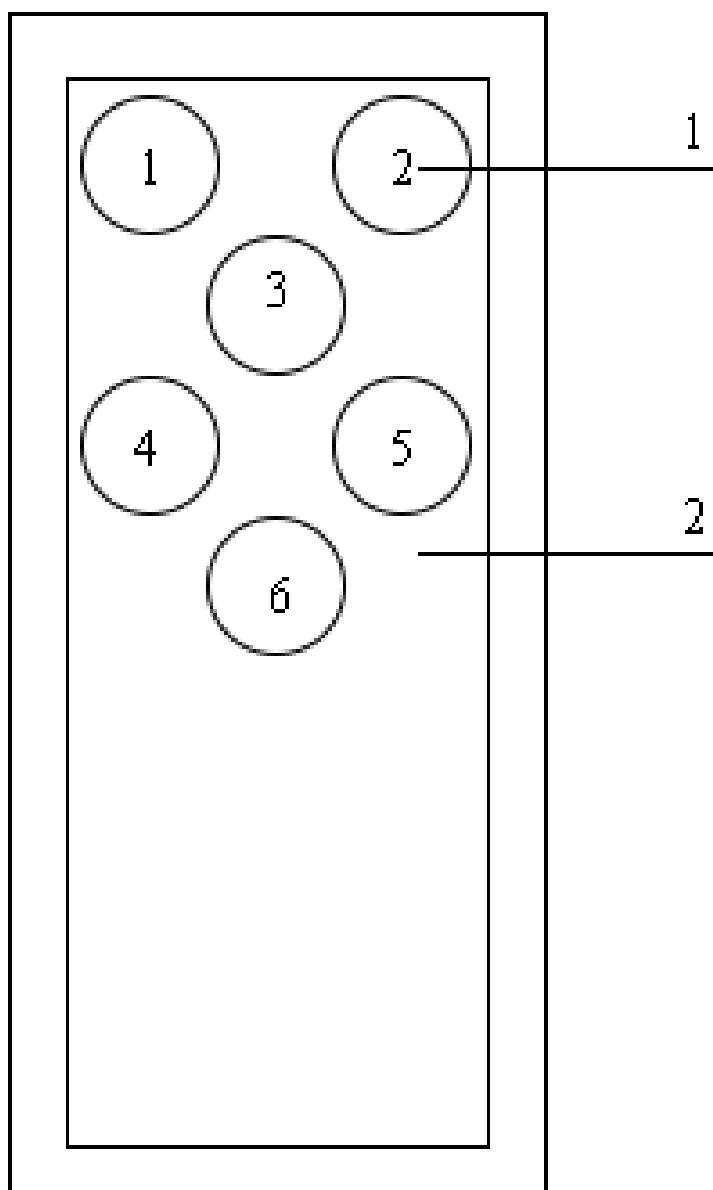


Рисунок 27 – Прототип экрана выбора номера задания

На рисунке 27 отмечены:

- 1 – кнопка выбора номера задания;
- 2 – область, содержащая список всех заданий.

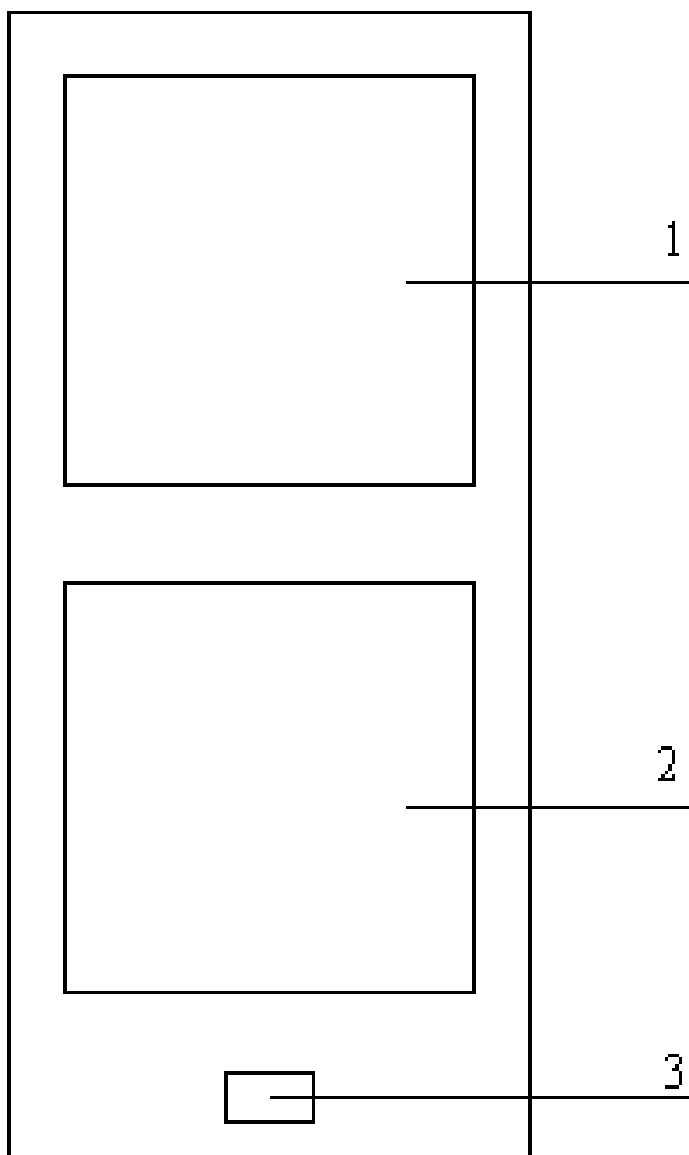


Рисунок 28 – Прототип экрана задания

На рисунке 28 цифрами отмечены:

1 – область вывода текущего задания;

2 – область ввода ответа;

3 – кнопка перехода к следующему уровню в задании.

Когда пользователь завершает выполнение 4-го уровня в задании, появляется экран результатов, прототип которого представлен на рисунке 29. Он выводит количество правильных ответов.

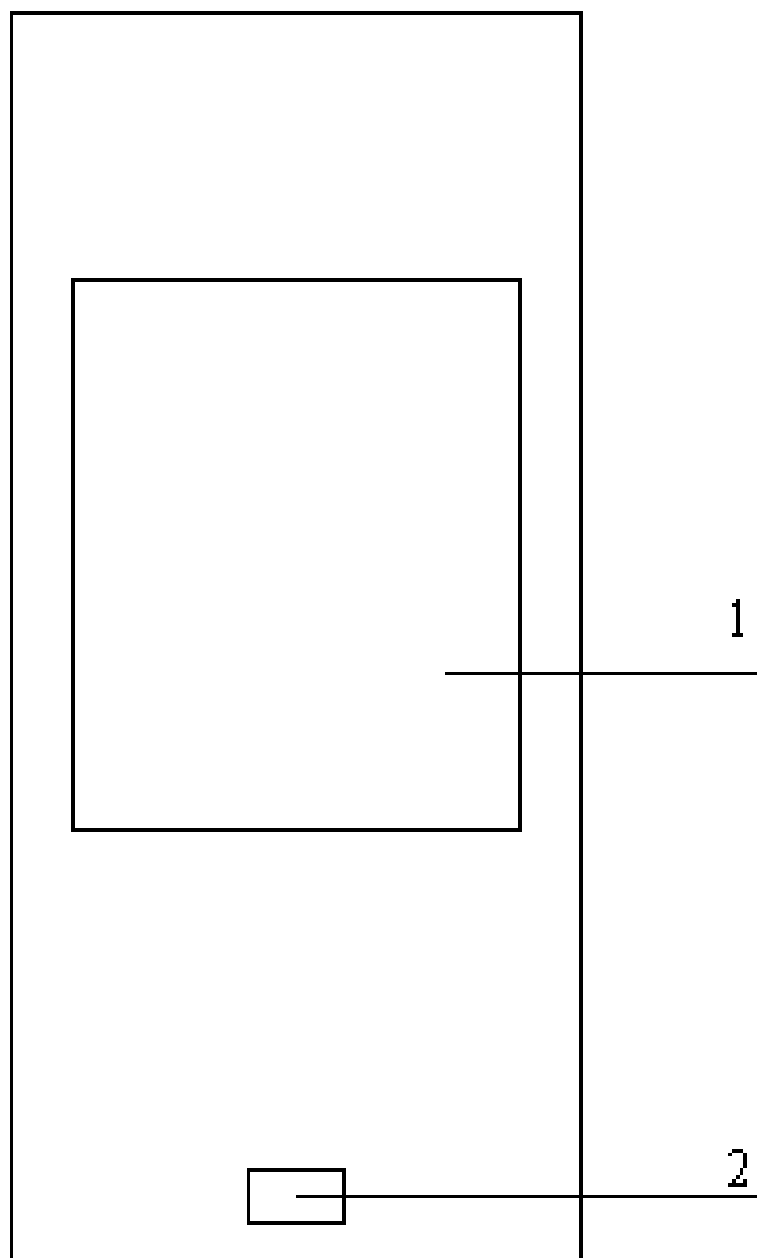


Рисунок 29 – Прототип экрана вывода результатов выполнения задания

Далее представлены основные элементы данного экрана:

1 – область вывода результатов выполнения задания;

2 – кнопка возврата к экрану выбора номера задания.

Экран входа в аккаунт является первым, который пользователь видит при запуске приложения. Его прототип представлен на рисунке 30.

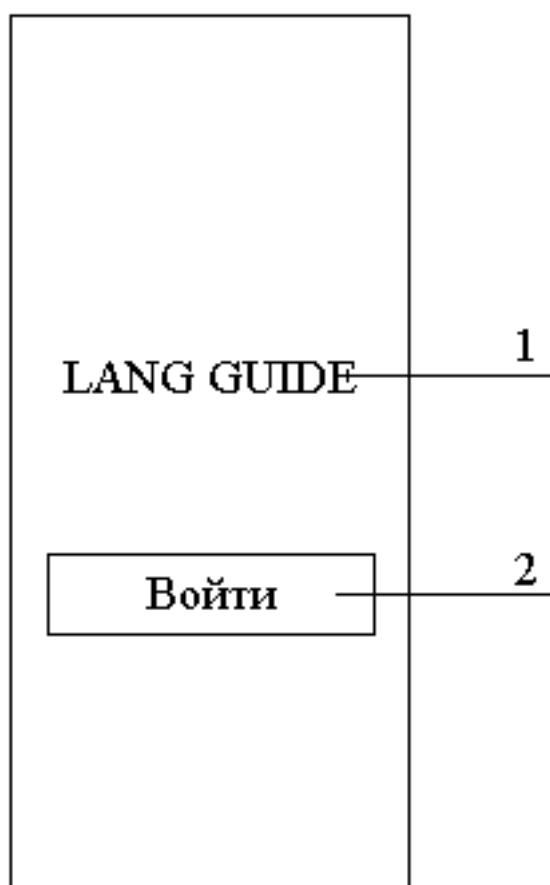


Рисунок 30 – Прототип экрана входа в аккаунт

Рассмотрим основные элементы данного экрана:

- 1 – название приложения;
- 2 – кнопка входа в аккаунт.

Спроектированные прототипы в полной мере описывают то, как должен выглядеть пользовательский интерфейс разрабатываемого мобильного приложения.

2.5 Проектирование серверной части приложения

Для хранения данных выбрана СУБД MySQL. Это реляционная система управления базами данных с открытым исходным кодом. В настоящее время она является одной из самых популярных в веб-приложениях, и почти все веб-фреймворки поддерживают ее при базовой конфигурации, то есть без

дополнительных модулей. Из основных преимуществ MySQL можно выделить гибкость и простоту использования, масштабируемость и производительность [22].

Серверная часть приложения написана на языке PHP. Это широко используемый язык сценариев общего назначения с открытым кодом, специально разработанный для исполнения их на web-сервере [23].

2.5.1 Логическое проектирование базы данных

На основе анализа предметной области, а также функциональных требований была построена исходная концептуальная модель. Были выделены типы сущностей, их свойства и типы связей между сущностями.

Типы сущности предметной области представлены в таблице 7.

Таблица 7 – Типы сущностей

№	Наименование	Описание	Категория	Количество экземпляров	Режим обновления
1	Пользователь	Зарегистрированные пользователи приложения	Стержневая	2000	10% в месяц
2	Результаты легкого уровня	Список результатов выполнения заданий легкого уровня сложности	Стержневая	40000	25% в месяц
3	Результаты среднего уровня	Список результатов выполнения заданий среднего уровня сложности	Стержневая	40000	25% в месяц
4	Результаты тяжелого уровня	Список результатов выполнения заданий тяжелого уровня сложности	Стержневая	40000	25% в месяц

Типы связей между типами сущностей предметной области представлены в таблице 8.

Таблица 8 – Типы связей

№	Тип сущности 1	Тип сущности 2	Описание	Мощность	Обязательность	Ограничения
1	Результаты легкого уровня	Пользователь	Результаты выполнения заданий легкого уровня сложности относятся к конкретному пользователю	М : 1	Обязательная (задание выполняется конкретным пользователем)	
2	Результаты среднего уровня	Пользователь	Результаты выполнения заданий среднего уровня сложности относятся к конкретному пользователю	М : 1	Обязательная (задание выполняется конкретным пользователем)	
3	Результаты тяжелого уровня	Пользователь	Результаты выполнения заданий тяжелого уровня сложности относятся к конкретному пользователю	М : 1	Обязательная (задание выполняется конкретным пользователем)	

Графическое представление концептуальной модели базы данных в виде диаграммы «сущность-связь» представлено на рисунке 31.

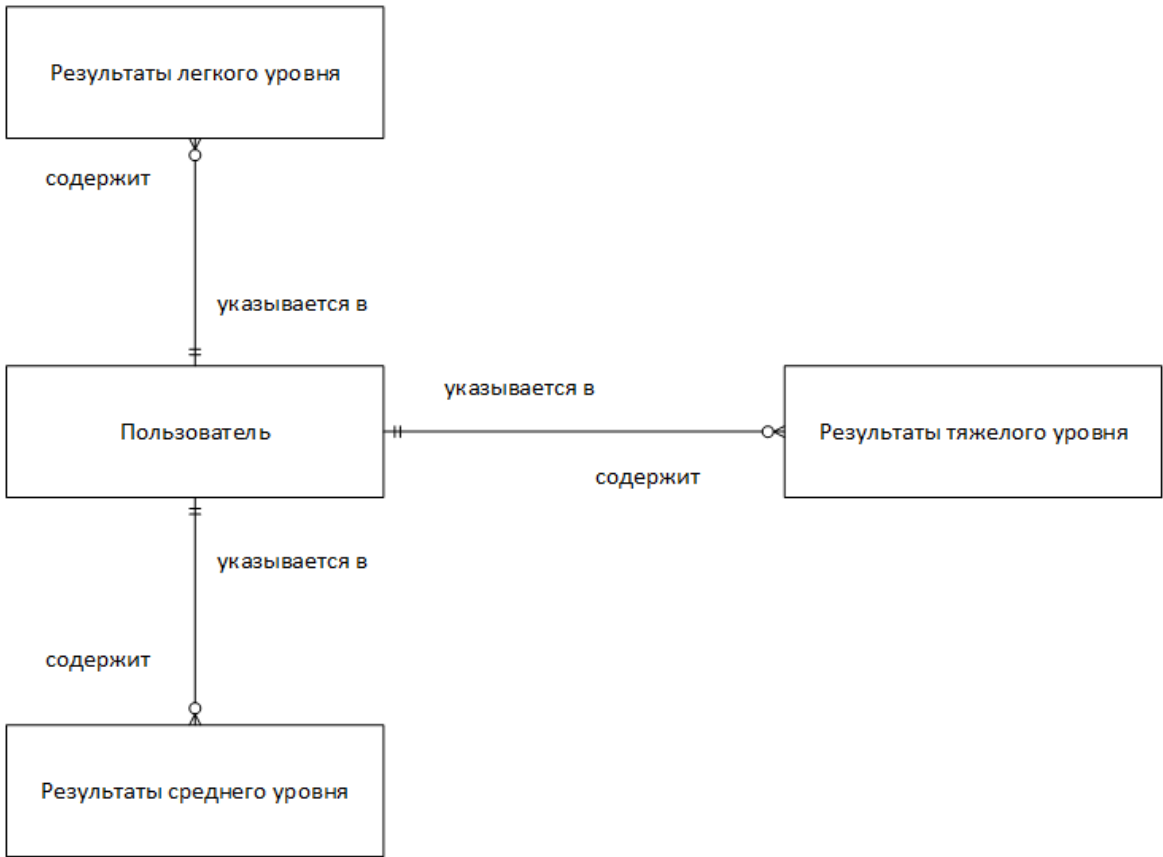


Рисунок 31 – Диаграмма «сущность-связь»

Проектирование базы данных осуществлялось с помощью методологии IDEF1X. Логическая модель базы данных представлена на рисунке 32.

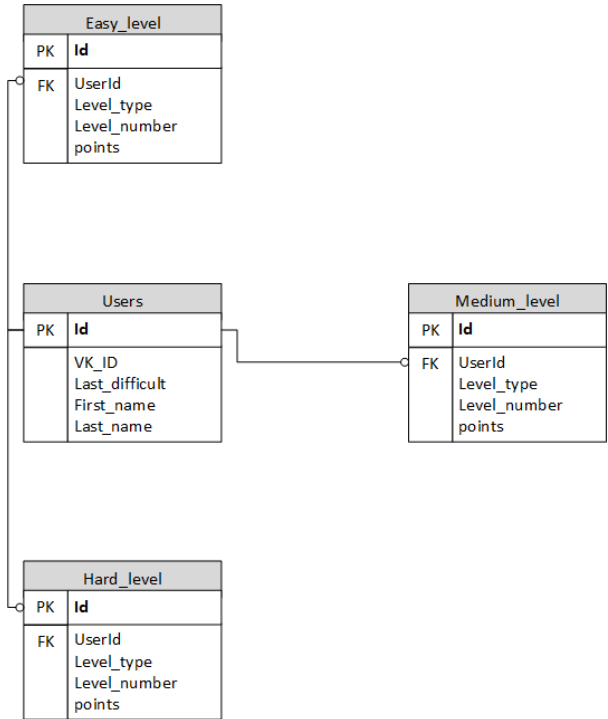


Рисунок 32 – Логическая модель базы данных

С помощью логического проектирования была отображена концептуальная модель базы данных вместе с объектами, используемыми в модели данных.

3 Реализация приложения

3.1 Реализация серверной части приложения

3.1.1 Реализация базы данных

Физическое моделирование базы данных проводилось в среде ErWin Data Modeler. Она была разработана на основе логической модели предметной области. Физическая модель представлена на рисунке 33.

Развертывание базы данных было выполнено запуском SQL скрипта с помощью средства администрирования phpMyAdmin [24]. Данное средство представляет интуитивный веб-интерфейс для управления базами данных MySQL.

