

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 управления
спортивными мероприятиями

Студент	_____	Васин Егор Андреевич
	(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
«__» _____ 2019 г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

1. Студент Васин Егор Андреевич гр. 16ВП1 факультета ВТ
<Фамилия, имя, отчество полностью>

2. Тема работы Мобильное приложение для управления спортивными мероприятиями

Тема утверждена приказом ПГУ № 1488/о от "30" декабря 2020

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 мобильное приложение для управления спортивными мероприятиями, позволяющее заполнять информацию о турнире, формировать протоколы спортивных матчей, просматривать информацию о турнире

2) Среда выполнения: Операционная система Android 7.0 и выше для клиентской части, операционная система Windows для серверной части

Языковые средства разработки: C#, SQL, UML

Инструментальные средства разработки: Microsoft Visual Studio 2019, MS SQL Server, смартфон Samsung Galaxy A5, Postman, Microsoft Project, Microsoft Visio

Технологии разработки: RUP, ООП, Xamarin

3) Основные функции мобильного приложения:

а) создание турниров;

б) создание команд;

в) создание игроков;

г) протоколирование матчей;

д) отображение информации о турнире.

5. Перечень подлежащих разработке вопросов _____

1) Анализ предметной области и постановка задач _____*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 и 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 149 листов, 53 рисунка, 13 таблиц, 26 использованных источников и 5 приложений.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, СПОРТИВНЫЕ МЕРОПРИЯТИЯ, ОРГАНИЗАЦИЯ ТУРНИРОВ, XAMARIN, MICROSOFT VISUAL STUDIO, MS SQL SERVER, POSTMAN, MICROSOFT PROJECT, MICROSOFT VISIO

Объектом разработки является мобильное приложение для управления спортивными мероприятиями.

Цель выпускной квалификационной работы – разработать мобильное приложения для управления спортивными мероприятиями.

В результате выполненной работы создано мобильное приложение для управления спортивными мероприятиями, позволяющее обеспечивать удобное взаимодействие между организаторским штатом спортивных мероприятий и участниками турниров.

Разработка проводилась на языке программирования C# в среде программирования Microsoft Visual Studio 2019. Среда исполнения мобильного приложения Android 7.0 и выше. В качестве СУБД была выбрана Microsoft SQL Server 2016. Тестирование серверной части приложения проводилось с помощью инструментального средства Postman.

					ПГУ 09.03.04– 8BP161.07 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Мобильное приложение для управления спортивными мероприятиями. Пояснительная записка	Лит.	Лист	Листов
Разраб.		Васин Е.А..					4	149
Пров.		Князев В.Н.				Группа 16ВП1		
Н. Контр.		Такташкин Д.В						
Утв.								

Содержание

Введение.....	7
1 Анализ предметной области и требований.....	9
1.1 Анализ предметной области и постановка задач.....	9
1.2 Анализ аналогичных решений.....	11
1.2.1 Анализ мобильного приложения «Winner-Создание турниров»	11
1.2.2 Анализ мобильного приложения «League App»	12
1.2.3 Анализ мобильного приложения «Международный симулятор».....	13
1.3 Анализ целевой аудитории.....	14
1.4 Анализ функциональных требований	16
1.5 Анализ нефункциональных требований	26
1.6 Планирование разработки и оценка бюджета	27
2 Проектирование мобильного приложения	37
2.1 Выбор технологий и инструментальных средств разработки	37
2.2 Проектирование архитектуры программного обеспечения.....	38
2.3 Проектирование пользовательского интерфейса.....	39
2.4 Проектирование серверной части приложения	51
2.4.1 Логическое проектирование базы данных.....	51
2.5 Проектирование API	58
3 Реализация мобильного приложения	60
3.1 Реализация серверной части	60
3.1.1 Реализация базы данных.....	60
3.1.2 Реализация серверной части приложения	62
3.2 Реализация клиентской части мобильного приложения.....	65
4 Тестирование программного продукта	68
4.1 Тестирование серверной части приложения	68
4.2 Тестирование клиентской части приложения	76
4.2.1 Функциональное тестирование.....	76
4.3 Метрики кода.....	78
4.3.1 SLOC-метрика	79

4.3.2 Метрики Холстеда.....	80
Заключение	83
Список использованных источников	84
Приложение А. Глоссарий	88
Приложение Б. Исходный код серверной части	90
Приложение В. Исходный код мобильного приложения	120
Приложение Г. Результаты тестирования серверной части	132
Приложение Д. Результаты тестирования мобильного приложения	136

Введение

В последнее время информационные технологии внедряются во все сферы человеческой жизни. Все страны мира уделяют этому сектору огромное внимание, Россия так же не является исключением. Об этом может свидетельствовать утверждение в 2017 году программы «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» [1]. За последнее десятилетие резко увеличилось количество владельцев смартфонов, что привело к появлению различных мобильных приложений, интернет-сервисов. «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 [2].

В настоящее время спортивные мероприятия становятся все более частым явлением в жизни людей. Их проводят как на профессиональном уровне, так и на любительском. Профессиональные спортивные команды, школы, университеты, предприятия соревнуются и выявляют сильнейших в различных спортивных турнирах. Как правило, когда проводится какой-либо спортивный турнир, например, между школами, то информация о турнире, его календарь, состав участников хранится у ответственного преподавателя физической культуры в бумажном виде, протоколы матчей заполняются судьями или секретарями игр так же в аналогичном виде, а иногда и вовсе игнорируются. Школьники, принимающие участие в турнире, могут получить информацию о положении своей команды в турнирной таблице, узнать как сыграли конкуренты из другой школы или узнать с кем им осталось провести

матчи в турнире только из уст преподавателя, но не всегда это является возможным и удобным. Все вышеописанные процессы можно представить в другой, более удобной форме в настоящее время – цифровой, а устройством представления должна выступать популярная вещь, имеющаяся в настоящее время у любого человека – на эту роль отлично подходит мобильный смартфон [3 - 6].

При наличии мобильного приложения, позволяющего представлять информацию и данные о спортивных турнирах в цифровом виде, документировать протоколы матчей, а так же отображать всю эту информацию в удобном для пользователя виде, можно упростить процесс проведения спортивных мероприятий, сделать его более удобным как для организаторов турниров, так и для самих спортсменов. В одном приложении будет производиться и протоколирование матчей и отображение информации о турнирах – турнирные таблицы, календари матчей, результаты сыгранных игр, списки лучших снайперов и ассистентов. Все участники турниров будут осведомлены о предстоящих играх, будут знать результаты всех сыгранных матчей, будут иметь представление о том, какие из игроков турнира являются наиболее «опасными» – списки лучших снайперов, ассистентов и бомбардиров будут давать ответы на этот вопрос.

Таким образом, подобные системы способны облегчить работу организаторов турнира, упростить процесс получения информации о турнире для участников соревнований.