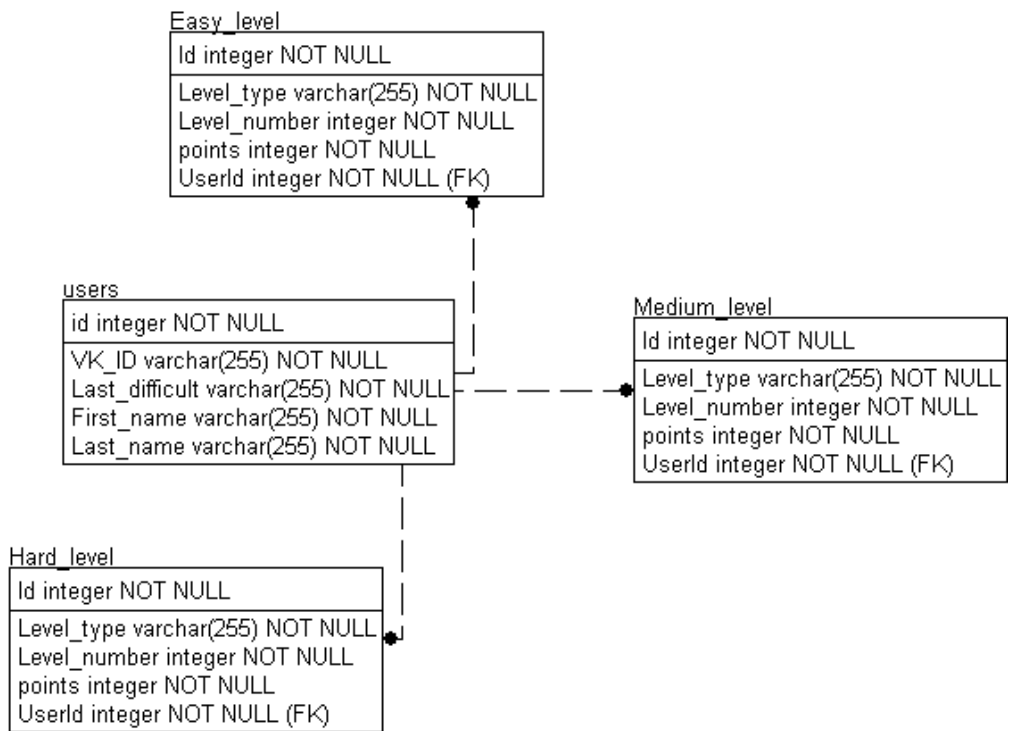


Рисунок 33 – Физическая модель базы данных

Исходный код SQL скрипта создания базы данных представлен в приложении В. База данных была развернута на локальном сервере Apache.

3.1.2 Разработка серверной части приложения

Для реализации сценариев работы сервера был использован язык программирования PHP.

Была реализована обработка всех спроектированных запросов. В теле каждого запроса содержится заголовок, с помощью которого выбирается нужный сценарий. Сценарии, как и запросы, можно условно разделить на две группы: запросы к базе данных и запросы на получение другой информации. К последним относятся: запросы на получение заданий и запросы на получение теории.

Сценарии для обращения к базе данных содержат схожий алгоритм действий: получение остальных данных из тела запроса, формирование и выполнение соответствующего SQL-запроса, возврат полученных данных, если это необходимо.

Рассмотрим подробнее сценарий для запроса на получение списка выполненных заданий. В теле данного запроса передается следующая информация: идентификатор пользователя, текущий уровень сложности, тип заданий. После получения данного запроса формируется SQL-запрос на выборку данных из таблицы, которая соответствует уровню сложности из запроса. Результат выполнения запроса возвращается в виде массива.

Сценарий для запроса на получение задания возвращает список уровней в соответствующем задании. Номер задания передается в теле запроса. После получения запроса, в зависимости от номера задания, возвращается список уровней в задании.

Полный исходный код серверной части представлен в приложении В.

3.2 Реализация мобильного приложения

Разработка мобильного приложения осуществлялась на языке программирования Dart в среде Android Studio 4.0. Листинг виджетов и классов приведет в приложении Г.

Диаграмма классов приложения представлена в приложении Д.

На рисунке 34 приведена диаграмма компонентов, отражающая общую структуру исходного кода программы.

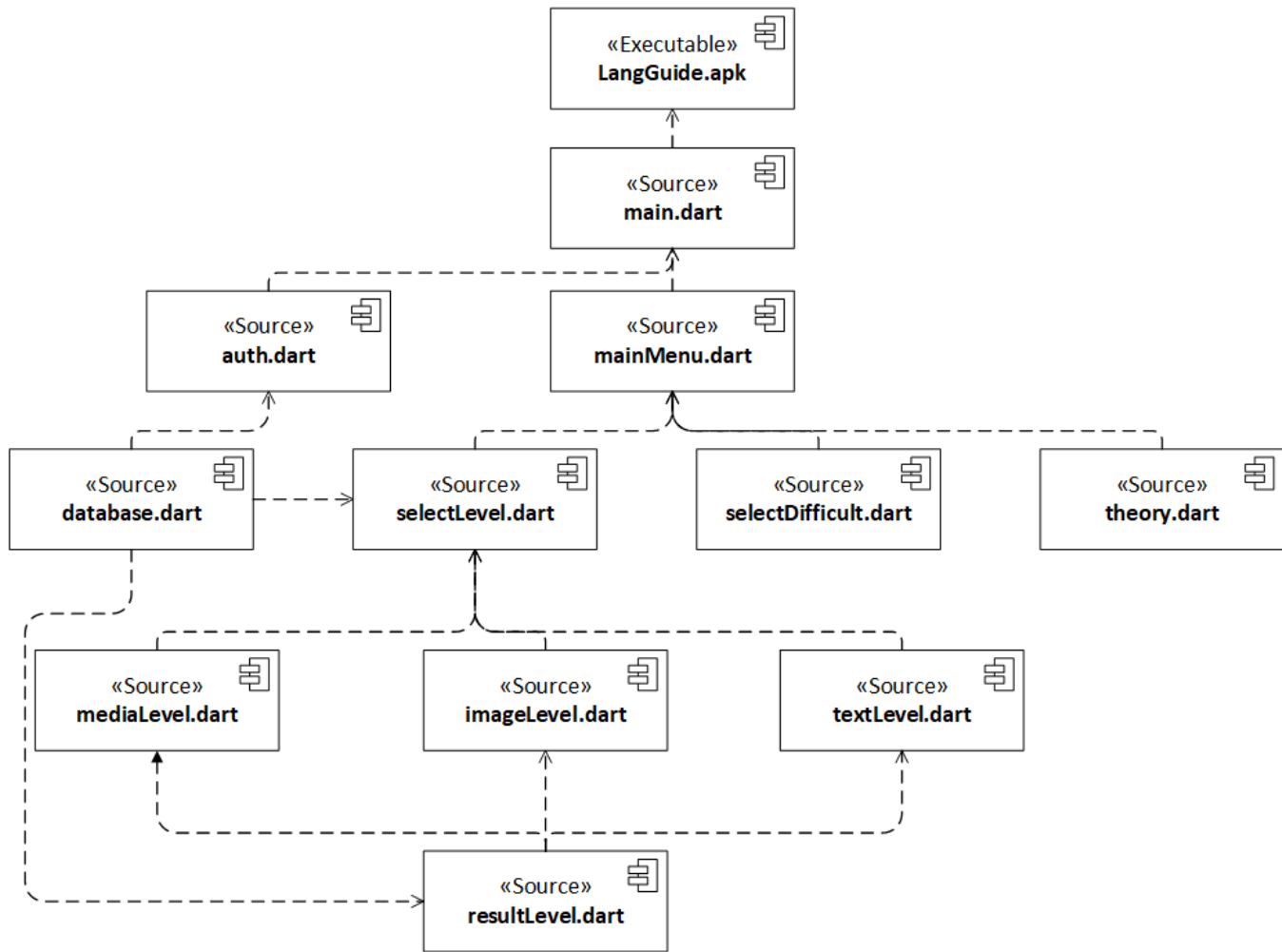


Рисунок 34 – Диаграмма компонентов мобильного приложения

Компонентами данного приложения являются файлы, содержащие исходный код каждого класса и виджета. В итоге все компоненты становятся частью главного класса «main», в котором идет инициализация экранов.

На рисунке 35 представлена диаграмма развертывания, на которой отображено распределение компонентов приложения.

На данном рисунке видно, что клиентская часть приложения работает на мобильных устройствах под управлением операционной системы Android.

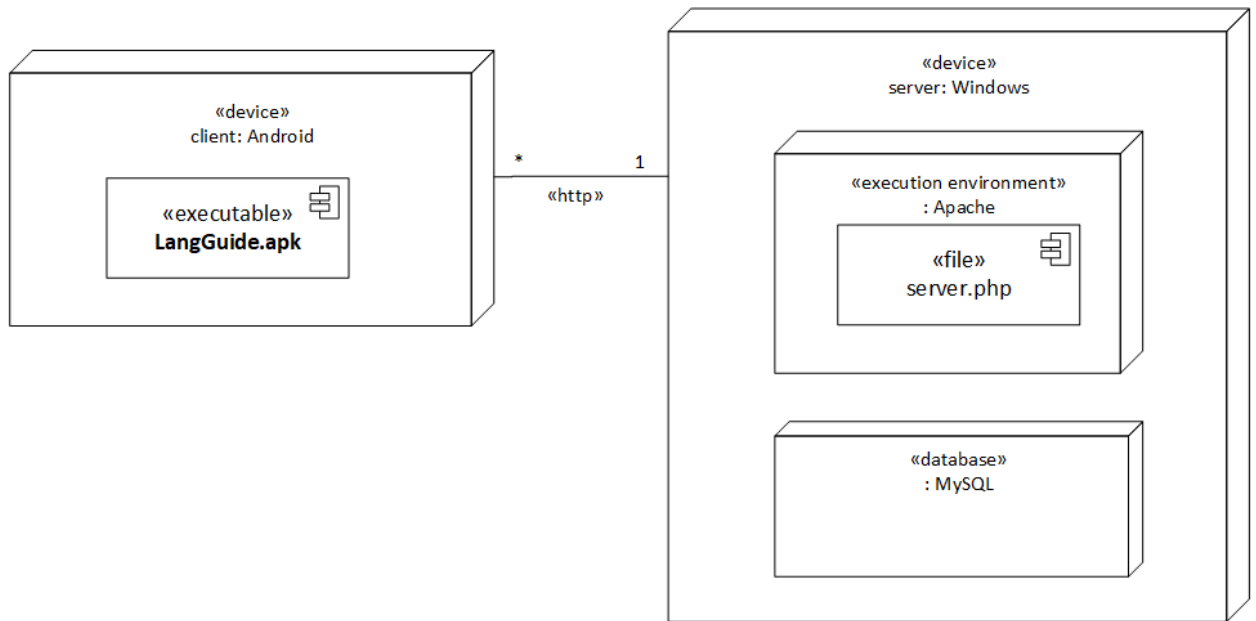


Рисунок 35 – Диаграмма развертывания мобильного приложения

При работе приложение связывается с помощью протокола HTTP с серверной частью, которая развернута на компьютере под управлением операционной системой Windows, на котором еще располагается база данных.

4 Контроль качества программного обеспечения

4.1 Тестирование серверной части

На данном этапе тестирования необходимо убедиться в работы серверной части и исправить ошибки, если это необходимо. Тестирование производится с помощью инструментального средства Postman.

Необходимо протестировать корректность выполнения каждого запроса, который может послать мобильное приложение. Все скриншоты представлены в приложении Е.

Начнем тестирование с отправки запроса на проверку о наличии сведений о пользователе. Так как пользователя с такими данными не существует, то запись о нем добавляется в базу данных, а ответом от сервера будет строка, содержащая название начального уровня сложности. Код ответа – 200 ОК, означающий успешный запрос (рисунок Е.1).

Далее проверим корректность обработки запроса на добавление или обновление данных о выполненном задании. В качестве выполненного задания возьмем текстовый уровень легкого уровня сложности с номером 1. Ответом от сервера будет строка, содержащая статус выполнения запроса. Код ответа – 200 ОК (рисунок Е.2).

Все проведенные выше тесты пройдены. На рисунках видны заголовки запроса к серверу и тело, а для ответа сервера – код ответа HTTP, тело и время выполнения запроса. Подробное описание тестов для всех запросов приведено в таблице 9.

Таблица 9 – Результаты тестирования серверной части

Номер теста	Описание теста	Запрос		Результат
		Метод	URI	
1	Добавление нового пользователя.	POST	DbScript.php/	Тест пройден. Новый пользователь добавлен. Код ответа HTTP – 200 (рисунок Е.1).

Продолжение таблицы 9

2	Добавление данных о выполненном текстовом задании легкого уровня с номером 1.	POST	DbScript.php/	Тест пройден. Добавлены данные о выполнении текстового задания легкого уровня с номером 1. Код ответа HTTP – 200 (рисунок Е.2).
3	Добавление данных о выполненном задании с картинками легкого уровня с номером 2.	POST	DbScript.php/	Тест пройден. Добавлены данные о выполнении задания с картинками легкого уровня с номером 2. Код ответа HTTP – 200 (рисунок Е.3).
4	Добавление данных о выполненном задании с медиа среднего уровня с номером 1.	POST	DbScript.php/	Тест пройден. Добавлены данные о выполнении задания с медиа легкого уровня с номером 1. Код ответа HTTP – 200 (рисунок Е.4).
5	Получение выполненных текстовых заданий легкого уровня сложности.	GET	DbScript.php/?action=GET_COMP_LVL&user_id=236568364&level_difficult=easy_level&level_type=text	Тест пройден. Получены данные о выполненных текстовых заданиях легкого уровня сложности. Код ответа HTTP – 200 (рисунок Е.5).
6	Получение подзаданий для текстового задания легкого уровня под номером 1.	GET	DbScript.php/?action=GET_EASY_TXT_TASK&user_id=236568364&level_type=text&level=1	Тест пройден. Получены подзадания для указанного задания. Код ответа HTTP – 200 (рисунок Е.6).
7	Получение подзаданий для текстового задания легкого уровня под номером 2.	GET	DbScript.php/?action=GET_EASY_TXT_TASK&user_id=236568364&level_type=text&level=2	Тест пройден. Получены подзадания для указанного задания. Код ответа HTTP – 200 (рисунок Е.7).
8	Получение подзаданий для задания с картинками легкого уровня под номером 1.	GET	DbScript.php/?action=GET_EASY_IMG_TASK&user_id=236568364&level_type=text&level=1	Тест пройден. Получены подзадания для указанного задания. Код ответа HTTP – 200 (рисунок Е.8).

Окончание таблицы 9

9	Получение подзаданий для задания с медиа легкого уровня под номером 1.	GET	DbScript.php/?action=GET_EASY_MEDIA_TASK&user_id=236568364&level_type=text&level=1	Тест пройден. Получены подзадания для указанного задания. Код ответа HTTP – 200 (рисунок Е.9).
10	Обновление последнего выбранного уровня сложности.	POST	DbScript.php/	Тест пройден. Новый пользователь добавлен. Код ответа HTTP – 200 (рисунок Е.10).

Все проведенные тесты были пройдены, что свидетельствует о корректности работы серверной части.

4.2 Тестирование клиентской части мобильного приложения

Тестирование программного обеспечения – важный процесс исследования программного продукта, целью которого является проверка соответствия между реальным поведением программы и ее ожидаемым поведением на конечном наборе тестов, выбранных определенным образом [25].

4.2.1 Функциональное тестирование

Рассмотрим данный вид тестирования на примере функции проверки правильного ответа в задании с медиа (аудио), используя метод эквивалентного разбиения.

Во время выполнения задания пользователь может прослушать фразу несколько раз, после чего должен ввести в специальное поле ответ и нажать кнопку «Далее». Введенный ответ должен содержать только латинские буквы и пробелы, регистр не важен. После нажатия кнопки «Далее» будет показан правильный ответ, а ответ пользователя будет подсвечен зеленым цветом, если он оказался правильным и красным в ином случае.

В таблице 10 приведены классы эквивалентности для поля ввода ответа, при проверяемой фразе «Привет мой друг».

Таблица 10 – Классы эквивалентности

Входные данные	Класс эквивалентности	Выходные данные	Полученный результат
-	Пустое значение	-	Рисунок Ж.1
Privet мой друг	Строка содержит символы кроме латинских букв	Введенный ответ выделен красным цветом	Введенный ответ выделен красным цветом (рисунок Ж.2)
Hello my friend	Строка содержит несколько пробелов между каждым словом	Введенный ответ выделен красным цветом	Введенный ответ выделен красным цветом (рисунок Ж.3)
Hello my friend	Строка содержит только символы русского алфавита и пробелы	Введенный ответ выделен зеленым цветом, ответ верный	Ответ верный (рисунок Ж.4)

Все тесты пройдены успешно, полученные результаты соответствуют ожидаемым.

4.3 Метрики кода

В листинге 1 приведена функция инициализации кнопок со словами для построения ответа в текстовых заданиях. Она слова для ответов берутся из списка слов, которые содержатся в правильном ответе и дополнительных. Эти слова перемешиваются случайным образом, после чего происходит создание кнопок.

Листинг 1 – функция инициализации кнопок для построения ответа

```
List<Widget> WidgetinitAnswerFlatButtons() {
    listOfButtons=[];
    if(shouldRefreshData) {
        answers = [
            ...mapOfTasks[taskIndex]['answer'].toString().split(' '),
            ...mapOfTasks[taskIndex]['additional'].toString().split(' ')
        ];
        answers.shuffle();
    }
}
```

```

        shouldRefreshData = false:
    );
    tempList=[];
    table = Table(children: []);
    for(int i=0;i<answers.length;i++){
        if((i!=0 && i%3 == 0)){
            table.children.add(TableRow(children: tempList));
            tempList=[];
        }
        tempList.add(
            child: FlatButton(
                onPressed: buttonPushed? (){} : (){
                    if(!listOfButtonsStates[i]){
                        setState(() {
                            finalAnswer+=answers[i]+' ';
                            listOfColors[i] = Colors.grey;
                        });
                    }else{
                        setState(() {
                            finalAnswer =
finalAnswer.replaceFirst(answers[i]+' ', '');
                            listOfColors[i] = Colors.lightBlue;
                        });
                    }
                    listOfButtonsStates[i]=!listOfButtonsStates[i];
                },
                shape: StadiumBorder(),
                color: listOfColors[i],
                child: Text(answers[i], style: TextStyle(color:
Colors.white),),),
            );
        }
        table.children.add(TableRow(children: tempList));
        listOfButtons.add(Container(child: table));
        return listOfButtons;
    }
}

```

На примере данной функции будут вычисляться метрики.