Исходя из вышесказанного, целью выпускной квалификационной работы является разработка мобильного приложения, позволяющего управлять спортивными мероприятиями. Разработка такого приложения является актуальной и перспективной задачей в настоящее время.

1 Анализ предметной области и требований

1.1 Анализ предметной области и постановка задач

В настоящее время спортивные мероприятия проводятся среди профессиональных спортсменов, обычных школьников, студентов, работников различных предприятий, фирм, организаций. Они позволяют выявить сильнейших среди соревнующихся спортсменов, пропагандируют здоровый образ жизни. Проведение таких мероприятий формирует у человека понятие сущности о здоровом образе жизни, способствует развитию коммуникабельности и активной жизненной позиции.

Всего в 2013 – 2020 гг. на развитие спортивной индустрии в рамках государственной программы «Развитие физической культуры и спорта» выделено 1 717 млрд рублей, на рисунке 1 более подробно представлено распределение денежных средств, выделенных на развитие спортивной индустрии в России. Значительная часть этих средств идет на развитие спортивной инфраструктуры. Объем производства отечественной продукции спортивной индустрии в России вырос на 6,6 % в 2017 году по отношению к 2016 году и составил 27,32 млрд. рублей [7].



Рисунок 1 – Распределение денежных средств в спортивной индустрии России

Данная статистика подчеркивает, что спортивный сектор в Российской Федерации растет и активно развивается, в него вкладываются немалые

денежные средства. Это говорит о том, что разработка приложений, предметной областью которых является спортивная отрасль является актуальной и перспективной на сегодняшний день [8].

При проведении спортивных мероприятий судья или секретарь матча заполняет протокол, в котором отображает авторов забитых мячей, шайб, тех, кто им ассистировал, т.е. отдал голевую передачу и штрафников – спортсменов, получающих наказание за совершенное нарушение, например, желтую или красную карточку. При проведении многих соревнований протоколы ведутся в бумажном виде, а так же вся информация о турнире – турнирная таблица, календарь игр, результаты матчей и т.п. хранится так же в бумажном виде у организаторов турнира. Для того, чтобы узнать дату и время следующей игры, турнирное положение команды – спортсменам команды приходится получать эту информацию от ответственных лиц: судей, организаторов турнира [9]. Процесс протоколирования и процесс получения информации о турнире может быть упрощен путем внедрения в область спортивных мероприятий информационных технологий. Необходимо создать такое приложение, которое позволит держать в курсе событий турнира всех его участников. Приложение должно быть установлено на общедоступном устройстве – мобильном смартфоне.

Вся информация о турнире должна быть представлена в виде компьютерных данных, хранящихся на сервере. Приложение, установленное на мобильном устройстве должно иметь Интернет-соединение, для того чтобы обращаться к серверу, а тот в свою очередь предоставлять данные о турнире.

Одно приложение позволит упростить работу судей матча путем отказа от оформления протоколов в бумажном формате и их заполнением в электронном виде, а так же позволит пользователям приложений – спортсменам или обычным людям, заинтересованным положением дел в каком-либо турнире получать о нем информацию в легкой, удобной, доступной форме.

Таким образом, необходимо разработать мобильное приложение для управления спортивными мероприятиями, позволяющее представлять информацию о турнире в виде компьютерных данных, документировать в электронном формате протоколы матчей, а так же предоставлять информацию о турнире в виде таблиц и списков лучших игроков турнира. Данное мобильное приложение будет являться многопользовательским, будут существовать учетные записи организаторов турниров, заполняющих всю информацию о турнире в базу данных, и секретарей матчей – основной функцией которых будет являться оформление протокола игры. Для обычных пользователей, основной возможностью которых будет являться получение информации наличие учетной записи не требуется.

1.2 Анализ аналогичных решений

В настоящее время существуют аналогичные приложения, позволяющие управлять спортивными мероприятиями. Проведем сравнительный анализ наиболее известных программных продуктов данного сектора.

1.2.1 Анализ мобильного приложения «Winner-Создание турниров»

«Winner-Создание турниров» – приложение для организации различных турниров. Данное приложение предоставляет возможность по созданию турниров в различных спортивных дисциплинах: футбол, хоккей, волейбол, теннис, баскетбол и др. [10]. Главный экран мобильного приложения представлен на рисунке 2.

Среди достоинств данного приложения можно выделить то, что приложение поддерживает большое число создаваемых форматов турниров – групповой турнир, плей-офф, групповой турнир совместно с плей-офф. Так же достоинством можно назвать интуитивно понятный интерфейс приложения, что позволяет пользователю с первых минут пользования грамотно пользоваться программой.

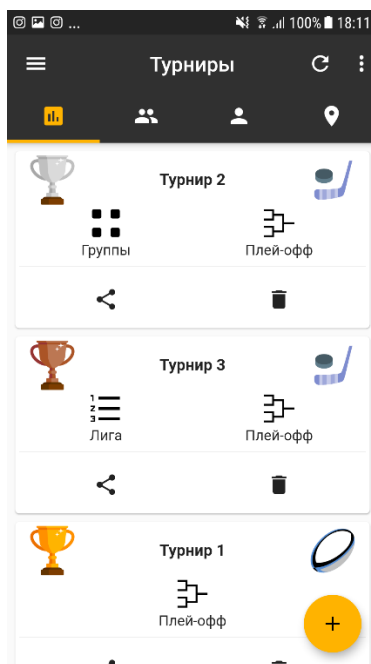


Рисунок 2 – Главный экран мобильного приложения «Winner – Создание турниров»

Однако, помимо преимуществ, данное приложение имеет и ряд недостатков, которые были выделены в ходе использования данного программного продукта. Недостатком приложения является то, что при добавлении игрока в команду можно указать только его имя и фамилию, отсутствует возможность указания других данных об игроке: возраст, позиция на поле и т.п. Другой слабой стороной является отсутствие возможности отмечать составы команд на матчи, указывать основные события игры – забитые мячи, удаления игроков. Из-за отсутствия возможности вести протокол матча, приложение не обладает никакими возможностями просмотра статистики игроков в турнире.

1.2.2 Анализ мобильного приложения «League App»

«League App» – приложение, позволяющее организовывать спортивные мероприятия в мобильном устройстве. Приложение поддерживает возможность создания турниров различных форматов: плей-офф, круговую систему, смешанную систему [11]. Существует возможность назначать дату и время для каждой игры. Главный экран приложения представлен на рисунке 3.

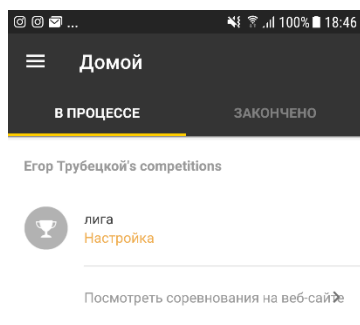


Рисунок 3 – Главный экран мобильного приложения «League App»

Отличительной особенностью и главным преимуществом приложения являются уведомления в режиме реального времени о событиях, происходящих в матче (голы, удаления, замены и т.п.). Также большим плюсом можно назвать обновление турнирных таблиц в режиме реального времени.