4.3.1 Метрика Холстеда

Для оценки листинга 1 вычислим метрику Холстеда. Для ее вычисления необходимо определить число уникальных операторов в коде (таблица 11) и операнд (таблица 12), а также их общее количество.

Таблица 11 – Уникальные операторы

№	Уникальные операторы
1	=
2	If
3	;
4	.
5	...
6	,
7	toString
8	Split
9	Add
10	For
11	Shuffle
12	++
13	:
14	<
15	+
16	&&
17	!=
18	%
19	?
20	Else
21	!
22	replaceFirst
23	Return

Таблица 12 – Уникальные операнды

№	Уникальные операнды
1	shouldRefreshData
2	Answers
3	finalAnswer
4	answers.length
5	Colors.lightBlue
6	0
7	3
8	tempList
9	buttonPushed
10	Colors.grey
11	StadiumBorder
12	Colors.white
13	I
14	i<answers.length
15	i!=0
16	i%3
17	!listOfButtonsStates[i]

Из таблиц 11 и 12 видно, чему равно число уникальных операторов ($n1$) и число уникальных операнд ($n2$):

$$n1 = 23,$$

$$n2 = 17.$$

В свою очередь общее число операторов $N1$ и общее число операнд $N2$ равны:

$$N1 = 102,$$

$$N2 = 84.$$

Далее вычислим словарь программы n и ее длину N :

$$n = n1 + n2 = 40,$$

$$N = N1 + N2 = 186.$$

Для вычисления объема необходимо подставить полученные значения в следующую формулу:

$$V = N * \log_2 n = 989.8786.$$

Комплексный показатель качества кода M рассчитывается по формуле:

$$M = \max(0, (171 - 5.2 * \ln(V) - 0.23 * CC - 16.2 * \ln(LoC))100/171),$$

где CC – структурная сложность кода;

LoC – количество строк кода.

$$CC = 1,$$

$$LoC = 43.$$

$$M = 43.258.$$

Комплексный показатель кода составляет 43. Данный показатель принимает значения от 0 до 100 и с его помощью определяют относительную сложность поддержки кода. Чем больше значение этого показателя, тем проще поддерживать код. Полученное значение для представленного варианта использования находится в допустимых границах и означает среднюю сложность поддержки кода. Таким образом, алгоритм кода оставления отзыва имеет приемлемый индекс сопровождаемости.

4.3.2 Цикломатическая метрика Мак-Кейба

Цикломатическая сложность программы основывается на анализе графа функции, представленного в приложении 3, и вычисляется по следующей формуле:

$$V(G) = e - n + 2p,$$

где e – количество дуг;

n – количество вершин;

p – число компонент связности.

Подсчитав их, получаем:

$$e = 14,$$

$$n = 13,$$

$$p = 2.$$

Далее подставляем эти величины в формулу выше и получаем:

$$V(G) = 14 - 13 + 2 * 2 = 5.$$

Полученное значение показывает, что рассматриваемая функция является понятной.

4.3.3 Метрика Чепина

Метрика Чепина позволяет определить информационную прочность кода с помощью анализа использования переменных. Она рассчитывается по следующей формуле:

$$Q = P + 2M + 3C + 0.5T,$$

где P – вводимые переменные;

M – переменные созданные внутри;

C – управляющие переменные;

T – неиспользуемые переменные.

Подсчитав их получаем:

$$P = 0,$$

$$M = 3$$

$$C = 2,$$

$$T = 0.$$

Подставим эти величины в формулу выше:

$$Q = 0 + 2 * 3 + 3 * 2 + 0.5 * 0 = 12.$$

Полученное значение показывает, что информационная прочность рассматриваемой функции достаточно велика.

4.4 Автоматизированное тестирование

Автоматизированное тестирование позволяет сэкономить программисту много времени, к тому же его результаты являются более точными, чем при ручном тестировании. Поэтому не одно полноценное тестирование программного продукта не может обойтись без проведения автоматизированных тестов.

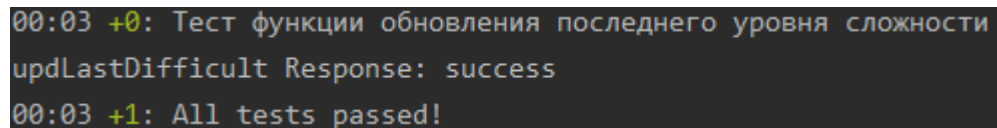
4.4.1 Unit-тесты

Для проверки правильности работы функции «Обновить уровень сложности», которая отправляет запрос на сервер, чтобы в базе данных обновилась информация о последнем уровне сложности, используемом пользователем, был написан unit-тест, представленный в листинге 2.

Листинг 2 – Unit-тест для функции «Обновить уровень сложности»

```
void main() {
    test('Тест функции обновления последнего уровня
сложности', () async{
        String initial = "success";
        String result = await
Database.updLastDifficult(39849830.toString(), "Средний");
        expect(initial, result);
        String expected = "error";
        result = await Database.updLastDifficult(39849830.toString(),
"Ошибка");
        expect(expected, result);    });}
```

Результат проведения теста представлен на рисунках 36, 37.



```
00:03 +0: Тест функции обновления последнего уровня сложности
updLastDifficult Response: success
00:03 +1: All tests passed!
```

Рисунок 36 – Результат проведения unit-теста в консоли

ID	last_difficult	first_name	last_name
39849830	Средний	Дмитрий	Угроватов

Рисунок 37 – Результат проведения unit-теста

В результате проведения unit-тестирования функции ошибки не были выявлены, из чего можно сделать вывод, что она работает корректно.

По результатам разностороннего тестирования различных функций программного обеспечения можно сделать вывод, что оно функционирует корректно. Был проведен анализ кода на основании метрик, который позволил сделать вывод об объеме программы и ее читаемости.

Заключение

В рамках выполнения выпускной квалификационной работы было разработано мобильное приложение для изучения иностранного языка.

На этапе выполнения анализа предметной области и требований были проанализированы аналоги разрабатываемого приложения, рассмотрены их преимущества и недостатки, разработаны функциональные и нефункциональные требования к создаваемому приложению, а также был проведен анализ целевой аудитории, был выбран английский язык для составления заданий и теории.

На этапе проектирования было осуществлено планирование разработки приложения и оценка бюджета, которое показало, что длительность реализации проекта составляет 57 дней и на его реализацию следует затратить 606 000 руб. Также на этом этапе были созданы концепции окон пользовательского интерфейса приложения, разработана структура классов, реализующая его.

Мобильное приложение было разработано с использованием языка dart в среде разработки Android Studio.

Тестирование клиентской части приложения проводилось методами функционального, структурного и unit-тестирования. Также были просчитаны метрики кода для оценки его качества.

В дальнейшем планируется опубликовать мобильное приложение в магазине приложений для android устройств Google Play и развернуть серверную часть на облачном сервисе. Также планируется расширить поддерживаемый набор языков для изучения.

Таким образом, в результате выполнения выпускной квалификационной работы, были выполнены все задачи технического задания.

Список использованных источников

1. 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: 17.05.2020)
2. 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: 17.05.2020)
3. Федеральный закон «Об образовании в Российской Федерации» № 273 – ФЗ от 29.12.2012 года с изменениями 2019 года. [Электронный ресурс]. – URL: <http://zakon-ob-obrazovanii.ru/> (дата обращения: 17.05.2020)
4. Приложение Duolingo // Pilot English School [Электронный ресурс]. URL: <https://pilot-school.ru/obzor-prilozheniya-duolingo-plyusy-minusy-stoit-li-polzovatsya/> (дата обращения: 18.05.2020)
5. Причины использовать Duolingo // Englishzoom. Изучение и преподавание английского [Электронный ресурс]. URL: <https://www.englishzoom.ru/poleznye-resursy/5-prichin-uchit-inostrannyj-yazyk-pri-pomoshhi-duolingo.html> (дата обращения: 18.05.2020)
6. Обзор Lingualeo // Это просто [Электронный ресурс]. URL: <https://prosto-eto.ru/obzor-lingualeo-servisa-po-izucheniyu-anglijskogo-yazyka/> (дата обращения: 18.05.2020)
7. Обзор школы английского Puzzle English // Английский язык [Электронный ресурс]. URL: <https://angliyskiyazik.ru/school/podrobnyj-obzor-shkoly-anglijskogo-puzzleenglish-otzyvy/> (дата обращения: 18.05.2020)
8. Средства для изучения английского // Cossa [Электронный ресурс]. URL: <https://www.cossa.ru/trends/200341/> (дата обращения: 18.05.2020)
9. Материалы с сайта ВЦИОМ [Электронный ресурс]. URL: <https://wciom.ru/index.php?id=236&uid=9924> (дата обращения: 18.05.2020)

10. Материалы с сайта Vox Media [Электронный ресурс]. URL: <https://www.theverge.com/2019/5/7/18528297/google-io-2019-android-devices-play-store-total-number-statistic-keynote> (дата обращения: 20.05.2020)
11. Материалы с сайта iXBT.com [Электронный ресурс]. URL: <https://www.ixbt.com/news/2019/01/30/skoro-milliard-apple-vpervye-rasskazala-skolko-iphone-ispolzuetsja-v-mire.html> (дата обращения: 20.05.2020)
12. Химонин Юрий. Сбор и анализ требований к программному продукту. // Московское отделение Project Management Institute [Электронный ресурс]. URL: http://pmi.ru/profes/Software_Requirements_Khimonin.pdf (дата обращения: 18.05.2020)
13. Вигерс, К. Разработка требований к программному обеспечению, 3е издание / К. Вигерс, Д. Битти – СПб.: БХВ-Петербург, 2017. –736с.
14. Справка по PhpStorm. // JetBrains [Электронный ресурс]. URL: <https://www.jetbrains.com/phpstorm/> (дата обращения: 18.05.2020)
15. Статья на Wikipedia о методологии RUP // Электронная энциклопедия [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Rational_Unified_Process (дата обращения: 20.05.2020)
16. Статья о методологии RUP // Информационный ресурс [Электронный ресурс]. URL: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=7> (дата обращения: 20.05.2020)
17. Flutter. Hot reload // Flutter [Электронный ресурс]. URL: <https://flutter.dev/docs/development/tools/hot-reload> (дата обращения: 20.05.2020)
18. Шкрыль, А.А. MS Project 2007. Современное управление проектами. / А.А. Шкрыль – СПб.: БХВ-Петербург, 2008. – 256 с.
19. Куперштейн В.И. Microsoft Project 2013 в управлении проектами. / В.И. Куперштейн – СПб.: БХВ-Петербург, 2014. – 432 с.

20. Джонсон Т. Microsoft Project 2016. Шаг за шагом. / Т. Джонсон, К. Четфилд – М.: КТК Галактика, 2018. – 594 с.
21. Бьяфоре Б. Все по плану! Успешное управление проектами с использованием Microsoft Project. / Б. Бьяфоре – М.: Русская редакция, 2006. — 305 с.
22. My SQL // Web creator [Электронный ресурс]. URL: <https://web-creator.ru/articles/mysql> (дата обращения: 22.05.2020)
23. Что такое PHP? // PHP.su [Электронный ресурс]. URL: <http://www.php.su/php/?php> (дата обращения: 22.05.2020)
24. PhpMyAdmin [Электронный ресурс]. URL: <https://www.phpmyadmin.net/> (дата обращения: 22.05.2020)
25. Тестирование ПО. [Электронный ресурс]. URL: https://www.edsd.ru/ru/uslugi/testirovanie_po - (дата обращения: 22.05.2020)

ПРИЛОЖЕНИЕ А
ГЛОССАРИЙ
(справочное)

RUP (Rational Unified Process) – методология разработки программного обеспечения, в основе которой лежит использование итеративной модели разработки.

UML (Unified Modeling Language) – язык графического описания, с помощью которого осуществляется объектное моделирование в области разработки программного обеспечения, проектирования систем и отображения организационных структур.

SDK (Software Development Kit) – набор средств разработки, позволяющий программистам разрабатывать приложения для определенной платформы.

Android – операционная система для мобильных устройств, разработанная компанией Google.

SQL (Structured Query Language) – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

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

ПРИЛОЖЕНИЕ Б

Сетевой график и критический путь
(рекомендуемое)

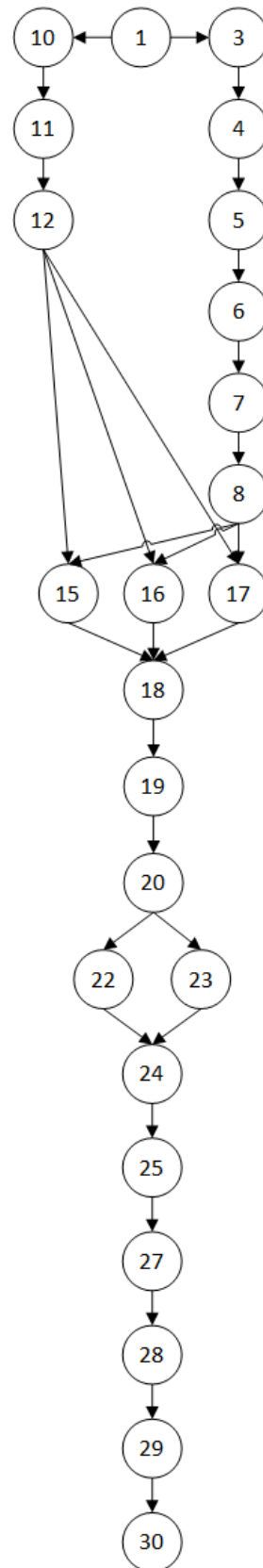


Рисунок Б.1 – Сетевой график проекта

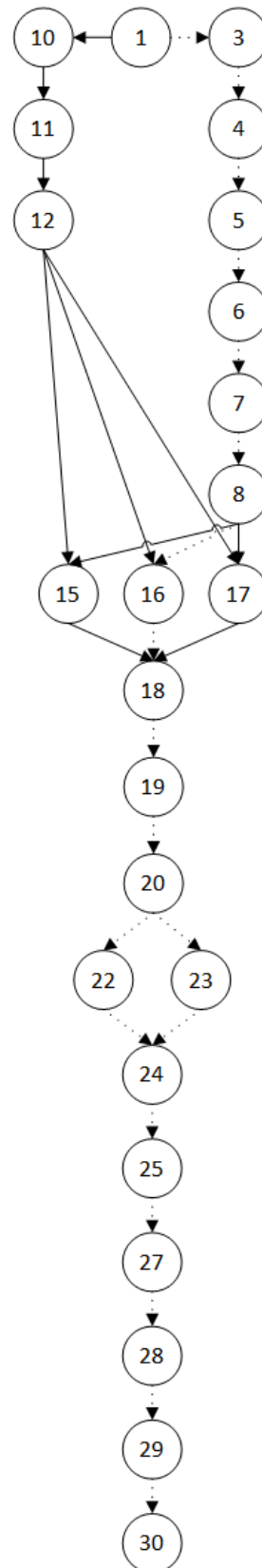


Рисунок Б.2 – Критический путь

ПРИЛОЖЕНИЕ В

Исходный код серверной части

(рекомендуемое)

Листинг В.1 – Файл main.php

```
<?php
    $servername = "127.0.0.1:3306";
    $username = "user";
    $password = "user";
    $dbname = "lang_guide";
    $table = "Employees"; // lets create a table named Employees.

    $tb_users = "users";
    $tb_easy_level = "easy_level";
    $tb_medium_level = "medium_level";
    $tb_hard_level = "hard_level";

    // we will get actions from the app to do operations in the
database...
    $action = $_POST["action"];

    // Create Connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check Connection
    if($conn->connect_error){
        die("Connection Failed: " . $conn->connect_error);
        return;
    }

    // If connection is OK...

    // Добавить информацию о пользователе в базу данных
    if("CHECK_ADD_USR" == $action){
        $user_id = $_POST["user_id"];
        $first_name = $_POST["first_name"];
        $last_name = $_POST["last_name"];
        $last_difficult = $_POST["last_difficult"];

        $sql = "SELECT last_difficult from $tb_users where VK_ID =
'$user_id'";
        $result = $conn->query($sql);
        if($result->num_rows === 0){
```

```

        $sql = "INSERT INTO $tb_users (VK_ID, first_name,
last_name, last_difficult) VALUES ('$user_id', '$first_name',
'$last_name', '$last_difficult')";
        $result = $conn->query($sql);
        echo "$last_difficult";
    }
    else{
        $row = $result->fetch_assoc();

        // Send back the complete records as a json
        echo $row["last_difficult"];
    }
    $conn->close();
    return;
}

// Добавить информацию о выполненном уровне
if("ADD_COMP_LVL" == $action){
    $user_id = $_POST["user_id"];
    $level_difficult = $_POST["level_difficult"];
    $level_type = $_POST["level_type"];
    $level_number = $_POST["level_number"];
    $points = $_POST["points"];

    $sql = "INSERT INTO $level_difficult (VK_ID, level_type,
level_number, points) VALUES ('$user_id', '$level_type',
'$level_number', $points)";
    $result = $conn->query($sql);
    echo "added succesfull";
    $conn->close();
    return;
}

if("UPD_LAST_DIFF" == $action){
    // App will be posting these values to this server
    $user_id = $_POST["user_id"];
    $last_difficult = $_POST["last_difficult"];

    $sql = "UPDATE $tb_users SET last_difficult =
'$last_difficult' WHERE VK_ID = '$user_id'";
    if($conn->query($sql) === TRUE){

```

```

        echo "success";
    }else{
        echo "error";
    }
    $conn->close();
    return;
}

// Добавить или обновить запись о выполненном уровне
if("ADD_UPD_COMP_LVL" == $action){
    // App will be posting these values to this server
    $user_id = $_POST["user_id"];
    $level_difficult = $_POST["level_difficult"];
    $level_type = $_POST["level_type"];
    $level_number = $_POST["level_number"];
    $points = $_POST["points"];

    $sql = "SELECT * from $level_difficult where VK_ID =
'$user_id' AND level_type = '$level_type' AND level_number =
'$level_number'";
    $result = $conn->query($sql);
    if($result->num_rows === 0){
        $sql = "INSERT INTO $level_difficult (VK_ID,
level_type, level_number, points) VALUES ('$user_id', '$level_type',
'$level_number', '$points')";
        $result = $conn->query($sql);
        echo "added completed level";
    }
    else{
        $sql = "UPDATE $level_difficult SET points = '$points'
WHERE VK_ID = '$user_id' and level_type = '$level_type' and
level_number = '$level_number'";
        if($conn->query($sql) === TRUE){
            echo "success";
        }else{
            echo "error";
        }
        echo " then updated completed level";
    }
}