Большим недостатком данного приложения является ограниченные функциональные возможности в бесплатной версии приложения. При отсутствии ПРО-версии существует возможность заполнения протоколов только 10 матчей в турнире. Все виды статистики турнира – лучшие бомбардиры, снайперы, ассистенты доступны только в платной версии приложения.

1.2.3 Анализ мобильного приложения «Международный симулятор»

«Международный симулятор» – приложение, поддерживающее возможность создания групповых футбольных турниров, создание футбольных лиг с автоматическим повышением и понижением команд в классах [12]. Главный экран приложения представлен на рисунке 4.

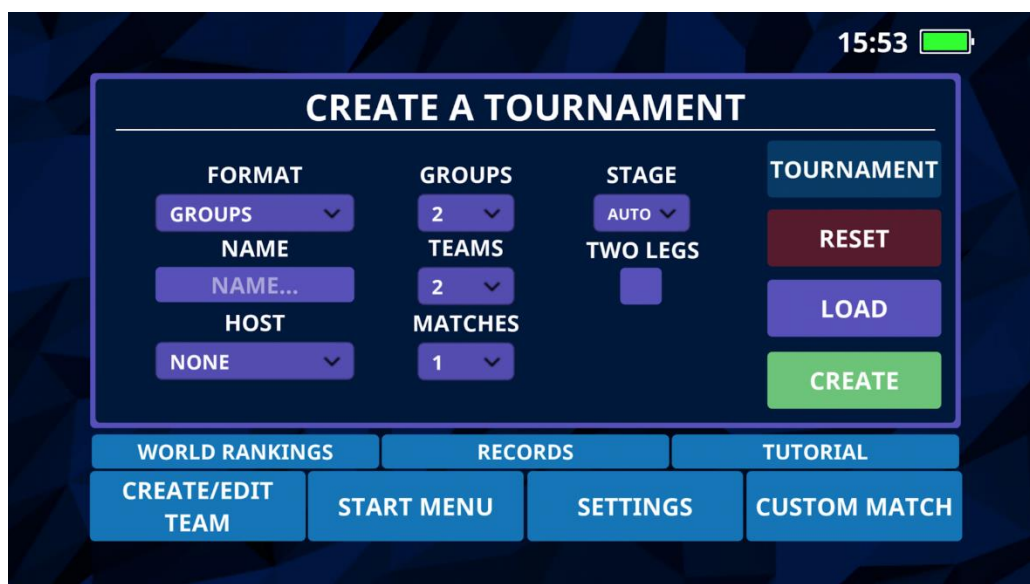


Рисунок 4 – Главный экран мобильного приложения «Международный симулятор»

Среди преимуществ приложения можно выделить поддержку статистики для каждого матча, существует возможность узнать авторов забитых голов игры, их ассистентов, игроков, получивших травмы, желтые карточки.

К недостаткам приложения можно отнести предназначенность приложения для одной дисциплины – футбол, а так же отсутствие поддержки русского языка.

Таким образом, в результате анализа программ-аналогов, было принято решение о создании собственного мобильного приложения для управления спортивными мероприятиями, которое не должно содержать выделенных у данных продуктов недостатков.

1.3 Анализ целевой аудитории

Целевой аудиторией для разрабатываемого приложения является организационный корпус лиц, отвечающих за проведение соревнований – организаторы и секретари соревнований, а так же сами спортсмены, принимающие участие в соревнованиях.

На рисунке 5 представлены доли производителей смартфонов на мировом рынке. Лидеры этого списка – Samsung, Huawei, Xiaomi являются

мобильными устройствами, разработанными на базе ОС Android. Из этого можно сделать вывод, что большая часть производителей выпускает смартфоны с ОС Android. Для российского рынка данная закономерность так же не является исключением. Согласно данным на 2019 г. смартфоны с ОС Android занимают 73 % от общего российского рынка мобильных устройств, доля другой популярной ОС iOS составляет всего 26 %, еще 1 % приходится на другие ОС [13, 14].

[Global Smartphone Share(%)]

Global Smartphone Shipments Market Share (%)	2018 Q2	2018 Q3	2018 Q4	2019 Q1	2019 Q2
Samsung	20%	19%	18%	21%	22%
Huawei#	15%	14%	15%	17%	16%
Apple	11%	12%	17%	12%	10%
Xiaomi	9%	9%	6%	8%	9%
Oppo	8%	9%	8%	8%	9%
vivo	7%	8%	7%	7%	8%
Lenovo**	2%	3%	3%	3%	3%
Others	31%	28%	26%	24%	23%

Рисунок 5 – Доли производителей смартфонов на мировом рынке

На рисунке 6 представлена статистика по распределению версий ОС Android среди пользователей по состоянию на апрель 2020 года, она показывает, что наиболее распространенными являются версии 7.0 «Nougat», 8.0 «Oreo», 9.0 «Pie». Вместе они составляют более 65 % среди всех версий ОС Android [15].

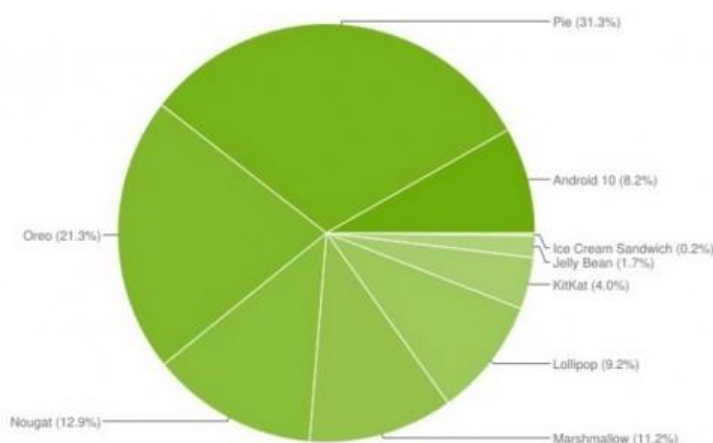


Рисунок 6 – Распределение версий ОС Android

На рисунке 7 представлены средние цены на смартфоны ОС Android и iOS в течение 2010 – 2013 гг. Данная статистика демонстрирует то, что смартфон с ОС Android стоит гораздо дешевле смартфона с ОС iOS. В настоящее время ситуация кардинально не поменялась. Android-смартфоны по-прежнему являются более выигрышными по сравнению с iOS-смартфонами в финансовом вопросе [16, 17].

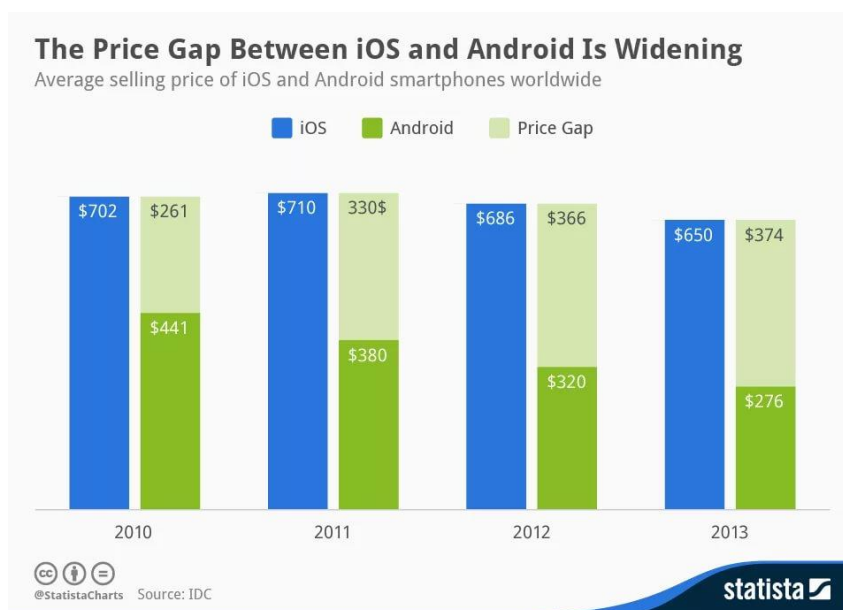


Рисунок 7 – Средние цены на смартфоны ОС Android и iOS

Таким образом, можно сделать, что разрабатываемое приложение должно быть реализовано на ОС Android и должно функционировать на платформе 7.0 «Nougat» и более поздних версий.

1.4 Анализ функциональных требований

Разрабатываемое приложение предназначено для следующих категорий пользователей: организатор турнира, секретарь матча и простой пользователь. Основной задачей организатора турнира будет являться внесение всей информации о турнире в приложение – название турнира, состав участников, календарь турнира и т.п. Секретарь матча оформляет протокол встречи. Простому пользователю доступна возможность получения различной информации о турнире.

В результате анализа предметной области и аналогичных программных продуктов были выделены функциональные требования к программе [18]. Для организатора турниров были выделены следующие функции:

- зарегистрироваться в приложении;
- добавить турнир;
- редактировать информацию о турнире;
- удалить турнир;
- добавить команду в турнир;
- удалить команду из турнира;
- добавить игрока в команду;
- редактировать информацию об игроке;
- удалить игрока из команды;
- сгенерировать календарь турнира;
- редактировать календарь турнира.

Для секретаря матча были выделены следующие требования:

- зарегистрироваться в приложении;
- заполнить состав команд на матч;
- заполнить протокол матча.

Для обычного пользователя приложения, для которого не требуется специальной учетной записи, были определены следующие функциональные возможности:

- просмотреть таблицу турнира;
- просмотреть календарь турнира;
- просмотреть результаты матчей турнира;
- просмотреть информацию о команде;
- просмотреть протокол матча;
- просмотреть списки лучших игроков турнира.

Диаграмма вариантов использования для организатора турнира и секретаря матча представлена на рисунках 8 и 9.



Рисунок 8 – Диаграмма вариантов использования для организатора турниров и секретаря матча

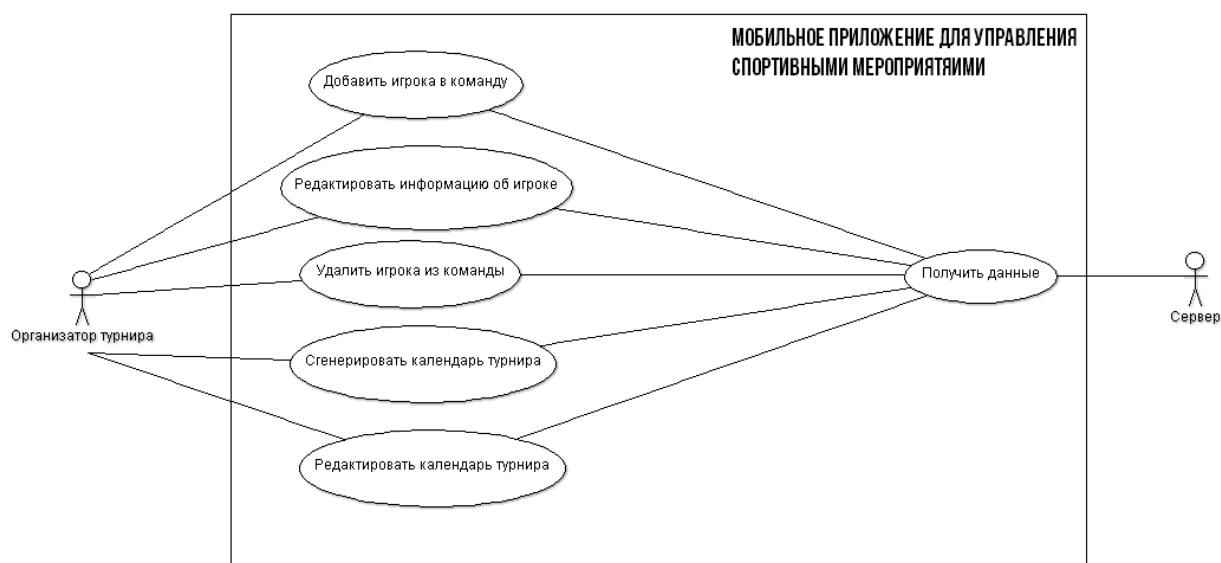


Рисунок 9 – Диаграмма вариантов использования для организатора турниров

Диаграмма вариантов использования для пользователя приложения представлена на рисунке 10.