```

```

$conn->close();
return;
}

// получить список выполненных уровней
if('GET_COMP_LVLIS' == $action){
    $db_data = array();
    $user_id = $_POST["user_id"];
    $level_difficult = $_POST["level_difficult"];
    $level_type = $_POST["level_type"];

    $sql = "SELECT level_number, points FROM $level_difficult
WHERE VK_ID = '$user_id' and level_type='$level_type'";
    $result = $conn->query($sql);
    if($result->num_rows > 0){
        while($row = $result->fetch_assoc()){
            $db_data[] = $row;
        }
        // Send back the complete records as a json
        echo json_encode($db_data);
    }else{
        echo "zero";
    }
    $conn->close();
    return;
}

//получить вопросы для текстового задания легкого уровня
if('GET_EASY_TXT_TASK' == $action){
    $db_data = array();
    $level = $_POST["level"];
    switch($level){
        case "1":
            $db_data =array(
                0 => array(
                    "Переведите предложение" => "У меня
есть яблоко.",
                    "answer" => "I have an apple",

```

```

        "additional" => "has a she"
    ),
    1 =>array(
        "Переведите предложение" => "У нее
есть молоко.",
        "answer" => "She has milk",
        "additional" => "have a he"
    ),
    2 =>array(
        "Переведите предложение" => "Я читаю
книгу.",
        "answer" => "I read a book",
        "additional" => "have an hee"
    ),
    3 =>array(
        "Переведите предложение" => "Мой папа
смотрит телевизор.",
        "answer" => "My father watch TV",
        "additional" => "mother read"
    )
);
break;
case "2":
    $db_data =array(
        0 => array(
            "Переведите предложение" => "I read a
book.",
            "answer" => "Я читаю книгу",
            "additional" => "смотрю телевизор"
        ),
        1 =>array(
            "Переведите предложение" => "I have an
apple.",
            "answer" => "У меня есть яблоко",
            "additional" => "него молоко"
        ),
        2 =>array(
            "Переведите предложение" => "She has
milk.",

```

```

        "answer" => "У нее есть молоко",
        "additional" => "него яблоко"
    ),
    3 => array(
        "Переведите предложение" => "My father
watch TV.",
        "answer" => "Мой папа смотрит
телевизор",
        "additional" => "мама читает Моя
книгу"
    )
);
break;
}
echo json_encode($db_data);
return;
}

if('GET_EASY_IMG_TASK' == $action){
    $db_data = array();
    $level = $_POST["level"];
    switch($level){
        case "1":
            $db_data = array(
                0 => array(
                    "task" => "Выберите картинку, на
которой изображено слово ",
                    "answerWord" => "Apple",
                    "answerImg" => "/images/apple.png",
                    "additionalImg" => "/images/melon.png
/images/milk.png /images/dog.png"
                ),
                1 => array(
                    "task" => "Выберите картинку, на
которой изображено слово ",
                    "answerWord" => "Milk",
                    "answerImg" => "/images/milk.png",
                    "additionalImg" => "/images/melon.png
/images/girl.png /images/orange.png"
                )
            )
        }
    }
}

```

```

        ),
        2 =>array(
            "task" => "Выберите картинку, на
которой изображено слово ",
            "answerWord" => "Melon",
            "answerImg" =>"/images/melon.png",
            "additionalImg" => "/images/orange.png
/images/watermelon.png /images/dog.png"
        ),
        3 =>array(
            "task" => "Выберите картинку, на
которой изображено слово ",
            "answerWord" => "Orange",
            "answerImg" =>"/images/orange.png",
            "additionalImg" =>
"/images/watermelon.png /images/bird.png /images/dog.png"
        )
    );
    break;
    case "2":
        $db_data =array(
            0 => array(
                "task" => "Выберите картинку, на
которой изображено слово ",
                "answerWord" => "Boy",
                "answerImg" =>"/images/boy.png",
                "additionalImg" => "/images/bird.png
/images/dog.png /images/girl.png"
            ),
            1 =>array(
                "task" => "Выберите картинку, на
которой изображено слово ",
                "answerWord" => "Girl",
                "answerImg" =>"/images/girl.png",
                "additionalImg" => "/images/dog.png
/images/cat.png /images/police.png"
            ),
            2 =>array(

```



```

        "additional" => "Where are our age"),
2 =>array(
    "task" => "Что сказал диктор?",
    "type" => "audio",
    "answer" => "Boy likes bananas",
    "additional" => "like Boys ban"
),
3 =>array(
    "task" => "Что сказал диктор?",
    "type" => "audio",
    "answer" => "I love my parents",
    "additional" => "live me apple"
)
);
break;
case "2":
    $db_data =array(
        0 => array(
            "task" => "Что сказал диктор?",
            "type" => "audio",
            "answer" => "This is my dog",
            "additional" => "Where me doctor"
        ),
        1 =>array(
            "task" => "Что сказал диктор?",
            "type" => "audio",
            "answer" => "My dog likes meat",
            "additional" => "like lie meal"
        ),
        2 =>array(
            "task" => "Что сказал диктор?",
            "type" => "audio",
            "answer" => "This is my cat",
            "additional" => "Where pad"
        ),
        3 =>array(
            "task" => "Что сказал диктор?",
            "type" => "audio",
            "answer" => "My cat likes chocolate",

```

```
                "additional" => "like Me crocodile"
            )
        );
        break;
    }
    echo json_encode($db_data);
    return;
}
?>
```

ПРИЛОЖЕНИЕ Г
Исходный код приложения
(рекомендуемое)

Листинг Г.1 – Файл main.dart

```
import 'package:flutter/material.dart';
import 'package:lang_guide/pages/auth.dart';
import 'package:lang_guide/pages/imageLevel.dart';
import 'package:lang_guide/pages/loading.dart';
import 'package:lang_guide/pages/main_menu.dart';
import 'package:lang_guide/pages/select_difficult.dart';
import 'package:lang_guide/pages/select_level.dart';
import 'package:lang_guide/pages/textLevel.dart';
import 'package:lang_guide/pages/resultLevel.dart';
import 'package:lang_guide/pages/mediaLevel.dart';
import 'package:lang_guide/pages/theoryPage.dart';

void main() {
  runApp(MaterialApp(
    initialRoute: '/authorization',
    routes: {
      '/loading': (context) => Loading(),
      '/authorization': (context) => Authorization(),
      '/mainMenu': (context) => MainMenu(),
      '/selectDifficult': (context) => SelectDifficult(),
      '/selectLevel' : (context) => SelectLevel(),
      '/textLevel' : (context) => textLevel(),
      '/imageLevel' : (context) => imageLevel(),
      '/resultLevel' : (context) => ResultLevel(),
      '/mediaLevel' : (context) => mediaLevel(),
      '/theoryPage' : (context) => theoryPage(),
      //'options': (context) => Options(),
      //'TrainingType': (context) => TrainingType(),
      //'TrainingLevel' : (context) => TrainingLevel(),
    },
  ));
}
```

Листинг Г.2 – Файл database.dart

```
import 'dart:convert';
import 'package:http/http.dart' as http;
```

```

class Database {
    static String db_name = "lang_guide";
    static String tb_users = "users";
    static String tb_easy_level = "easy_level";
    static String tb_medium_level = "medium_level";
    static String tb_hard_level = "hard_level";

    static const ROOT = 'http://192.168.0.106/DbScript.php';

    static const _GET_COMP_LVLIS_ACTION = 'GET_COMP_LVLIS';
    static const _CHECK_OR_ADD_USR_ACTION = 'CHECK_ADD_USR';
    static const _UPD_LAST_DIFF_ACTION = 'UPD_LAST_DIFF';
    static const _UPD_COMP_LVL_ACTION = 'ADD_UPD_COMP_LVL';
    static const _GET_EASY_TXT_TASK = 'GET_EASY_TXT_TASK';
    static const _GET_EASY_IMG_TASK = 'GET_EASY_IMG_TASK';
    static const _GET_EASY_MEDIA_TASK = 'GET_EASY_MEDIA_TASK';

    static String translateLevelDifficult(String level_difficult){
        switch(level_difficult){
            case 'Начинающий':
                return tb_easy_level;
            case 'Средний':
                return tb_medium_level;
            case 'Продвинутый':
                return tb_hard_level;
            case 'Начинающий':
                return 'Начинающий';
            case 'Средний':
                return 'Средний';
            case 'Продвинутый':
                return 'Продвинутый';
            default:
                return "error";
        }
    }

    static Future<String> checkOrAddUser(String user_id, String
first_name, String last_name) async {

```

```

try{
    var map = Map<String,dynamic>();
    map['action'] = _CHECK_OR_ADD_USR_ACTION;
    map['user_id'] = user_id;
    map['first_name'] = first_name;
    map['last_name'] = last_name;
    map['last_difficult'] = 'Начинающий';

    final response = await http.post(ROOT, body: map);
    print('addUser Response: ${response.body}');
    if (200 == response.statusCode) {
        return response.body;
    } else {
        return "error";
    }
} catch(e) {
    return "error";
}
}

static Future<String> updLastDifficult(String user_id,String
last_difficult) async {
    try{
        var map = Map<String,dynamic>();
        map['action'] = _UPD_LAST_DIFF_ACTION;
        map['user_id'] = user_id;
        map['last_difficult'] = last_difficult;

        if(translateLevelDifficult(last_difficult)=="error")
            return "error";

        final response = await http.post(ROOT, body: map);
        print('updLastDifficult Response: ${response.body}');
        if (200 == response.statusCode) {
            return response.body;
        } else {
            return "error";
        }
    } catch(e) {

```

```

        return "exc error";
    }
}

```

```

static Future<String> addUpdCompletedLevel(String user_id,String
level_difficult, String level_type, String level_number, String
points) async {

```

```

    try{
        var map = Map<String,dynamic>();
        map['action'] = _UPD_COMP_LVL_ACTION;
        map['user_id'] = user_id;
        map['level_difficult'] =
translateLevelDifficult(level_difficult);
        map['level_type'] = level_type;
        map['level_number'] = level_number;
        map['points'] = points;
        if(translateLevelDifficult(level_difficult)=="error")
            return "error";
        final response = await http.post(ROOT, body: map);
        print('updCompletedLevel Response: ${response.body}');
        if (200 == response.statusCode) {
            return response.body;
        } else {
            return "error";
        }
    }catch(e){
        print (e);
        return "exc error";
    }
}

```

```

static Future<List<Map<String, dynamic>>> getCompletedLevels(String
user_id, String level_difficult, String level_type) async{

```

```

    List<Map<String, dynamic>>resultList=[];
    try{
        var map = Map<String,dynamic>();
        map['action'] = _GET_COMP_LVL_ACTION;
        map['user_id'] = user_id;

```

```

        map['level_difficult'] =
translateLevelDifficult(level_difficult);
        map['level_type'] = level_type;
        if(translateLevelDifficult(level_difficult)=="error")
            return resultList;

        final response = await http.post(ROOT, body: map);
        print('getCompletedLevels Response: ${response.body}');
        if(response.body == 'zero')
            return resultList;
        final parsed = json.decode(response.body).cast<Map<String,
dynamic>>());
        resultList = parsed.toList();

        if (200 == response.statusCode) {
            return resultList;
        } else {
            return resultList;
        }
    }catch(e){
        print (e);
        return resultList;
    }
}

static Future<List<Map<String, dynamic>>> getEasyTextTasks(int
level) async{
    try{
        var map = Map<String,dynamic>();
        map['action'] = _GET_EASY_TXT_TASK;
        map['level'] = level.toString();

        final response = await http.post(ROOT, body: map);
        //print('getEasyTextTasks ${level} Response: ${response.body}');
        final parsed = json.decode(response.body).cast<Map<String,
dynamic>>());
        List<Map<String, dynamic>> resultList = parsed.toList();

        if (200 == response.statusCode) {

```

```

        return resultList;
    } else {
        return [];
    }
} catch(e) {
    print(e);
    return [];
}
}

```

```

static Future<List<Map<String, dynamic>>> getEasyImageTasks(int
level) async{
    try{
        var map = Map<String,dynamic>();
        map['action'] = _GET_EASY_IMG_TASK;
        map['level'] = level.toString();

        final response = await http.post(ROOT, body: map);
        final parsed = json.decode(response.body).cast<Map<String,
dynamic>>>();
        List<Map<String, dynamic>> resultList = parsed.toList();

        if (200 == response.statusCode) {
            return resultList;
        } else {
            return [];
        }
    } catch(e) {
        print(e);
        return [];
    }
}

```

```

static Future<List<Map<String, dynamic>>> getEasyMediaTasks(int
level) async{
    try{
        var map = Map<String,dynamic>();
        map['action'] = _GET_EASY_MEDIA_TASK;
        map['level'] = level.toString();

```

```

        final response = await http.post(ROOT, body: map);
        final parsed = json.decode(response.body).cast<Map<String,
dynamic>>();
        List<Map<String, dynamic>> resultList = parsed.toList();

        if (200 == response.statusCode) {
            return resultList;
        } else {
            return [];
        }
    } catch(e) {
        print(e);
        return [];
    }
}
}

```

Листинг Г.3 – Файл CustomSelectButton.dart

```

import 'package:flutter/material.dart';
class CustomSelecetLevelButton extends StatelessWidget{
    GestureTapCallback onTap;
    IconData iconData;
    int donutCount;
    CustomSelecetLevelButton({this.iconData, this.onTap,
this.donutCount});
    List<Widget> getDonuts(){
        List<Widget> listOfDonuts=[];
        for(int i=0;i<4;i++){
            double opacity = i<donutCount?1:0.1;
            listOfDonuts.add(
                Opacity(
                    opacity: opacity,
                    child: Image.asset('assets/donut.png', width: 25, height:
25,))
            );
        }
        return listOfDonuts;
    }
}

```

```

@override
Widget build(BuildContext context){
  return Column(
    children: <Widget>[
      ClipOval(
        child: Material(
          color: Colors.lightBlue, // button color
          child: InkWell(
            splashColor: Colors.purple, // inkwell color
            child: SizedBox(width: 90, height: 90, child:
Icon(iconData,size: 90,)),
            onTap: onTap,
          ),
        ),
      ),
      Row(children: getDonuts(),)
    ],
  );
}

```

Листинг Г.4 – Файл auth.dart

```

import 'package:flutter_vk_sdk/flutter_vk_sdk.dart';
import 'package:flutter_vk_sdk/vk_scope.dart';
import
'package:material_design_icons_flutter/material_design_icons_flutter.d
art';
import 'dart:convert';
import 'package:http/http.dart';
import 'package:flutter/material.dart';
import 'package:lang_guide/services/database.dart';

const String appId = '7442177';
const String apiVersion = '5.103';

class Authorization extends StatefulWidget {
  @override
  _AuthorizationState createState() => _AuthorizationState();
}

```

```

}

class _AuthorizationState extends State<Authorization> {
  int id;
  String token;
  String firstName='Unkown';
  String lastName='Unkown';
  String level_difficult = 'Начинающий';
  String avatar =
'https://1001freedownloads.s3.amazonaws.com/vector/thumb/75167/1366695
174.png';

  initVKSdk() {
    return FlutterVKSdk.init(appId: appId, apiVersion: apiVersion);
  }

  @override
  void initState() {
    super.initState();
    //SystemChrome.setEnabledSystemUIOverlays([]);
  }

  Future<Map> getInfo() async{
    Response response = await
get('https://api.vk.com/method/users.get?user_ids=$id&fields=crop_phot
o&access_token=$token&v=$apiVersion');
    return jsonDecode(response.body)['response'][0];
  }

  void vkLogin() async {
    await initVKSdk();
    await FlutterVKSdk.login(
      scope: '${VKScope.email}',
      onSuccess: (res) async {
        token = res.token;
        id = res.userId;
        Map result = await getInfo();
        String check_result = await
Database.checkOrAddUser(id.toString(), firstName, lastName);
        if(check_result != 'error'){

```

```

        level_difficult = check_result;
    }

    setState(() {
        firstName = result['first_name'];
        lastName = result['last_name'];
        avatar = result['crop_photo']['photo']['sizes'][4]['url'];
        Navigator.pushNamed(context, '/mainMenu',arguments: {
            'firstName': firstName,
            'lastName': lastName,
            'avatar': avatar,
            'user_id' : id.toString(),
            'currentDifficult' : level_difficult
        });
    });
},
onError: (error) {
    print('LOGIN ERROR: $error');
    setState(() {
        firstName = 'error';
    });
},
);
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.lightBlue,
        body: Center(
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.center,
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                    Text('Lang guide', style: TextStyle(fontSize: 60.0,color:
Colors.white)),
                    SizedBox(height: 20,),
                    FlatButton.icon(
                        icon: Icon(MdiIcons.vk,size: 40,color: Colors.white),

```

```

        onPressed: vkLogin,
        color: Colors.blue[800],
        padding: EdgeInsets.all(10.0),
        label: Text("Авторизироваться", style:
TextStyle(fontSize: 20.0,color: Colors.white))
      ),
    ],
  ),
),
);
}
}

```

Листинг Г.5 – Файл imageLevel.dart

```

import 'dart:ui';
import 'package:flutter/material.dart';
import 'package:transparent_image/transparent_image.dart';

class imageLevel extends StatefulWidget {
  @override
  _imageLevelState createState() => _imageLevelState();
}

class _imageLevelState extends State<imageLevel> {

  int resultPoints = 0;
  int taskIndex = 0;
  List<Map<String,dynamic>> mapOfTasks=[];
  List<Color> listOfColors=[];
  bool shouldRefreshData = true;
  var finalAnswerColor = Colors.black;
  ValueKey<String> rightAnswer;
  List<String> answers;
  Future<Map<String,List<Image>>> futureMapOfImages;
  Map<String,List<Image>> mapOfImages;
  bool isButtonPushed =false;
  List<bool> listIsButtonPushed;
  bool isRightPushed = false;
  List<double> whenRightPushed=[100,100,100,100];

```

```

String level_number;
String user_id;
static const String Level_Type = 'image';
String level_difficult;

Widget initImageInkWell(int index, bool isRightAnswer){
  Color backgroundColor = Colors.transparent;
  if(isButtonPushed){
    backgroundColor=isRightAnswer?Colors.green:Colors.red;
  }

  return Expanded(
    flex: 5,
    child: Padding(
      padding: const EdgeInsets.all(4.0),
      child: InkWell(
        splashColor: Colors.purple, // inkwell color
        child: Container(color: backgroundColor,width:
whenRightPushed[index], height: whenRightPushed[index], child:
mapOfImages['$taskIndex'][index],),
        onTap: (){
          setState(() {
            if(!isButtonPushed){
              isButtonPushed = true;
              isRightPushed = true;
              whenRightPushed[index]=140;
              if(mapOfImages['$taskIndex'][index].key
==rightAnswer){
                resultPoints++;
              }
              return;
            }
          });
        },
      ),
    ),
  );
}