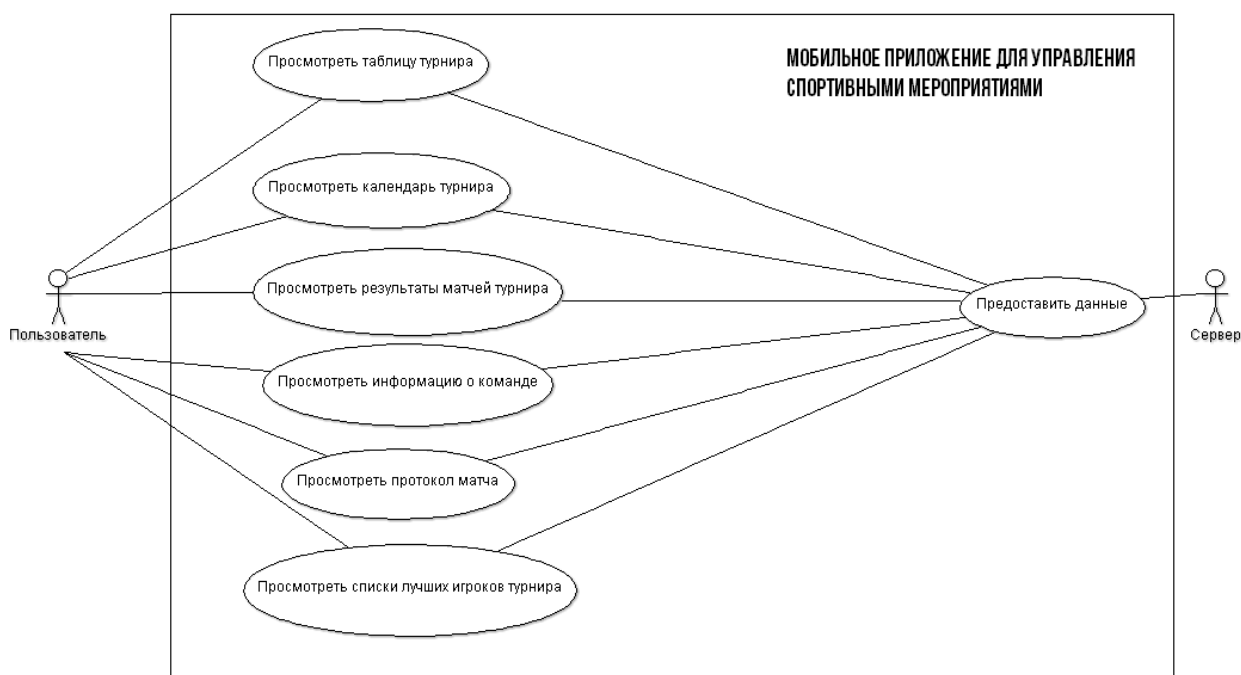


Рисунок 10 – Диаграмма вариантов использования для пользователя

Рассмотрим некоторые прецеденты более подробно, для этого приведем их подробную спецификацию.

На рисунке 11 описана спецификация для варианта использования «Добавить игрока в команду», который предназначен для организатора турнира [19]. Если провести сравнение с программами-аналогами, то стоит отметить, что данный вариант использования используется в некоторых приложениях, но является довольно-таки слаборазвитым в плане информационной составляющей об игроке – отсутствие возможности добавления амплуа игрока, игрового номера. Так как в рамках разработки мобильного приложения было принято решение ликвидировать недостатки приложений-аналогов, то данную функциональность необходимо расширить за счет добавления дополнительной информации об игроке. Эти данные в дальнейшем будут использоваться секретарем матча для удобства заполнения протокола матча.

Препедент: Добавить игрока в команду
ID: 1
Краткое описание: Организатор турнира заполняет информацию об игроке и добавляет его в команду
Главные актеры: Организатор турнира
Второстепенные актеры: Сервер
Предусловия: Организатор турнира авторизовался в приложении
Основной поток: <ol style="list-style-type: none"> 1. Организатор турнира вводит в поля ввода имя и фамилию игрока, указывает дату его рождения 2. Организатор турнира выбирает турнир, в который необходимо добавить игрока 3. Выполняется запрос на сервер 4. Сервер возвращает список команд, участвующих в выбранном турнире 5. Организатор турнира выбирает команду 6. Организатор турнира выбирает амплуа игрока 7. Организатор турнира указывает номер игрока из перечня доступных 8. Организатор турнира уведомляет приложение о том, что он собирается добавить игрока в команду
Постусловия: Организатор турнира переходит на главную страницу приложения
Альтернативные потоки: Нет

Рисунок 11 – Спецификация варианта использования «Добавить игрока в команду»

На рисунке 12 описана спецификация для варианта использования «Заполнить составы команд на матч», который предназначен для секретаря матча [19]. Стоит отметить, что лишь в одном приложении-аналоге есть данная функция, однако, она доступна в ограниченном варианте и ее возможно использовать только для 10 матчей турнира. Использование данного варианта использования позволит отмечать действия игроков (голы, голевые передачи, удаления), принимающих участие в матче. В дальнейшем всю хронологию матча может посмотреть пользователь, используя вариант использования «Просмотреть протокол матча», там будут отображены действия игроков в

матче. Так же отмечая голы и голевые передачи игроков, пополняется статистика игроков в турнире – списки лучших снайперов, ассистентов и бомбардиров турнира.

Прецедент: Заполнить составы команд на матч
ID: 2
Краткое описание: Секретарь матча указывает составы команд на игру
Главные актеры: Секретарь матча
Второстепенные актеры: Сервер
Предусловия: Секретарь матча авторизовался в приложении
Основной поток: <ol style="list-style-type: none"> 1. Секретарь матча выбирает необходимую игру турнира из перечня матчей 2. Выполняется запрос на сервер 3. Сервер возвращает составы команд, заявленные командой на турнир 4. Секретарь матча указывает игроков 5. Секретарь матча уведомляет приложение о том, что он собирается сохранить внесенные составы команд на матч
Постусловия: Нет
Альтернативные потоки: Нет

Рисунок 12 – Спецификация варианта использования «Заполнить составы команд на матч»

На рисунке 13 описана спецификация для варианта использования «Просмотреть протокол матча», который предназначен для пользователя [19]. Используя данную возможность, пользователь может узнать итоговый счет встречи, составы команд на матч, просмотреть хронологию событий матча – по минутам выводятся действия игроков в матче. Стоит отметить, что данная функциональная возможность реализована лишь в одном из рассмотренных приложений-аналогов, но ее доступность проявляется только при действии платной версии.

Прецедент: Просмотреть протокол матча
ID: 3
Краткое описание: Пользователь просматривает информацию сыгранного матча турнира
Главные актеры: Пользователь
Второстепенные актеры: Сервер
Предусловия: Нет
Основной поток: <ol style="list-style-type: none"> 1. Пользователь выбирает необходимый матч из перечня состоявшихся игр 2. Выполняется запрос на сервер 3. Сервер возвращает информацию о матче 4. Пользователь просматривает информацию о матче
Постусловия: Нет
Альтернативные потоки: Нет

Рисунок 13 – Спецификация варианта использования «Просмотреть протокол матча»

Для некоторых вариантов использования составим диаграммы деятельности. На рисунке 14 приведена диаграмма деятельности для прецедента «Добавить команду в турнир». Данный вариант использования применяется организатором турнира. После открытия организатором страницы, осуществляющей добавление команды в турнир, приложение должно отобразить лишь те спортивные мероприятия, в которые возможно добавление команды. Это связано с тем, что некоторые турниры уже ведутся и в них проводятся матчи между командами, а в некоторых соревнованиях состав участников сформирован, но первая игра пока что не состоялась. Для выполнения этого требования мобильное приложение формирует и отправляет HTTP-запрос серверной части, возвращающий те турниры, в которые возможно добавить команду, серверная часть обращается к базе данных и возвращает результаты поиска, после этого мобильное приложение отображает доступные турниры. Далее организатор турнира должен выбрать какой-либо турнир. В целях удобства организации работы пользователя,

каждый раз при выборе определенного турнира, загружается дополнительное текстовое поле, отображающее название дисциплины, по которой выбран турнир. Схема получения названия спортивной дисциплины заключается в отправке запроса на сервер, который должен вернуть в качестве ответа название вида спорта, по которому выбран турнир. Пользователь указывает название команды, которую необходимо добавить в турнир, и уведомляет приложение о том, что хочет подтвердить ее добавление, мобильное приложение отправляет запрос на добавление команды в турнир на сервер, последний обращается к базе данных и добавляет в нее данные.

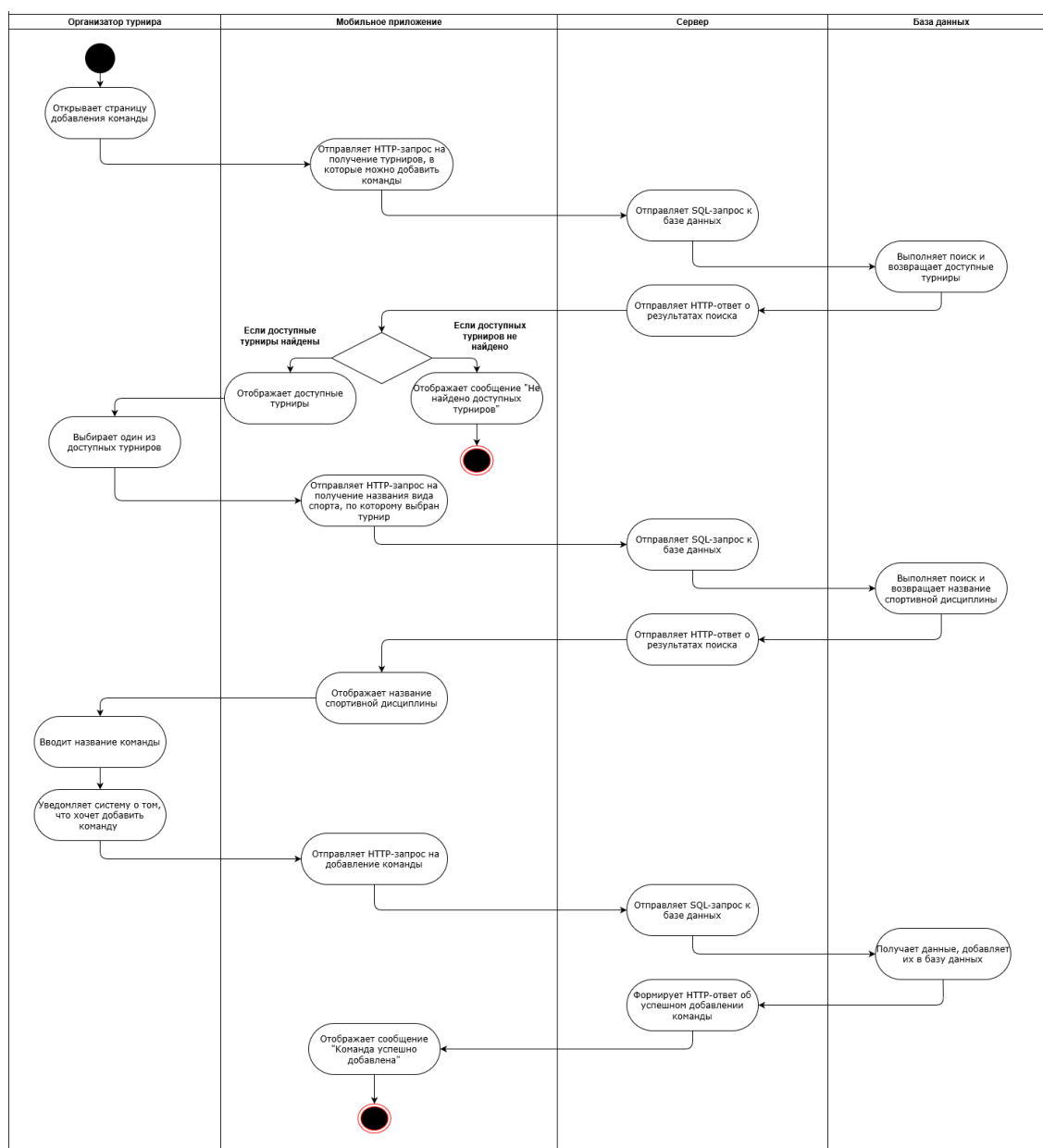


Рисунок 14 – Диаграмма деятельности для варианта использования «Добавить команду в турнир»

На рисунке 15 приведена диаграмма деятельности для прецедента «Заполнить составы команд на матч». Данный вариант использования применяется секретарем матча. После того, как секретарь матча открывает вкладку заполнения составов команд на матч, мобильное приложение формирует HTTP-запрос к серверной части мобильного приложения с целью получения всех игроков, заявленных за 2 команды, принимающих участие во встрече. Сервер после приема запроса, обращается к базе данных, получает необходимую информацию и формирует HTTP-ответ, предназначенный для отправки на клиентскую часть мобильного приложения. Далее секретарю матча необходимо определить составы играющих команд, при помощи элемента пользовательского интерфейса «флажок», установленного напротив каждого игрока команды, секретарь выбирает таких игроков, определив у каждого из этих элементов состояние «включено». Для каждого вида спорта необходимо выбрать минимальное допустимое количество игроков у команды, для того чтобы можно было сохранить указанные составы. Например, для футбола – одиннадцать человек, для хоккея и регби - по пятнадцать человек. Секретарь подтверждает, что хочет сохранить указанные составы, мобильное приложение проверяет количество выбранных секретарем игроков на матч, если количество игроков не равно минимальному возможному числу, то мобильное приложение отображает соответствующее сообщение, в противном случае мобильное приложение отправляет запрос на сервер на добавление этой информации в базу данных. Серверная часть принимает данные, обращается к базе данных и добавляет их в нее.

Стоит отметить, что данная функциональная возможность реализована лишь в одном из рассмотренных приложений-аналогов, но ее доступность проявляется только при действии платной версии.