```

```

Widget getImage(String path) {
    return FadeInImage.memoryNetwork(
        fadeInCurve: Curves.bounceIn,
        placeholder: kTransparentImage,
        image: 'http://192.168.0.105$path',
        height: MediaQuery.of(context).size.height*0.35,
        width: MediaQuery.of(context).size.height*0.3,);
}

Future<Map<String,List<Image>>> initImages()async{
    Map<String,List<Image>> result= Map<String,List<Image>>();
    for(int i=0;i<4;i++){
        answers = [
            ...mapOfTasks[i]['answerImg'].toString().split(' '),
            ...mapOfTasks[i]['additionalImg'].toString().split(' ')
        ];
        List<Image> listOfImage=[];
        for(int j=0;j<4;j++){
            listOfImage.add(
                Image(image:
NetworkImage('http://192.168.0.105${answers[j]}'),
                    key: ValueKey<String>(answers[j]),
                    height: MediaQuery.of(context).size.height*0.35,
                    width: MediaQuery.of(context).size.height*0.3,));
            listOfImage.shuffle();
            result['$i']=listOfImage;
        }
        return result;
    }
}

void initAnswers(){
    if(shouldRefreshData) {
        answers = [
            ...mapOfTasks[taskIndex]['answerImg'].toString().split(' '),
            ...mapOfTasks[taskIndex]['additionalImg'].toString().split(' ')
        ];
        whenRightPushed=[100,100,100,100];
    }
}

```

```

        isRightPushed = false;
        rightAnswer =ValueKey(answers[0]);
        answers.shuffle();
        shouldRefreshData = false;
        isButtonPushed =false;
        listIsButtonPushed = [false,false,false,false];
    }
}

Widget initImageTable(){

    List<Row> listOfRows=[];
    bool first = true;
    int j=0;
    for(int i=0;i<4;i++){
        bool isRightAnswer = mapOfImages['$taskIndex'][i].key
==rightAnswer;

        if(first){
            listOfRows.add(Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: <Widget>[
                    initImageInkWell(i,isRightAnswer)
                ]
            ));
        }else{
            listOfRows[j].children.add(
                initImageInkWell(i,isRightAnswer)
            );
            j++;
        }
        first=!first;
    }
    return Column(mainAxisAlignment:
MainAxisAlignment.center,children: listOfRows, );
}

Widget initImageTask(){
    return SafeArea(

```

```
child: Column(  
    crossAxisAlignment: CrossAxisAlignment.center,  
    children: <Widget>[  
        SizedBox(height: 10,),  
        LinearProgressIndicator(  
            value: taskIndex.toDouble()*0.25,  
            valueColor: AlwaysStoppedAnimation<Color>(Colors.blue),  
            backgroundColor: Colors.grey,  
        ),  
        SizedBox(height: 50,),  
        Card(  
            margin: EdgeInsets.symmetric(vertical: 15,horizontal:  
10),  
  
            color: Colors.lightBlue,  
            child: ListTile(  
                title:RichText(  
                    textAlign: TextAlign.center,  
                    text: TextSpan(  
                        text: mapOfTasks[taskIndex]['task'],  
                        style: TextStyle(  
                            fontWeight: FontWeight.bold,  
                            fontSize: 22,  
                            color: Colors.black),  
                        children: <TextSpan>[  
                            TextSpan(  
                                text: mapOfTasks[taskIndex]['answerWord'],  
                                style: TextStyle(  
                                    fontWeight: FontWeight.bold,  
                                    fontSize: 22,  
                                    color: Colors.pink)  
                                )  
                            ],  
                        ),  
                    ),  
                ),  
            ),  
        ),  
        SizedBox(height: 30,),  
        initImageTable(),
```

```

Flexible(child: FractionallySizedBox(heightFactor:
0.7,)),),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: <Widget>[
    FlatButton(
      onPressed: ()async{
        if(taskIndex==3){
          await Navigator.pushNamed(context,
'/resultLevel', arguments: {
            'resultPoints' : resultPoints,
            'user_id' : user_id,
            'level_difficult' : level_difficult,
            'level_type' : Level_Type,
            'level_number' : level_number});
          Navigator.pop(context);
          return;
        }
        setState(() {
          if(!isButtonPushed) {
            isButtonPushed = true;
            return;
          }else {
            isButtonPushed = false;
            shouldRefreshData = true;
          }

          taskIndex++;
        });
      },
      shape: StadiumBorder(),
      color: Colors.lightBlue,
      child: Text("Далее"),
    ),
  ],
),
],
),
);

```

```

    }

    @override
    Widget build(BuildContext context) {
      if (mapOfTasks.isEmpty) {
        final Map<String, dynamic> data =
ModalRoute.of(context).settings.arguments;
        user_id = data['user_id'];
        level_difficult = data['difficult'];
        level_number = data['level_number'];
        for (int i = 0; i < 4; i++) {
          mapOfTasks.add(data['$i']);
        }
      }
      if (shouldRefreshData)
        futureMapOfImages = initImages();
      initAnswers();

      return Scaffold(
        body: FutureBuilder<Map<String, dynamic>>(
          future: futureMapOfImages,
          builder: (BuildContext context,
AsyncSnapshot<Map<String, dynamic>> snapshot) {
            if (snapshot.hasData) {
              mapOfImages = snapshot.data;
              return initImageTask();
            } else {
              return Center(child: CircularProgressIndicator());
            }
          },
        ),
      );
    }
  }
}

```

Листинг Г.6 – Файл main_menu.dart

```

import 'package:flutter/material.dart';
import 'package:flutter/rendering.dart';
import 'package:flutter_vk_sdk/flutter_vk_sdk.dart';

```

```

import
'package:material_design_icons_flutter/material_design_icons_flutter.d
art';
import 'package:lang_guide/services/database.dart';

class MainMenu extends StatefulWidget {
  @override
  _MainMenuState createState() => _MainMenuState();
}

String currentDifficult='Начинающий';

class _MainMenuState extends State<MainMenu> with
WidgetsBindingObserver {

  int _selectedWidgetIndex = 0;
  Map data={};
  static String firstName = 'Unkown';
  static String lastName = 'Unkown';
  static String id = '';
  static String avatar = '';
  static Widget widget1 = Column();
  static Widget widget3 = Column();

  static const Color SpecialTextColor = Colors.black;
  static const Color UsualTextColor = Colors.lightBlue;

  List<Widget> _widgets = <Widget>[
    widget1,
    Center(
      child: Text('Грамматика'),
    ),
    widget3,
  ];

  void updateWidgets(){
    _widgets[0] = Center(
      child: Card(
        color: UsualTextColor,

```

```

child: Column(
  mainAxisAlignment: MainAxisAlignment.center,
  mainAxisAlignment: MainAxisAlignment.min,
  children: <Widget>[
    Card(
      margin: EdgeInsets.symmetric(vertical: 15,horizontal:
10),

      color: Colors.white,
      child: ListTile(
        title: Text('Текст',style: TextStyle(fontSize:
20,color: SpecialTextColor)),
        leading: Icon(Icons.textsms, color: UsualTextColor),
        onTap: () async{
          await Navigator.pushNamed(context,
'/selectLevel',arguments: {
            'levelType' : 'text',
            'levelDifficult' :currentDifficult,
            'user_id' : id
          });
        },
      ),
    ),
    Card(
      margin: EdgeInsets.symmetric(vertical: 0,horizontal:
10),

      color: Colors.white,
      child: ListTile(
        title: Text('Картинки',style: TextStyle(fontSize:
20,color: SpecialTextColor)),
        leading: Icon(Icons.image, color: UsualTextColor),
        onTap: () async{
          await Navigator.pushNamed(context,
'/selectLevel',arguments: {
            'levelType' : 'image',
            'levelDifficult' :currentDifficult,
            'user_id' : id
          });
        },
      ),
    ),
  ],
)

```

```

    ),
    Card(
      margin: EdgeInsets.symmetric(vertical: 15,horizontal:
10),

      color: Colors.white,
      child: ListTile(
        title: Text('Медиа',style: TextStyle(fontSize:
20,color: SpecialTextColor)),
        leading: Icon(Icons.headset, color: UsualTextColor),
        onTap: () async{
          await Navigator.pushNamed(context,
'/selectLevel',arguments: {
            'levelType' : 'media',
            'levelDifficult' :currentDifficult,
            'user_id' : id
          });
        }
      ),
    )
  ],
),
),
);
_widgets[1] = Center(
  child: Card(
    color: UsualTextColor,
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      mainAxisSize: MainAxisSize.min,
      children: <Widget>[
        Card(
          margin: EdgeInsets.symmetric(vertical: 15,horizontal:
10),

          color: Colors.white,
          child: ListTile(
            title: Text('Время',style: TextStyle(fontSize:
20,color: SpecialTextColor)),
            leading: Icon(Icons.access_time, color:
UsualTextColor),

```

```

        onTap: () async{
          await Navigator.pushNamed(context,
'/theoryPage',arguments: {
          'pageType' : 'times'
        });
      },
    ),
  ),
  Card(
    margin: EdgeInsets.symmetric(vertical: 0,horizontal:
10),

    color: Colors.white,
    child: ListTile(
      title: Text('Местоимения',style: TextStyle(fontSize:
20,color: SpecialTextColor)),
      leading: Icon(MdiIcons.human, color: UsualTextColor),
      onTap: () async{
        await Navigator.pushNamed(context,
'/theoryPage',arguments: {
          'pageType' : 'pronouns'
        });
      },
    ),
  ),
  Card(
    margin: EdgeInsets.symmetric(vertical: 15,horizontal:
10),

    color: Colors.white,
    child: ListTile(
      title: Text('Предлоги',style: TextStyle(fontSize:
20,color: SpecialTextColor)),
      leading: Icon(Icons.arrow_upward, color:
UsualTextColor),
      onTap: () async{
        await Navigator.pushNamed(context,
'/theoryPage',arguments: {
          'pageType' : 'prepositions'
        });
      }
    )
  )
}

```

```

        ),
      ),
      Card(
        margin: EdgeInsets.fromLTRB(10, 0, 10, 15),
        color: Colors.white,
        child: ListTile(
          title: Text('Неправильные глаголы', style:
TextStyle(fontSize: 20, color: SpecialTextColor)),
          leading: Icon(Icons.work, color: UsualTextColor),
          onTap: () async{
            await Navigator.pushNamed(context,
'/theoryPage', arguments: {
              'pageType' : 'verbs'
            });
          }
        ),
      ),
    ],
  ),
),
);

_widgets[2] =Column(
  mainAxisAlignment: MainAxisAlignment.max,
  children: <Widget>[
    Card(
      shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(10.0)),
      margin: EdgeInsets.symmetric(vertical: 5, horizontal: 10),
      color: UsualTextColor,
      child: ListTile(
        title: Text('$firstName $lastName',
          style: TextStyle(fontSize: 25, color:
Colors.white)),
        leading: CircleAvatar(
          radius: 20,
          backgroundImage: NetworkImage(avatar),
        ),
      ),
    ),
  ],
),

```

```

Flexible(
  child: FractionallySizedBox(
    heightFactor: 0.45,
  ),
),
Card(
  shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(10.0)),
  margin: EdgeInsets.symmetric(vertical: 0,horizontal: 10),
  color: UsualTextColor,
  child: Column(
    children: <Widget>[
      Card(
        margin: EdgeInsets.symmetric(vertical:
15,horizontal: 10),
        color: Colors.white,
        child: ListTile(
          title: RichText(
            text: TextSpan(
              //style: DefaultTextStyle.of(context).style,
              children: <TextSpan>[
                TextSpan(text:"Уровень:\n",style:
TextStyle(fontSize: 25.0,color: UsualTextColor) ),
                TextSpan(text:"$currentDifficult",
style: TextStyle(fontWeight: FontWeight.bold,fontSize: 25.0,color:
SpecialTextColor)),//orange[400]
              ])),
          leading: Icon(Icons.arrow_forward,size: 40,color:
UsualTextColor,)),
          onTap: () async{
            dynamic result = await
Navigator.pushNamed(context, '/selectDifficult',arguments: {
              'currentDifficult' : currentDifficult
            });
            setState(() {
              data={
                'currentDifficult' :
result['currentDifficult']
              };

```

```

        });},
    ),
),
Card(
    margin: EdgeInsets.symmetric(vertical: 0,horizontal:
10),

    color: Colors.white,
    child: ListTile(
        title: Text("О приложении", style:
TextStyle(fontSize: 25.0,color: SpecialTextColor)),
        leading: Icon(MdiIcons.information,size: 40,color:
UsualTextColor,),
        onTap: (){},
    ),
),
Card(
    margin: EdgeInsets.symmetric(vertical:
15,horizontal: 10),
    color: Colors.white,
    child: ListTile(
        title: Text("Выйти", style: TextStyle(fontSize:
25.0,color: SpecialTextColor)),
        leading: Icon(MdiIcons.vk,size: 40,color:
UsualTextColor,),
        onTap: (){
            FlutterVKSdk.logout();
            Navigator.pushNamed(context, '/authorization');
        },
    ),
),
],
),
),
]);
}

```

```

void _onItemTapped(int index) {
    setState(() {

```

```

        _selectedWidgetIndex = index;
    });
}

@override
void initState() {
    super.initState();
    WidgetsBinding.instance.addObserver(this);
}

@override
void dispose() {
    WidgetsBinding.instance.removeObserver(this);
    super.dispose();
}

@override
void didChangeAppLifecycleState(AppLifecycleState state) {
    if(state==AppLifecycleState.inactive){
        Database.updLastDifficult(id, currentDifficult);
    }
}

@override
Widget build(BuildContext context) {

    data= data.isNotEmpty?data :
ModalRoute.of(context).settings.arguments;

    firstName = data['firstName'] == null? firstName:
data['firstName'];
    lastName = data['lastName']== null? lastName: data['lastName'];
    avatar = data['avatar']== null? avatar: data['avatar'];
    id = data['user_id'] == null? id : data['user_id'];
    currentDifficult = data['currentDifficult'] == null?
currentDifficult: data['currentDifficult'];

    updateWidgets();
    return Scaffold(

```

```

        backgroundColor: Colors.white,
        bottomNavigationBar: BottomNavigationBar(
          backgroundColor: Colors.lightBlue[700],
          items: const <BottomNavigationBarItem>[
            BottomNavigationBarItem(
              icon: Icon(MdiIcons.glasses),
              title: Text('Задания'),
            ),
            BottomNavigationBarItem(
              icon: Icon(MdiIcons.book),
              title: Text('Теория'),
            ),
            BottomNavigationBarItem(
              icon: Icon(MdiIcons.accountSettings),
              title: Text('Настройки'),
            ),
          ],
          currentIndex: _selectedWidgetIndex,
          unselectedItemColor: Colors.white,
          selectedItemColor: SpecialTextColor,
          onTap: _onItemTapped,
        ),
        body: SafeArea(
          child: _widgets.elementAt(_selectedWidgetIndex),
        ),
      );
    }
  }
}

```

Листинг Г.7 – Файл medialevel.dart

```

import 'dart:async';
import 'package:flutter/material.dart';
import 'package:flutter_tts/flutter_tts.dart';

class mediaLevel extends StatefulWidget {
  @override
  _mediaLevelState createState() => _mediaLevelState();
}

enum TtsState { playing, stopped }

```

```

class _mediaLevelState extends State<mediaLevel> {

  FlutterTts flutterTts;
  dynamic languages;
  String language = "en-US";
  double volume = 0.5;
  double pitch = 1;
  double rate = 1;
  String _newVoiceText;
  TtsState ttsState = TtsState.stopped;
  get isPlaying => ttsState == TtsState.playing;
  get isStopped => ttsState == TtsState.stopped;

  int resultPoints = 0;
  int taskIndex = 0;
  List<Map<String,dynamic>> mapOfTasks=[];
  String finalAnswer='';
  List<bool>  listOfButtonsStates=[];
  List<Widget> listOfButtons=[];
  List<Color> listOfColors = [];
  bool buttonPushed = false;
  bool shouldRefreshData = true;
  var finalAnswerColor = Colors.black;
  String rightAnswer='';
  List<String> answers;

  String level_number;
  String user_id;
  static const String Level_Type = 'media';
  String level_difficult;
  var _controller = TextEditingController();

  @override
  initState() {
    super.initState();
    initTts();
    flutterTts.setLanguage(language);
  }
}

```

```

initTts() {
  flutterTts = FlutterTts();

  _getLanguages();

  flutterTts.setStartHandler(() {
    setState(() {
      print("playing");
      ttsState = TtsState.playing;
    });
  });

  flutterTts.setCompletionHandler(() {
    setState(() {
      print("Complete");
      ttsState = TtsState.stopped;
    });
  });

  flutterTts.setErrorHandler((msg) {
    setState(() {
      print("error: $msg");
      ttsState = TtsState.stopped;
    });
  });
}

Future _getLanguages() async {
  languages = await flutterTts.getLanguages;
  if (languages != null) setState(() => languages);
}

Future _speak() async {
  _newVoiceText=mapOfTasks[taskIndex]['answer'].toString();
  await flutterTts.setVolume(volume);
  await flutterTts.setSpeechRate(rate);
  await flutterTts.setPitch(pitch);
}

```

```

    if (_newVoiceText != null) {
      if (_newVoiceText.isNotEmpty) {
        var result = await flutterTts.speak(_newVoiceText);
        if (result == 1) setState(() => ttsState = TtsState.playing);
      }
    }
  }

Future _stop() async {
  var result = await flutterTts.stop();
  if (result == 1) setState(() => ttsState = TtsState.stopped);
}

@override
void dispose() {
  super.dispose();
  flutterTts.stop();
}

void checkAnswers(){
  setState(() {
    rightAnswer = mapOfTasks[taskIndex]['answer'].toString();
    rightAnswer= rightAnswer.toLowerCase();
    finalAnswer=finalAnswer.trimRight();
    if(finalAnswer == rightAnswer){
      finalAnswerColor = Colors.green;
      resultPoints++;
      //Все правильно
    }else{
      finalAnswerColor = Colors.red;
    }
  });
}

List<Widget> WidgetinitAnswerFlatButtons(){
  listOfButtons=[];
  if(shouldRefreshData) {
    finalAnswerColor = Colors.black;
    listOfButtonsStates=[];

```

```

listOfColors=[];
answers = [
    ...mapOfTasks[taskIndex]['answer'].toString().split(' '),
    ...mapOfTasks[taskIndex]['additional'].toString().split(' ')
];
answers.shuffle();
finalAnswer = '';
rightAnswer = '';
_speak();
shouldRefreshData = false;
for(int i=0;i<answers.length;i++){
    listOfButtonsStates.add(false);
    listOfColors.add(Colors.lightBlue);
}
}

listOfButtons.add(Text(rightAnswer,style: TextStyle(fontSize:
20)));
listOfButtons.add(
    Container(
        padding: EdgeInsets.symmetric(vertical: 0, horizontal:
10),

        child:
        TextField(
            style: TextStyle(color: finalAnswerColor),
            controller: _controller,
            decoration: InputDecoration(
                border: InputBorder.none,
                hintText: 'Введите ответ',
            ),
            onChanged: (text){
                finalAnswer = text.toLowerCase();
            },
        )

    ));
return listOfButtons;
}

Widget initPlayer(){

```

```

Row row;
switch (mapOfTasks[taskIndex]['type']) {
    case 'audio':
        row = Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
                IconButton(
                    icon: Icon(Icons.volume_up, color: Colors.lightBlue,),
                    onPressed: isPlaying? _stop : _speak,
                ),
                Slider(
                    value: rate,
                    onChanged: (newRate) {
                        setState(() => rate = newRate);
                    },
                    min: 0.0,
                    max: 1.0,
                    divisions: 10,
                    label: "Скорость печи: $rate",
                    activeColor: Colors.lightBlue,
                )
            ],
        );

        break;
    case 'video':
        row =Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
                Column(
                    children: <Widget>[
                        Text('В разработке'),
                        CircularProgressIndicator()
                    ],
                )
            ],
        );

        break;
}

```

```

        return row;
    }

Widget initMediaTask() {

    return Scaffold(
        body: SafeArea(
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.center,
                children: <Widget>[
                    SizedBox(height: 10,),
                    LinearProgressIndicator(
                        value: taskIndex.toDouble()*25*0.01,
                        valueColor: AlwaysStoppedAnimation<Color>(Colors.blue),
                        backgroundColor: Colors.grey,
                    ),
                    SizedBox(height: 50,),
                    Card(
                        margin: EdgeInsets.symmetric(vertical: 15,horizontal:
10),

                        color: Colors.lightBlue,
                        child: ListTile(
                            title:RichText(
                                textAlign: TextAlign.center,
                                text: TextSpan(
                                    text: mapOfTasks[taskIndex]['task'],
                                    style: TextStyle(
                                        fontWeight: FontWeight.bold,
                                        fontSize: 22,
                                        color: Colors.black),
                                ),
                            ),
                        ),
                    ),
                    SizedBox(height: 18,),
                    initPlayer(),//сюда вставляем аудио/видео блок
                    SizedBox(height: 18,),
                    Column(

```

```

        children: listOfButtons,
      ),
    Flexible(child: FractionallySizedBox(heightFactor:
0.7,)),),
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        FlatButton(
          onPressed: () async{
            bool next = false;
            setState(() {
              if(!buttonPushed) {
                buttonPushed = true;
                checkAnswers();
                return;
              }else {
                _controller.clear();
                buttonPushed = false;
                if(taskIndex!=3)
                  shouldRefreshData = true;
              }
              if(taskIndex!=3)
                taskIndex++;
              else
                next = true;
            });
            if(next){
              await Navigator.pushNamed(context,
'/resultLevel', arguments: {
                'resultPoints' : resultPoints,
                'user_id' : user_id,
                'level_difficult' : level_difficult,
                'level_type' : Level_Type,
                'level_number' : level_number});
              Navigator.pop(context);
            }
          },
          shape: StadiumBorder(),
          color: Colors.lightBlue,

```

```

        child: Text("Далее"),
      ),
    ],
  )
],
),
),
);
}

@override
Widget build(BuildContext context) {

  if (mapOfTasks.isEmpty) {
    final Map<String,dynamic> data =
ModalRoute.of(context).settings.arguments;
    user_id = data['user_id'];
    level_difficult = data['difficult'];
    level_number = data['level_number'];
    for(int i = 0;i<4;i++){
      mapOfTasks.add(data['$i']);
    }
  }
  WidgetinitAnswerFlatButtons();
  return initMediaTask();
}
}

```

Листинг Г.8 – Файл selectDifficult.dart

```

import 'dart:async';
import 'package:flutter/material.dart';
import 'package:flutter_tts/flutter_tts.dart';

class mediaLevel extends StatefulWidget {
  @override
  _mediaLevelState createState() => _mediaLevelState();
}

enum TtsState { playing, stopped }

```

```

class _mediaLevelState extends State<mediaLevel> {

  FlutterTts flutterTts;
  dynamic languages;
  String language = "en-US";
  double volume = 0.5;
  double pitch = 1;
  double rate = 1;
  String _newVoiceText;
  TtsState ttsState = TtsState.stopped;
  get isPlaying => ttsState == TtsState.playing;
  get isStopped => ttsState == TtsState.stopped;

  int resultPoints = 0;
  int taskIndex = 0;
  List<Map<String,dynamic>> mapOfTasks=[];
  String finalAnswer='';
  List<bool>  listOfButtonsStates=[];
  List<Widget> listOfButtons=[];
  List<Color> listOfColors = [];
  bool buttonPushed = false;
  bool shouldRefreshData = true;
  var finalAnswerColor = Colors.black;
  String rightAnswer='';
  List<String> answers;

  String level_number;
  String user_id;
  static const String Level_Type = 'media';
  String level_difficult;
  var _controller = TextEditingController();

  @override
  initState() {
    super.initState();
    initTts();
    flutterTts.setLanguage(language);
  }
}

```

```

initTts() {
  flutterTts = FlutterTts();

  _getLanguages();

  flutterTts.setStartHandler(() {
    setState(() {
      print("playing");
      ttsState = TtsState.playing;
    });
  });

  flutterTts.setCompletionHandler(() {
    setState(() {
      print("Complete");
      ttsState = TtsState.stopped;
    });
  });

  flutterTts.setErrorHandler((msg) {
    setState(() {
      print("error: $msg");
      ttsState = TtsState.stopped;
    });
  });
}

Future _getLanguages() async {
  languages = await flutterTts.getLanguages;
  if (languages != null) setState(() => languages);
}

Future _speak() async {
  _newVoiceText=mapOfTasks[taskIndex]['answer'].toString();
  await flutterTts.setVolume(volume);
  await flutterTts.setSpeechRate(rate);
  await flutterTts.setPitch(pitch);

  if (_newVoiceText != null) {

```

```

        if (_newVoiceText.isNotEmpty) {
          var result = await flutterTts.speak(_newVoiceText);
          if (result == 1) setState(() => ttsState = TtsState.playing);
        }
      }
    }
  }
}

```

```

Future _stop() async {
  var result = await flutterTts.stop();
  if (result == 1) setState(() => ttsState = TtsState.stopped);
}

```

```

@override
void dispose() {
  super.dispose();
  flutterTts.stop();
}

```

```

void checkAnswers(){
  setState(() {
    rightAnswer = mapOfTasks[taskIndex]['answer'].toString();
    rightAnswer= rightAnswer.toLowerCase();
    finalAnswer=finalAnswer.trimRight();
    if(finalAnswer == rightAnswer){
      finalAnswerColor = Colors.green;
      resultPoints++;
      //Все правильно
    }else{
      finalAnswerColor = Colors.red;
    }
  });
}

```

```

List<Widget> WidgetinitAnswerFlatButtons(){
  listOfButtons=[];
  if(shouldRefreshData) {
    finalAnswerColor = Colors.black;
    listOfButtonsStates=[];
    listOfColors=[];
  }
}

```

```

        answers = [
            ...mapOfTasks[taskIndex]['answer'].toString().split(' '),
            ...mapOfTasks[taskIndex]['additional'].toString().split(' ')
        ];
        answers.shuffle();
        finalAnswer = '';
        rightAnswer = '';
        _speak();
        shouldRefreshData = false;
        for(int i=0;i<answers.length;i++){
            listOfButtonsStates.add(false);
            listOfColors.add(Colors.lightBlue);
        }
    }

    listOfButtons.add(Text(rightAnswer,style: TextStyle(fontSize:
20)));
    listOfButtons.add(
        Container(
            padding: EdgeInsets.symmetric(vertical: 0, horizontal:
10),

            child:
                TextField(
                    style: TextStyle(color: finalAnswerColor),
                    controller: _controller,
                    decoration: InputDecoration(
                        border: InputBorder.none,
                        hintText: 'Введите ответ',
                    ),
                    onChanged: (text){
                        finalAnswer = text.toLowerCase();
                    },
                )
            ));
    return listOfButtons;
}

Widget initPlayer(){
    Row row;

```

```

switch (mapOfTasks[taskIndex]['type']) {
    case 'audio':
        row = Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
                IconButton(
                    icon: Icon(Icons.volume_up, color: Colors.lightBlue,),
                    onPressed: isPlaying? _stop : _speak,
                ),
                Slider(
                    value: rate,
                    onChanged: (newRate) {
                        setState(() => rate = newRate);
                    },
                    min: 0.0,
                    max: 1.0,
                    divisions: 10,
                    label: "Скорость печи: $rate",
                    activeColor: Colors.lightBlue,
                )
            ],
        );

        break;
    case 'video':
        row =Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
                Column(
                    children: <Widget>[
                        Text('В разработке'),
                        CircularProgressIndicator()
                    ],
                )
            ],
        );
        break;
}
return row;

```

```

}

Widget initMediaTask() {

  return Scaffold(
    body: SafeArea(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        children: <Widget>[
          SizedBox(height: 10,),
          LinearProgressIndicator(
            value: taskIndex.toDouble()*25*0.01,
            valueColor: AlwaysStoppedAnimation<Color>(Colors.blue),
            backgroundColor: Colors.grey,
          ),
          SizedBox(height: 50,),
          Card(
            margin: EdgeInsets.symmetric(vertical: 15,horizontal:
10),

            color: Colors.lightBlue,
            child: ListTile(
              title:RichText(
                textAlign: TextAlign.center,
                text: TextSpan(
                  text: mapOfTasks[taskIndex]['task'],
                  style: TextStyle(
                    fontWeight: FontWeight.bold,
                    fontSize: 22,
                    color: Colors.black),
                ),
              ),
            ),
          ),
          SizedBox(height: 18,),
          initPlayer(),//сюда вставляем аудио/видео блок
          SizedBox(height: 18,),
          Column(
            children: listOfButtons,

```

```

),
Flexible(child: FractionallySizedBox(heightFactor:
0.7,)),),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: <Widget>[
    FlatButton(
      onPressed: () async{
        bool next = false;
        setState(() {
          if(!buttonPushed) {
            buttonPushed = true;
            checkAnswers();
            return;
          }else {
            _controller.clear();
            buttonPushed = false;
            if(taskIndex!=3)
              shouldRefreshData = true;
          }
          if(taskIndex!=3)
            taskIndex++;
          else
            next = true;
        });
        if(next){
          await Navigator.pushNamed(context,
'/resultLevel', arguments: {
            'resultPoints' : resultPoints,
            'user_id' : user_id,
            'level_difficult' : level_difficult,
            'level_type' : Level_Type,
            'level_number' : level_number});
          Navigator.pop(context);
        }
      },
      shape: StadiumBorder(),
      color: Colors.lightBlue,
      child: Text("Далее"),

```

```

        ),
      ],
    ),
  ],
),
),
);
}

@override
Widget build(BuildContext context) {

  if (mapOfTasks.isEmpty) {
    final Map<String, dynamic> data =
ModalRoute.of(context).settings.arguments;
    user_id = data['user_id'];
    level_difficult = data['difficult'];
    level_number = data['level_number'];
    for (int i = 0; i < 4; i++) {
      mapOfTasks.add(data['$i']);
    }
  }
  WidgetinitAnswerFlatButtons();
  return initMediaTask();
}
}

```

Листинг Г.9 – Файл select_level.dart

```

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import
'package:material_design_icons_flutter/material_design_icons_flutter.d
art';
import 'package:lang_guide/services/CustomSelectLevelButton.dart';
import 'package:lang_guide/services/database.dart';

class SelectLevel extends StatefulWidget {
  @override

```

```

    _SelectLevelState createState() => _SelectLevelState();
}

class _SelectLevelState extends State<SelectLevel> {

    String user_id;
    String levelType;
    String levelDifficult;
    String titleText;
    String levelIndex;
    Color iconsColor = Colors.white;
    Map data;
    List<Map<dynamic,dynamic>> listOfTasks;
    List<Map<String,dynamic>> listOfDonuts;
    Map<String,dynamic> mapOfDonuts;
    Future<Map<String,dynamic>> futureMapOfDonuts;

    static const List<IconData> iconDataList=[
        MdiIcons.numeric1,
        MdiIcons.numeric2,
        MdiIcons.numeric3,
        MdiIcons.numeric4,
        MdiIcons.numeric5,
        MdiIcons.numeric6,
        MdiIcons.numeric7,
        MdiIcons.numeric8
    ];

    Future<Map<String,dynamic>> getAllDonuts() async {
        listOfDonuts=[];
        if (mapOfDonuts==null)
            mapOfDonuts = Map<String,dynamic>();
        listOfDonuts = await Database.getCompletedLevels(user_id,
levelDifficult, levelType);
        for (int i=0;i<listOfDonuts.length;i++){
            mapOfDonuts['${listOfDonuts[i]['level_number']}']
=listOfDonuts[i]['points'];
        }
        //print (mapOfDonuts);
    }
}

```

```

        return mapOfDonuts;
    }

    int getDonutsOfLevel(int level){
        if(mapOfDonuts['$level']!=null){
            print('level: $level, donuts:
${int.parse(mapOfDonuts['$level'])}');
            return int.parse(mapOfDonuts['$level']);
        }
        return 0;
    }

    void initDifficult(int index) async{
        switch(levelDifficult){
            case 'Начинающий':
                return await initEasy(index);
            case 'Средний':
                return initMedium(index);
            case 'Продвинутый':
                return initHard(index);
        }
    }

    void initEasy(int index) async{
        levelIndex = index.toString();
        switch(levelType){
            case 'text':
                listOfTasks = await Database.getEasyTextTasks(index);
                break;
            case 'image':
                listOfTasks = await Database.getEasyImageTasks(index);
                break;
            case 'media':
                listOfTasks = await Database.getEasyMediaTasks(index);
                break;
        }
    }

    void initMedium(int index){
        switch(levelType){

```

```

        case 'text':
            break;
        case 'image':
            break;
        case 'media':
            break;
    }
}

void initHard(int index){
    switch(levelType){
        case 'text':
            break;
        case 'image':
            break;
        case 'media':
            break;
    }
}

List<Widget> initButtons(){
    List<Widget> listOfButtons=[];
    listOfButtons.add(SizedBox(height: 25,,));
    bool odd = false;
    for(int i=0;i<8;i++){
        if(odd){
            listOfButtons.add(Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                    CustomSelecetLevelButton(donutCount:
getDonutsOfLevel(i+1),iconData: iconDataList[i],onTap:())async{
                    await initDifficult(i+1);
                    await Navigator.pushNamed(context,
'/${levelType}Level',arguments: {
                        'user_id' : user_id,
                        'difficult' : levelDifficult,
                        'level_number' : levelIndex,
                        '0' : listOfTasks[0],
                        '1' : listOfTasks[1],
                        '2' : listOfTasks[2],

```

```

        '3' : listOfTasks[3]
    });
    setState(() {
        levelType = data['levelType'];
    });
    ),
    ],),
);
}else{
    listOfButtons.add(Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
            CustomSelecetLevelButton(donutCount:
getDonutsOfLevel(i+1), iconData: iconDataList[i++],onTap:() async{
            await initDifficult(i);
            await Navigator.pushNamed(context,
'/${levelType}Level',arguments: {
                'user_id' : user_id,
                'difficult' : levelDifficult,
                'level_number' : levelIndex,
                '0' : listOfTasks[0],
                '1' : listOfTasks[1],
                '2' : listOfTasks[2],
                '3' : listOfTasks[3]
            });
            setState(() {
                levelType = data['levelType'];
            });
        }),
        Flexible(
            child: FractionallySizedBox(widthFactor: 0.4,),
        ),
        CustomSelecetLevelButton(donutCount:
getDonutsOfLevel(i+1), iconData: iconDataList[i],onTap:() async{
            await initDifficult(i+1);
            await Navigator.pushNamed(context,
'/${levelType}Level',arguments: {
                'user_id' : user_id,
                'difficult' : levelDifficult,

```

```

        'level_number' : levelIndex,
        '0' : listOfTasks[0],
        '1' : listOfTasks[1],
        '2' : listOfTasks[2],
        '3' : listOfTasks[3]
    });
    setState(() {
        levelType = data['levelType'];
    });
    }),
    ],
    )
    );
}
odd=!odd;
}
return listOfButtons;
}

@override
void initState() {
    super.initState();
}

@override
Widget build(BuildContext context) {
    data= ModalRoute.of(context).settings.arguments;
    levelType = data['levelType'];
    levelDifficult = data['levelDifficult'];
    user_id = data['user_id'];
    futureMapOfDonuts = getAllDonuts();
    intTitle();

    return Scaffold(
        appBar: AppBar(
            title: Text(titleText),
        ),
        body: FutureBuilder<Map<String,dynamic>>(
            future: futureMapOfDonuts,

```

```

        builder: (BuildContext context,
AsyncSnapshot<Map<String,dynamic>> snapshot){
        if(snapshot.hasData){
            mapOfDonuts = snapshot.data;
            return Column(
                children: initButtons(),
            );
        }else{
            return Center(child: CircularProgressIndicator());
        }
    },
    )
);
}
intTitle(){
    switch(levelType){
        case 'text':
            titleText='Текстовые задания';
            break;
        case 'image':
            titleText='Задания с картинками';
            break;
        case 'media':
            titleText='Задания с медиа';
            break;
    }
}
}
}

```

Листинг Г.10 – Файл theoryPage.dart

```

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';

class theoryPage extends StatefulWidget {
    @override
    _theoryPageState createState() => _theoryPageState();
}

```

```

class _theoryPageState extends State<theoryPage> {

  List<TextSpan> changeToBold(String build,String eng,String rus,
String description){
    List<TextSpan> resultList=[];
    TextStyle txtStyle = TextStyle(color: Colors.black,fontSize: 12);
    TextStyle txtPr = TextStyle(color: Colors.lightBlue,fontSize: 12);
    TextStyle verbStyle = TextStyle(color: Colors.black, fontWeight:
FontWeight.bold);
    var arr=build.split('`');
    for(int i=0;i<arr.length;i++){
      resultList.add(TextSpan(style: arr[i]=='V'?verbStyle:txtStyle,
text: arr[i]));
    }
    resultList.add(TextSpan(style: txtPr,text: eng));
    resultList.add(TextSpan(style: txtPr,text: rus));
    resultList.add(TextSpan(style: txtStyle,text: description));
    return resultList;
  }

  Widget initTimes(String state){

    Map mapOfTimes={
      'statement':{
        'simple': {
          'present': {
            'description' : 'Повторяющиеся действия\nДействия,
следующие одно за другим\nПланируемые события',
            'build' : 'V`, `V`s',
            'eng' : 'I read.',
            'rus' : 'Я читаю. (обычно, часто)'
          },
          'past' : {
            'description' : 'Действие, произошедшее в прошлом (не
связанное с текущим моментом)\nПривычные, регулярные действия',
            'build' : 'V`ed, `V`2',
            'eng' : 'I readed.',
            'rus' : 'Я читал. (вчера)'
          }
        },