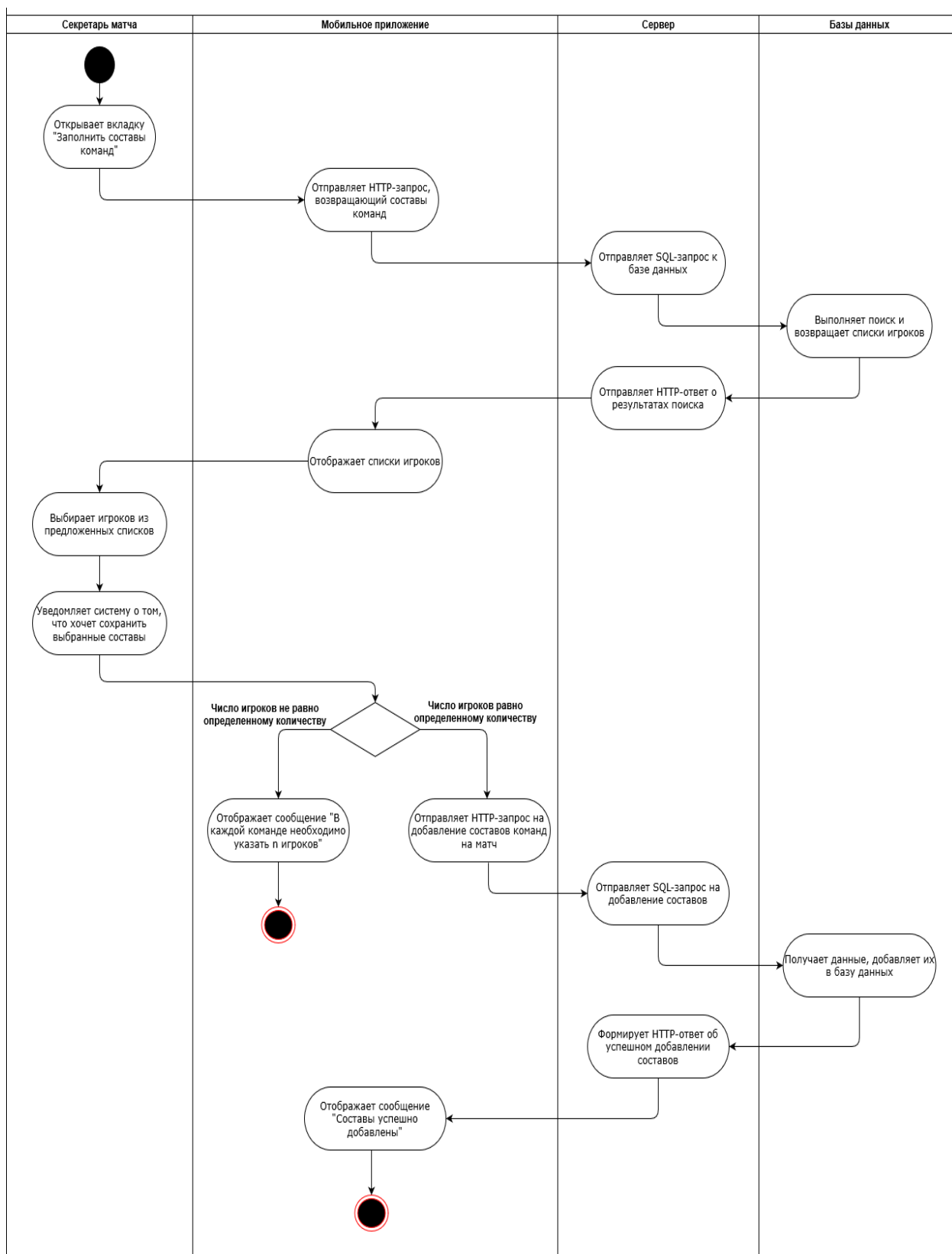


Рисунок 15 – Диаграмма деятельности для варианта использования «Заполнить составы команд на матч»

На рисунке 16 представлена диаграмма последовательности для варианта использования «Добавить турнир».



Рисунок 16 – Диаграмма последовательности для варианта использования «Добавить турнир»

В результате анализа функциональных требований, были уточнены и определены функции, которые необходимо реализовать в разрабатываемой программе, были разработаны диаграммы вариантов использования, диаграммы деятельности и диаграмма последовательности.

1.5 Анализ нефункциональных требований

Разработка серверной части приложения должна быть выполнена на языке программирования C#, в среде программирования Microsoft Visual Studio 2019. Серверная часть должна включать в себя API (интерфейс прикладного программирования). Клиентская часть приложения должна быть реализована на языке программирования C#, при помощи технологии Xamarin.Forms, в качестве целевой СУБД должна выступать Microsoft SQL Server, запросы к базе данных должны осуществляться при помощи языка SQL.

Клиентская часть приложения должна иметь удобный и интуитивно понятный интерфейс, клиентская часть должна быть полностью отлажена на предмет возникновения исключительных ситуаций. Мобильное приложение

должно исполняться на мобильных устройствах под управлением ОС Android 7.0 и выше.

1.6 Планирование разработки и оценка бюджета

Под планированием понимается определение ясных и точных задач, служащих для достижения поставленных задач. Ресурсы и усилия могут быть потрачены впустую, если не осуществить надлежащего планирования программного проекта. Оценка бюджета – это распределение статей доходов и расходов [20].

В таблице 1 представлен перечень задач, представляющих из себя этапы создания разрабатываемого проекта.

Таблица 1 – Перечень задач по созданию проекта

№	Название задачи	Длительность	Предшественники
1	Начало проекта	-	-
2	Анализ требований для проекта разработки мобильного приложения для управления спортивными мероприятиями	-	-
3	Обзор приложений-аналогов	5 дней	1
4	Составление списка функциональных возможностей приложения	4 дня	1
5	Выбор программного средства для реализации приложения	4 дня	3;4
6	Анализ требований завершен	-	5
7	Проектирование для проекта разработки мобильного приложения для управления спортивными мероприятиями	-	-
8	Проектирование интерфейса приложения	3 дня	6
9	Проектирование структур данных	5 дней	6
10	Выбор архитектуры программного обеспечения	4 дня	9

Продолжение таблицы 1

№	Название задачи	Длительность	Предшественники
11	Проектирование завершено	-	8;10
12	Реализация	-	-
13	Программирование классов приложения	8 дней	11
14	Создание базы данных	5 дней	11
15	Интеграция клиентской и серверной частей приложения	4 дня	13;14
16	Реализация завершена	-	15
17	Отладка	-	-
18	Тестирование работы приложения	8 дней	16
19	Исправление ошибок	7 дней	18
20	Отладка завершена	-	19
21	Конец проекта	-	20

Из таблицы видно, что для реализации проекта необходимо выполнить 21 задачу. Таблица 2 содержит список затрачиваемых ресурсов проекта.

Таблица 2 – Список ресурсов проекта

Название ресурса	Тип	Стандартная ставка	Ставка сверхурочных
Программист1	Трудовой	70 Р\ч	130 Р\ч
Программист2	Трудовой	60 Р\ч	130 Р\ч
Интернет	Затраты		
Компьютер	Материальный	1000 Р	
Смартфон с ОС Android	Материальный	1000 Р	

Отообразим список ресурсов в задачах, необходимых для выполнения проекта (таблица 3).