```

```

    'future' : {
        'description' : 'Предположение о ходе будущих
действий\nВ условных предложениях (If + Present Simple + will do)',
        'build' : 'will + `V`',
        'eng' : 'I will read.',
        'rus' : 'Я прочитаю. (завтра)',
    }
},
'continuous': {
    'present': {
        'description' : 'Действие, происходящее в текущий
момент\nЗапланированные будущие действия',
        'build' : 'am/is/are + `V`ing',
        'eng' : 'I am reading.',
        'rus' : 'Я читаю. (сейчас)',
    },
    'past' : {
        'description' : 'Длительное действие в
прошлом\nОдновременные действия',
        'build' : 'was/were + `V`ing',
        'eng' : 'I was reading.',
        'rus' : 'Я читал. (вчера, в 2 часа)',
    },
    'future' : {
        'description' : 'Действие, которое обязательно
произойдет в будущем',
        'build' : 'will + `V`',
        'eng' : 'I will read.',
        'rus' : 'Я прочитаю. (завтра)',
    }
},
'perfect' : {
    'present': {
        'description' : 'Завершенное действие, связанное с
настоящим\nДействие, идущее из прошлого по настоящий момент',
        'build' : 'have/has + `V`ed, `V`3',
        'eng' : 'I have readed.',
        'rus' : 'Я прочитал. (уже, только что)',
    },

```

```

    'past' : {
        'description' : 'Прошлое действие, которое началось и
завершилось к определенному моменту в прошлом',
        'build' : 'had + `V`ed, `V`3',
        'eng' : 'I had readed.',
        'rus' : 'Я прочитал. (вчера к 3м часам, до того
как...) ',
    },
    'future' : {
        'description' : 'Действие, которое завершится в
определенный момент в будущем',
        'build' : 'will + have + `V`3',
        'eng' : 'I will have read.',
        'rus' : 'Я прочитаю. (завтра к 3м часам, до того,
как...) ',
    }
},
'perfectContinuous' : {
    'present': {
        'description' : 'Действие, начавшееся в прошлом и только
что завершившееся',
        'build' : 'have/has + been + `V`ing',
        'eng' : 'I have been reading.',
        'rus' : 'Я читаю. (уже час, с 2х часов) ',
    },
    'past' : {
        'description' : 'Длительное действие, происходившее
перед другим действием',
        'build' : 'had + been + `V`ing',
        'eng' : 'I had been reading.',
        'rus' : 'Я читал. (уже 2 часа, когда он пришел) ',
    },
    'future' : {
        'description' : 'Действие, которое завершится до
определенного события\n(Почти не используется)',
        'build' : 'will + have + been + `V`ing',
        'eng' : 'I will have been reading.',
        'rus' : 'Я буду читать. (завтра, к тому времени, когда
он придет) ',
    }
}

```

```

        }
    }
},
};

TextStyle txtStyle = TextStyle(color: Colors.black);
TextStyle verbStyle = TextStyle(color: Colors.pink, fontSize: 13);
TextStyle rowNameStyle = TextStyle(color: Colors.pink, fontSize:
15);

Table table=Table(
    border: TableBorder.symmetric(inside: BorderSide(color:
Colors.black,width: 1)),
    children: <TableRow>[
        TableRow(
            children: <Widget>[
                Text(' '),
                Text('Настоящее (Present)',style: verbStyle),
                Text('Прошедшее (Past)',style: verbStyle),
                Text('Будущее (Future)',style: verbStyle),
            ]
        ),
        TableRow(
            children: <Widget>[
                Column(children: <Widget>[
                    Text('Simple\n\n',style: rowNameStyle),
                    Image.asset('assets/donut.png',width: 25,height: 25),
                    Text('\n\n'),
                    Image.asset('assets/donut.png',width: 25,height: 25,)
                ],),
                RichText(
                    textAlign: TextAlign.center,
                    text: TextSpan(
                        style: txtStyle,
                        children:
changeToBold(mapOfTimes[state]['simple']['present']['build']+'\n\n',

mapOfTimes[state]['simple']['present']['eng']+'\n',

mapOfTimes[state]['simple']['present']['rus']+'\n\n',

```

```

mapOfTimes[state]['simple']['present']['description']+'\\n')
        ),
    ),
    RichText(
        textAlign: TextAlign.center,
        text: TextSpan(
            style: txtStyle,
            children:
changeToBold(mapOfTimes[state]['simple']['past']['build']+'\\n\\n',

mapOfTimes[state]['simple']['past']['eng']+'\\n',

mapOfTimes[state]['simple']['past']['rus']+'\\n\\n',

mapOfTimes[state]['simple']['past']['description']+'\\n')
        ),
    ),
    RichText(
        textAlign: TextAlign.center,
        text: TextSpan(
            style: txtStyle,
            children:
changeToBold(mapOfTimes[state]['simple']['future']['build']+'\\n\\n',

mapOfTimes[state]['simple']['future']['eng']+'\\n',

mapOfTimes[state]['simple']['future']['rus']+'\\n\\n',

mapOfTimes[state]['simple']['future']['description']+'\\n')
        ),
    ),
]
),
TableRow(
    children: <Widget>[
        Column(children: <Widget>[
            Text('Continuous\\n\\n',style: rowNameStyle,),
            Image.asset('assets/donut.png',width: 25,height: 25,)),

```

```

        Text('\n\n'),
        Image.asset('assets/donut.png',width: 25,height: 25,)
    ],),
    RichText(
        textAlign: TextAlign.center,
        text: TextSpan(
            style: txtStyle,
            children:
changeToBold(mapOfTimes[state]['continuous']['present']['build']+'\n\n
',

mapOfTimes[state]['continuous']['present']['eng']+'\n',

mapOfTimes[state]['continuous']['present']['rus']+'\n\n',

mapOfTimes[state]['continuous']['present']['description']+'\n')
        ),
    ),
    RichText(
        textAlign: TextAlign.center,
        text: TextSpan(
            style: txtStyle,
            children:
changeToBold(mapOfTimes[state]['continuous']['past']['build']+'\n\n',

mapOfTimes[state]['continuous']['past']['eng']+'\n',

mapOfTimes[state]['continuous']['past']['rus']+'\n\n',

mapOfTimes[state]['continuous']['past']['description']+'\n')
        ),
    ),
    RichText(
        textAlign: TextAlign.center,
        text: TextSpan(
            style: txtStyle,
            children:
changeToBold(mapOfTimes[state]['continuous']['future']['build']+'\n\n\
n',

```

```

mapOfTimes[state]['continuous']['future']['eng']+'\n',

mapOfTimes[state]['continuous']['future']['rus']+'\n\n',

mapOfTimes[state]['continuous']['future']['description']+'\n')
    ),
    ),
]
),
TableRow(
    children: <Widget>[
        Column(children: <Widget>[
            Text('Perfect\n\n',style: rowNameStyle,),
            Image.asset('assets/donut.png',width: 25,height: 25,),
            Text('\n\n\n'),
            Image.asset('assets/donut.png',width: 25,height: 25,)
        ],),
        RichText(
            textAlign: TextAlign.center,
            text: TextSpan(
                style: txtStyle,
                children:
changeToBold(mapOfTimes[state]['perfect']['present']['build']+'\n\n',

mapOfTimes[state]['perfect']['present']['eng']+'\n',

mapOfTimes[state]['perfect']['present']['rus']+'\n\n',

mapOfTimes[state]['perfect']['present']['description']+'\n')
            ),
            ),
            RichText(
                textAlign: TextAlign.center,
                text: TextSpan(
                    style: txtStyle,
                    children:
changeToBold(mapOfTimes[state]['perfect']['past']['build']+'\n\n',

```

```

mapOfTimes[state]['perfect']['past']['eng']+'\\n',

mapOfTimes[state]['perfect']['past']['rus']+'\\n\\n',

mapOfTimes[state]['perfect']['past']['description']+'\\n')
    ),
    ),
    RichText(
        textAlign: TextAlign.center,
        text: TextSpan(
            style: txtStyle,
            children:
changeToBold(mapOfTimes[state]['perfect']['future']['build']+'\\n\\n',

mapOfTimes[state]['perfect']['future']['eng']+'\\n',

mapOfTimes[state]['perfect']['future']['rus']+'\\n\\n',

mapOfTimes[state]['perfect']['future']['description']+'\\n')
        ),
    ),
    ],
    ),
    TableRow(
        children: <Widget>[
            Column(children: <Widget>[
                Text('Perfect Continuous\\n\\n',style: rowNameStyle,),
                Image.asset('assets/donut.png',width: 25,height: 25,),
                Text('\\n\\n'),
                Image.asset('assets/donut.png',width: 25,height: 25,)
            ],),
            RichText(
                textAlign: TextAlign.center,
                text: TextSpan(
                    style: txtStyle,
                    children:
changeToBold(mapOfTimes[state]['perfectContinuous']['present']['build'
]+'\\n\\n',

```

```

mapOfTimes[state]['perfectContinuous']['present']['eng']+'\n',

mapOfTimes[state]['perfectContinuous']['present']['rus']+'\n\n',

mapOfTimes[state]['perfectContinuous']['present']['description']+'\n')
    ),
    ),
    RichText(
        textAlign: TextAlign.center,
        text: TextSpan(
            style: txtStyle,
            children:
changeToBold(mapOfTimes[state]['perfectContinuous']['past']['build']+'
\n\n',

mapOfTimes[state]['perfectContinuous']['past']['eng']+'\n',

mapOfTimes[state]['perfectContinuous']['past']['rus']+'\n\n',

mapOfTimes[state]['perfectContinuous']['past']['description']+'\n')
    ),
    ),
    RichText(
        textAlign: TextAlign.center,
        text: TextSpan(
            style: txtStyle,
            children:
changeToBold(mapOfTimes[state]['perfectContinuous']['future']['build']
+'\n\n',

mapOfTimes[state]['perfectContinuous']['future']['eng']+'\n',

mapOfTimes[state]['perfectContinuous']['future']['rus']+'\n\n',

mapOfTimes[state]['perfectContinuous']['future']['description']+'\n')
    ),
    ),
]

```

```

        ),
      ],
    );
    return Scaffold(
      appBar: AppBar(
        title: Text('Времена'),
      ),
      body: SingleChildScrollView(child: Padding(padding:
EdgeInsets.symmetric(vertical: 0,horizontal: 10),child: table)),
    );
  }

Widget initPronouns(){

}

Widget initPrepositions(){

}

Widget initVerbs(){

}

@override
Widget build(BuildContext context) {
  Map data = ModalRoute.of(context).settings.arguments;

  switch(data['pageType']){
    case 'times':
      return initTimes('statement');
    case 'pronouns':
      return initPronouns();
    case 'prepositions':
      return initPrepositions();
    case 'verbs':
      return initVerbs();
  }
  return Container();
}

```

```

    }
}

```

Листинг Г.11 – Файл resultLevel.dart

```

import 'package:flutter/material.dart';
import 'package:lang_guide/services/database.dart';

class ResultLevel extends StatefulWidget {

  @override
  _ResultLevelState createState() => _ResultLevelState();
}

class _ResultLevelState extends State<ResultLevel> with
TickerProviderStateMixin{

  String user_id;
  String level_difficult;
  String level_type;
  String level_number;
  bool blocked =false;

  int resultPoints = 0;
  int maxPoints = 4;
  bool showResult = false;
  List<Widget> listOfDonuts=[];
  AnimationController controller;

  void CountDonuts(){
    listOfDonuts=[];
    controller = AnimationController(
      vsync: this,
      duration: Duration (seconds: 1)
    );
    for(int i = 0;i<maxPoints;i++){
      final animation = Tween(
        begin: 0.0,
        end: resultPoints> i ? 1.0:0.1
      ).animate(controller);

```

```

        listOfDonuts.add(
            FadeTransition(
                opacity: animation ,
                child: Image(image: AssetImage("assets/donut.png"), width:
50, height: 50,)
            )
        );
    }
}
@override
void initState(){
}

@override
void dispose(){
    controller.dispose();
    super.dispose();
}

Future<void> sendData() async{
    await Database.addUpdCompletedLevel(user_id, level_difficult,
level_type, level_number, resultPoints.toString());
}

Widget ShowResult(BuildContext context){
    if(!blocked) {
        Map data = ModalRoute
            .of(context)
            .settings
            .arguments;
        resultPoints = data['resultPoints'];
        user_id = data['user_id'];
        level_difficult = data['level_difficult'];
        level_type = data['level_type'];
        level_number = data['level_number'];
        sendData();
        blocked = true;
    }
}
CountDonuts();

```

```

controller.forward();

return Scaffold(
  body: SafeArea(
    child:
      Table(
        children: <TableRow>[
          TableRow(
            children:<Widget> [
              SizedBox(height: 10,)
            ],
          ),
          TableRow(
            children:<Widget> [
              LinearProgressIndicator(
                value: 1,
                valueColor:
AlwaysStoppedAnimation<Color>(Colors.blue),
                backgroundColor: Colors.grey,
              ),
            ],
          ),
          TableRow(
            children:<Widget> [
              SizedBox(height:
MediaQuery.of(context).size.height*0.35,)
            ],
          ),
          TableRow(
            children:<Widget> [
              Center(
                child: Container(
                  //color: Colors.lightBlue,
                  height:
MediaQuery.of(context).size.height*0.4,
                  child: Column(
                    children: <Widget>[
                      Text("Ваш результат",
                        textAlign: TextAlign.center,

```

```

        style: TextStyle(
          color: Colors.black,
          fontSize: 40
        ),),
      Row(
        mainAxisAlignment:
MainAxisAlignment.center,

        children: listOfDonuts,),
      FlatButton(
        onPressed: () {
          setState(() {
            Navigator.pop(context);
          });
        },
        shape: StadiumBorder(),
        color: Colors.lightBlue,
        child: Text("Готово"),
      ),
    ],
  ),
),
),
],
),
]
),
),
);
}
@override
Widget build(BuildContext context) {
  return ShowResult(context);
}
}

```

ПРИЛОЖЕНИЕ Д
Диаграмма классов приложения
(рекомендуемое)

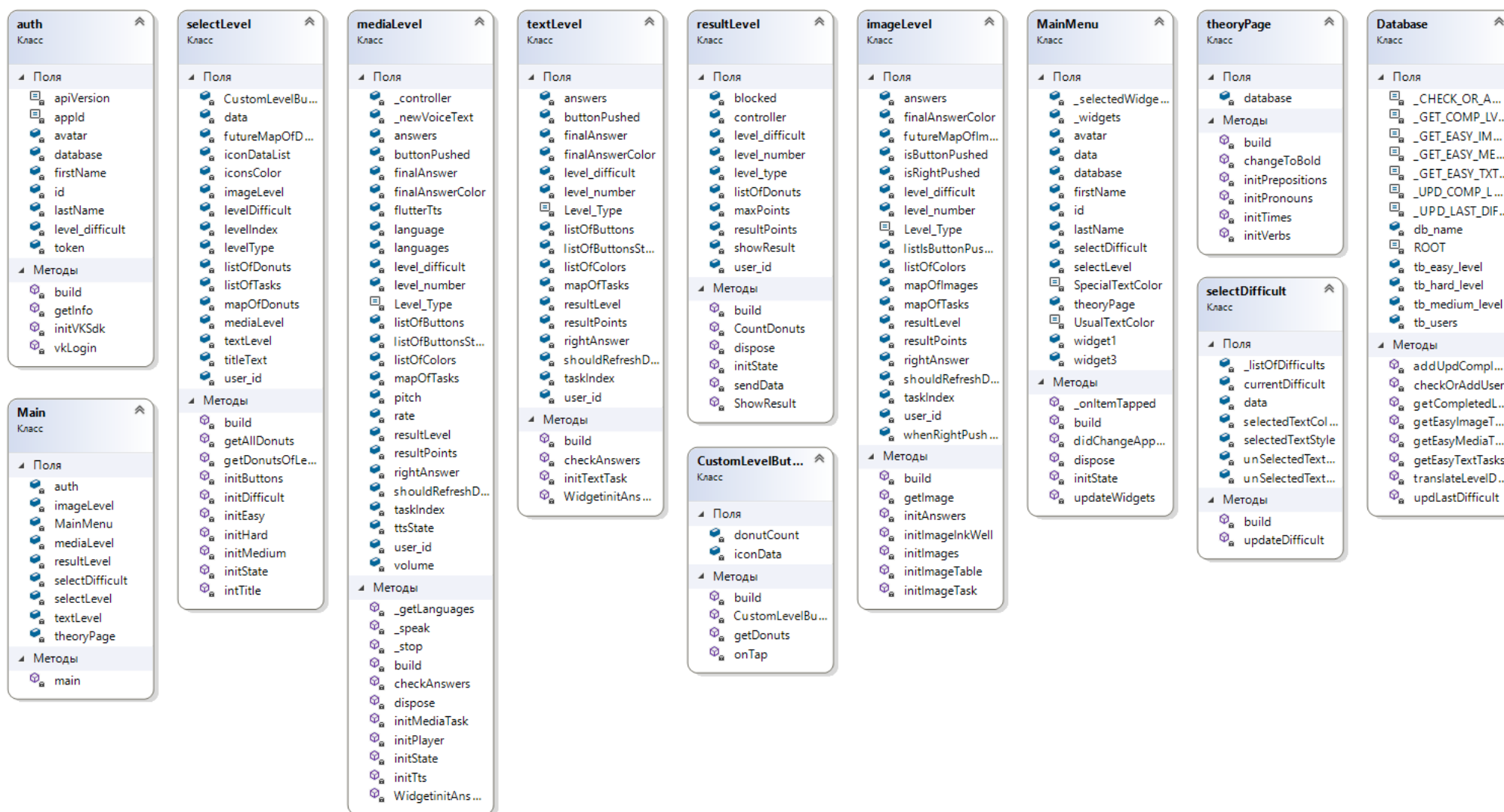


Рисунок Д.1 – Диаграмма классов приложения

ПРИЛОЖЕНИЕ Е

Результаты тестирования серверной части
(рекомендуемое)

POST 192.168.0.106/DbScript.php/ Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE	DESCRIPTION
☒ action	UPD_LAST_DIFF	
☒ user_id	236568364	
☒ last_difficult	Средний	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 79 ms Size: 196 B Save

Pretty Raw Preview Visualize JSON ≡

```
1 success
```

Рисунок Е.1 – Добавление нового пользователя

POST 192.168.0.106/DbScript.php/ Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE	DESCRIPTION
☒ action	ADD_UPD_COMP_LVL	
☒ user_id	236568364	
☒ level_difficult	easy_level	
☒ level_type	text	
☒ level_number	1	
☒ points	4	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 72 ms Size: 211 B Save

Pretty Raw Preview Visualize HTML ≡

```
1 added completed level
```

Рисунок Е.2 – Добавление записи о выполнении текстового задания легкого уровня с номером 1

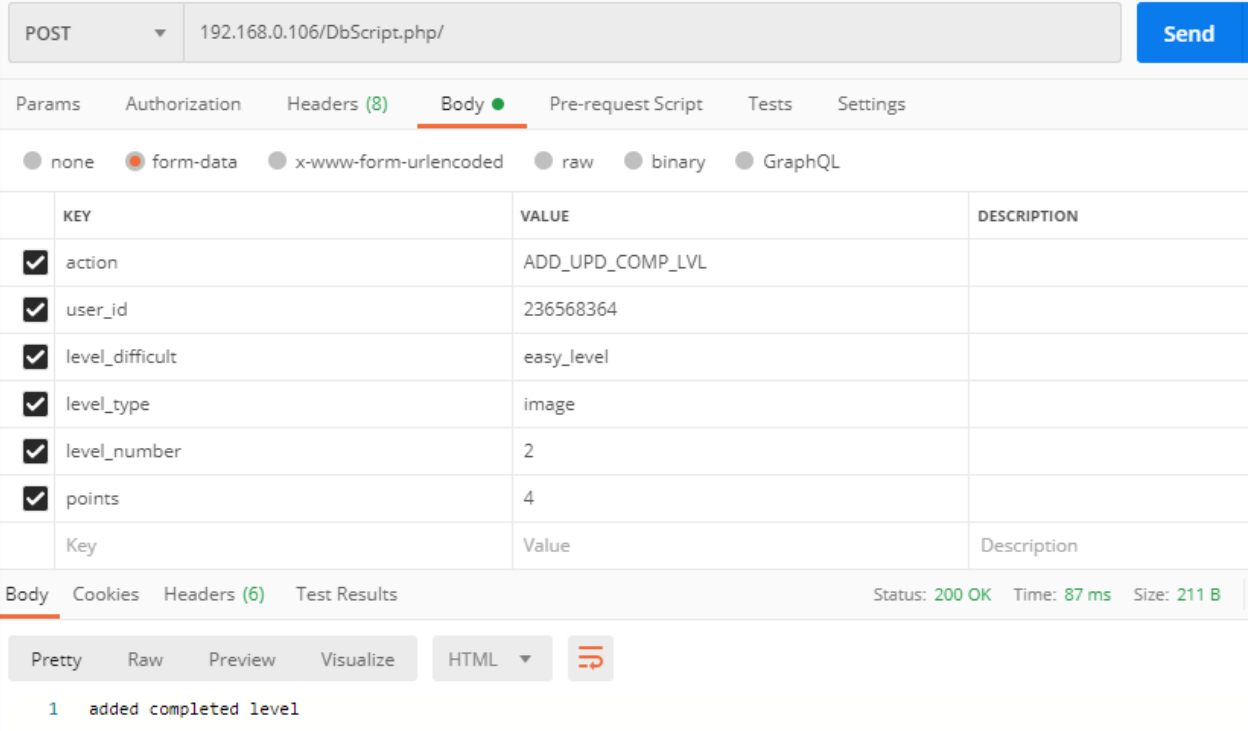


Рисунок Е.3 – Добавление записи о выполнении задания с картинками легкого уровня с номером 2

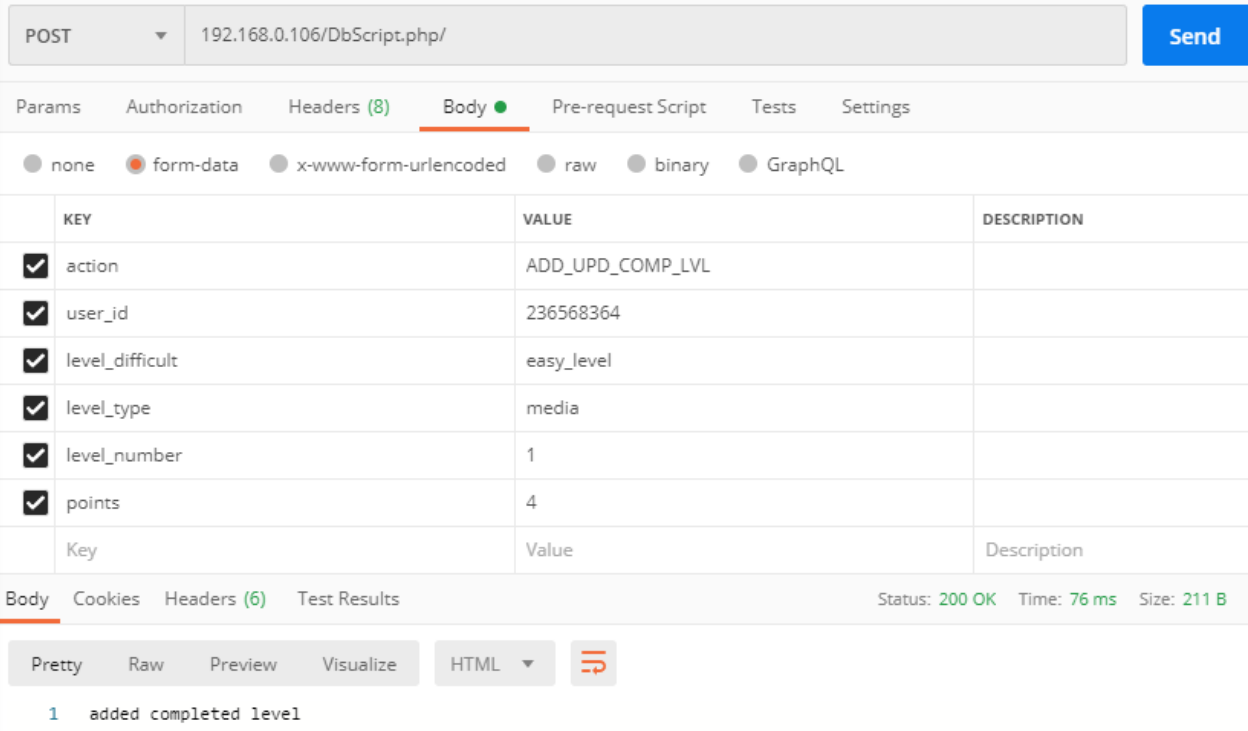


Рисунок Е.4 – Добавление записи о выполнении задания с медиа легкого уровня с номером 1

GET 192.168.0.106/DbScript.php?action=GET_COMP_LVL&user_id=236568364&level_type=text&level_difficult...

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
action	GET_COMP_LVL	
user_id	236568364	
level_type	text	
level_difficult	easy_level	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 8 ms Size: 225 B

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "level_number": "1",
4     "points": "4"
5   }
6 ]

```

Рисунок Е.5 – Получение списка выполненных текстовых заданий легкого уровня

GET 192.168.0.106/DbScript.php?action=GET_EASY_TXT_TASK&user_id=236568364&level_type=text&level=1

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
action	GET_EASY_TXT_TASK	
user_id	236568364	
level_type	text	
level	1	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 11 ms Size: 1.3 KB

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "Переведите предложение": "У меня есть яблоко.",
4     "answer": "I have an apple",
5     "additional": "has a she"
6   },
7   {
8     "Переведите предложение": "У нее есть молоко.",
9     "answer": "She has milk",
10    "additional": "have a he"
11  },
12  {
13    "Переведите предложение": "Я читаю книгу.",
14    "answer": "I read a book",
15    "additional": "have an hee"
16  },
17  {
18    "Переведите предложение": "Мой папа смотрит телевизор.",
19    "answer": "My father watch TV",
20    "additional": "mother read"
21  }
22 ]

```

Рисунок Е.6 – Получение подзаданий для текстового задания легкого уровня с номером 1

GET 192.168.0.106/DbScript.php/?action=GET_EASY_TXT_TASK&user_id=236568364&level_type=text&level=2 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION
☒	action	GET_EASY_TXT_TASK	
☒	user_id	236568364	
☒	level_type	text	
☒	level	2	
	Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 9 ms Size: 1.58 KB

Pretty Raw Preview Visualize JSON

```

1  [
2    {
3      "Переведите предложение": "I read a book.",
4      "answer": "Я читаю книгу",
5      "additional": "смотрю телевизор"
6    },
7    {
8      "Переведите предложение": "I have an apple.",
9      "answer": "У меня есть яблоко",
10     "additional": "него молоко"
11   },
12   {
13     "Переведите предложение": "She has milk.",
14     "answer": "У нее есть молоко",
15     "additional": "него яблоко"
16   },
17   {
18     "Переведите предложение": "My father watch TV.",
19     "answer": "Мой папа смотрит телевизор",
20     "additional": "мама читает Моя книгу"
21   }
22 ]

```

Рисунок Е.7 – Получение подзаданий для текстового задания легкого уровня с номером 2

GET
192.168.0.106/DbScript.php?action=GET_EASY_IMG_TASK&user_id=236568364&level_type=text&level=1
Send

	KEY	VALUE	DESCRIPTION
☒	action	GET_EASY_IMG_TASK	
☒	user_id	236568364	
☒	level_type	text	
☒	level	1	
	Key	Value	Description

Body
Cookies
Headers (6)
Test Results
Status: 200 OK
Time: 7 ms
Size: 1.7 KB

Pretty
Raw
Preview
Visualize
JSON

```

1  [
2    {
3      "task": "Выберите картинку, на которой изображено слово ",
4      "answerWord": "Apple",
5      "answerImg": "/images/apple.png",
6      "additionalImg": "/images/melon.png /images/milk.png /images/dog.png"
7    },
8    {
9      "task": "Выберите картинку, на которой изображено слово ",
10     "answerWord": "Milk",
11     "answerImg": "/images/milk.png",
12     "additionalImg": "/images/melon.png /images/girl.png /images/orange.png"
13   },
14   {
15     "task": "Выберите картинку, на которой изображено слово ",
16     "answerWord": "Melon",
17     "answerImg": "/images/melon.png",
18     "additionalImg": "/images/orange.png /images/watermelon.png /images/dog.png"
19   },
20   {
21     "task": "Выберите картинку, на которой изображено слово ",
22     "answerWord": "Orange",
23     "answerImg": "/images/orange.png",
24     "additionalImg": "/images/watermelon.png /images/bird.png /images/dog.png"
25   }
26 ]

```

Рисунок Е.8 – Получение подзаданий для задания с картинками легкого уровня с номером 1

GET 192.168.0.106/DbScript.php/?action=GET_EASY_MEDIA_TASK&user_id=236568364&level_type=text&level=1 **Send**

KEY	VALUE	DESCRIPTION
☒ action	GET_EASY_MEDIA_TASK	
☒ user_id	236568364	
☒ level_type	text	
☒ level	1	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 8 ms Size: 914 B

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "task": "Что сказал диктор?",
4     "type": "audio",
5     "answer": "Hello my friend",
6     "additional": "Holla me friends fire"
7   },
8   {
9     "task": "Что сказал диктор?",
10    "type": "audio",
11    "answer": "What is your name",
12    "additional": "Where are our age"
13  },
14  {
15    "task": "Что сказал диктор?",
16    "type": "audio",
17    "answer": "Boy likes bananas",
18    "additional": "like Boys ban"
19  },
20  {
21    "task": "Что сказал диктор?",
22    "type": "audio",
23    "answer": "I love my parents",
24    "additional": "live me apple"
25  }
26 ]

```

Рисунок Е.9 – Получение подзаданий для задания с медиа легкого уровня с номером 1

POST 192.168.0.106/DbScript.php/ **Send**

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE	DESCRIPTION
☒ action	UPD_LAST_DIFF	
☒ user_id	236568364	
☒ last_difficult	Средний	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 17 ms Size: 196 B

Pretty Raw Preview Visualize JSON

```

1 success

```

Рисунок Е.10 – Обновление последнего выбранного уровня сложности

ПРИЛОЖЕНИЕ Ж

Результаты тестирования мобильного приложения
(рекомендуемое)

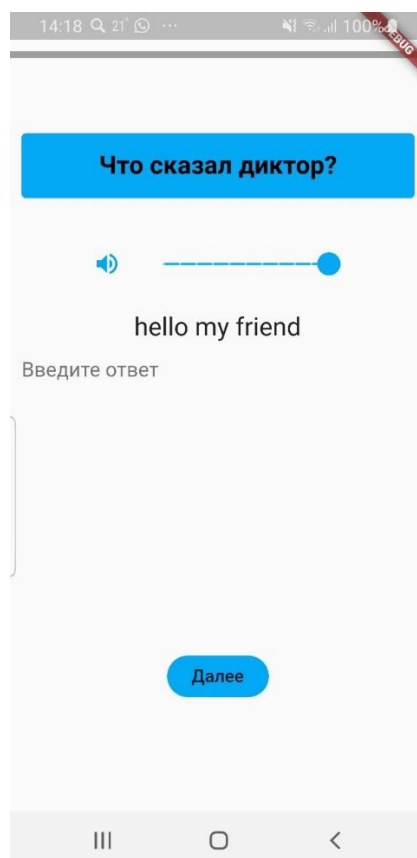


Рисунок Ж.1 – Результат тестирования с пустым значением

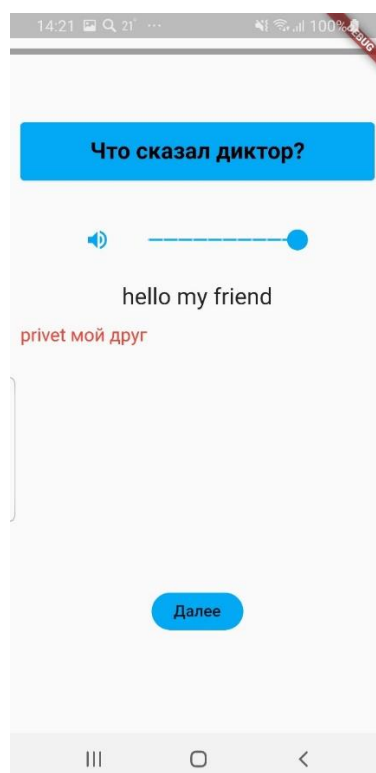


Рисунок Ж.2 – Результат тестирования с русскими символами

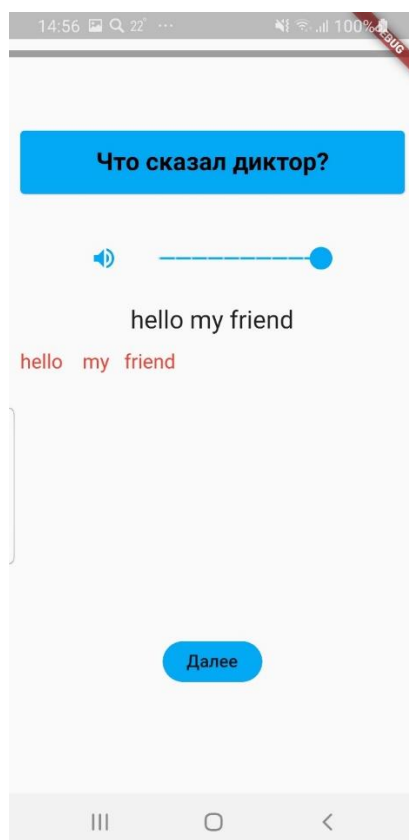


Рисунок Ж.3 – Результат тестирования с несколькими пробелами

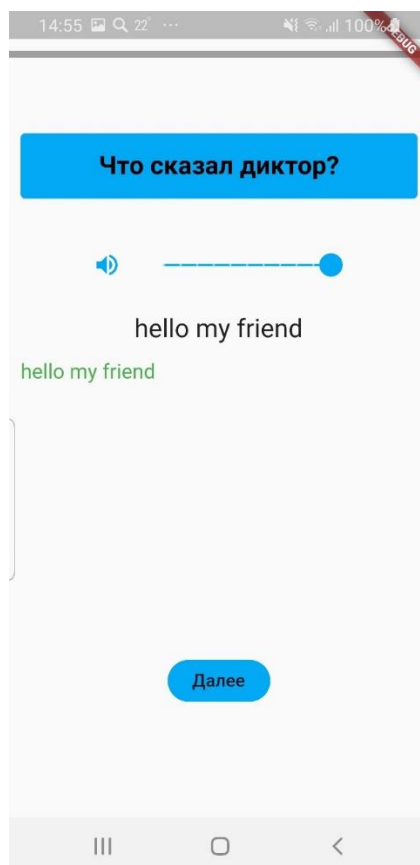


Рисунок Ж.4 – Результат тестирования с корректным значением

ПРИЛОЖЕНИЕ 3

Граф функции
(рекомендуемое)

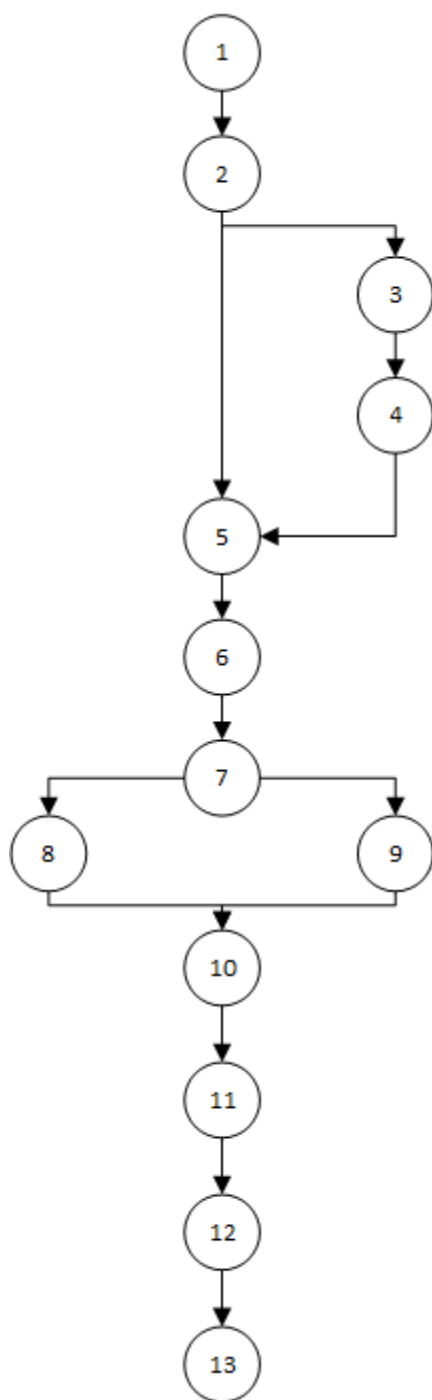


Рисунок 3.1 – Граф функции