Таблица 3 – Назначения ресурсов

№	Название задачи	Ресурс	Единицы
1	Начало проекта		
2	Анализ требований для проекта разработки мобильного приложения для управления спортивными мероприятиями		
3	Обзор приложений-аналогов	Программист1 Интернет Компьютер	100 % 1 шт.
4	Составление списка функциональных возможностей приложения	Программист1 Компьютер	100 % 1 шт.
5	Выбор программного средства для реализации приложения	Программист1 Интернет Компьютер	100 % 1 шт.
6	Анализ требований завершен		
7	Проектирование для проекта разработки мобильного приложения для управления спортивными мероприятиями		
8	Проектирование интерфейса приложения	Программист1 Программист2 Интернет Компьютер	100 % 100 % 2 шт.
9	Проектирование структур данных	Программист1 Программист2 Интернет Компьютер	100 % 100 % 2 шт.

Продолжение таблицы 3

№	Название задачи	Ресурс	Единицы
10	Выбор архитектуры программного обеспечения	Программист1 Программист2 Интернет Компьютер	100 % 50 % 2 шт.
11	Проектирование завершено		
12	Реализация		
13	Программирование классов приложения	Программист1 Интернет Компьютер	100 % 1 шт.
14	Создание базы данных	Программист2 Интернет Компьютер	100 % 1 шт.
15	Интеграция клиентской и серверной частей приложения	Программист1 Программист2 Интернет Компьютер	100 % 50 % 2 шт.
16	Реализация завершена		
17	Отладка		
18	Тестирование работы приложения	Программист1 Программист2 Интернет Компьютер Смартфон с ОС Android	100 % 100 % 2 шт. 1 шт.
19	Исправление ошибок	Программист1 Программист2 Интернет Компьютер	100 % 100 % 2 шт.

Окончание таблицы 3

№	Название задачи	Ресурс	Единицы
20	Отладка завершена		
21	Конец проекта		

Из представленных таблиц видно, что решением задач будут заниматься два программиста, которым для этого необходимы компьютеры, доступ к Интернету, а также смартфон с ОС Android для тестирования приложения.

Вручную просчитаем время, которое необходимо для реализации проекта и вычислим его критический путь. Для этого построим сетевой график проекта (рисунок 17).

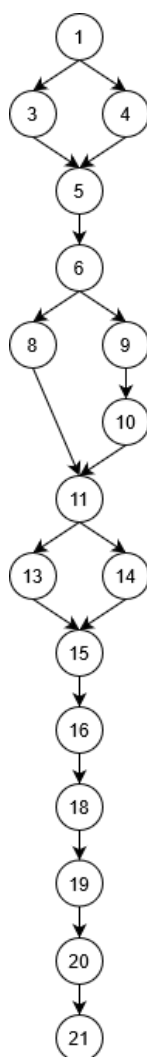


Рисунок 17 – Сетевой график проекта

Обозначим через t_i время выполнения работы i , G – множество работ, непосредственно предшествующих работе i , H – множество работ, непосредственно следующих за работой i . Для вычисления критического пути необходимо для каждой задачи определить раннее и позднее время начала работ. Раннее время определяется по формуле:

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

Позднее время начала следует вычислять по формуле:

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

Вычисление раннего времени начала работ приведено в таблице 4, вычисление позднего – в таблице 5.

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

Для вычисления критического пути необходимо для каждой задачи определить раннее и позднее время начала работ [21]. Раннее время определяется по формуле:

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

Позднее время начала следует вычислять по формуле:

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

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	1	0	0	0+0	0
5	3	0	5	Max(0+5;0+4)	5
	4	0	4		
6	5	5	4	5+4	9
8	6	9	0	9+0	9
9	6	9	0	9+0	9
10	9	9	5	9+5	14

Окончание таблицы 4

Работа	Предшественники			Вычисление	Раннее время начала
	№	Раннее время начала	Длительность		
11	8 10	9 14	3 4	Max(9+3;14+4)	18
13	11	18	0	18+0	18
14	11	18	0	18+0	18
15	13 14	18 18	8 5	Max(18+8;18+5)	26
16	15	26	4	26+4	30
18	16	30	0	30+0	30
19	18	30	8	30+8	38
20	19	38	7	38+7	45
21	20	45	0	45+0	45

Из результатов таблицы видно, что выполнение проекта должно занять 45 рабочих дней.

Таблица 5 – Вычисление позднего времени начала работ

Работа	Последователи		Длительность	Вычисление	Позднее время начала
	№	Позднее время начала			
21	-	-	-	-	45
20	21	45	0	45-0	45
19	20	45	7	45-7	38
18	19	38	8	38-8	30
16	18	30	0	30-0	30
15	16	30	4	30-4	26
14	15	26	5	26-5	21
13	15	26	8	26-8	18
11	13 14	18 21	0 0	Min(18-0;21-0)	18
10	11	18	4	18-4	14
9	10	14	5	14-5	9
8	11	18	3	18-3	15
6	8 9	15 9	0	Min(15-0;9-0)	9
5	6	9	4	9-4	5
4	5	5	4	5-4	1
3	5	5	5	5-5	0
1	3 4	0 1	0 0	Min(0-0;1-0)	0

Полученные результаты раннего и позднего времени начала работ позволяют вычислить критический путь для проекта (таблица 6).

Таблица 6 – Вычисление критического пути

Работа №	Раннее время начала	Позднее время начала	Резерв времени
1	0	0	0
3	0	0	0
4	0	1	1
5	5	5	0
6	9	9	0
8	9	15	6
9	9	9	0
10	14	14	0
11	18	18	0
13	18	18	0
14	18	21	3
15	26	26	0
16	30	30	0
18	30	30	0
19	38	38	0
20	45	45	0
21	45	45	0

Критический путь представлен на рисунке 18.

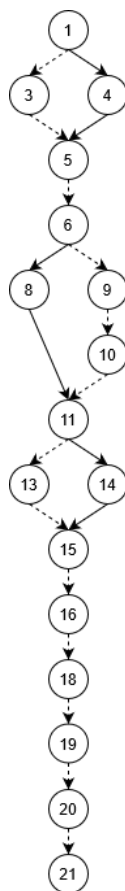


Рисунок 18 – Критический путь проекта

При помощи системы управления проектами Microsoft Project выполним планирование задач проекта (рисунок 19). Датой начала разработки проекта было принято 23 марта [22].

	Название задачи	Длительность	Начало	Окончание	Предшественники
1	Начало проекта	0 дней	Пн 23.03.20	Пн 23.03.20	
2	Анализ требований для проекта разработки приложения	9 дней	Пн 23.03.20	Чт 02.04.20	
3	Обзор приложений-аналогов	5 дней	Пн 23.03.20	Пт 27.03.20	1
4	Составление списка функциональных требований	4 дня	Пн 23.03.20	Чт 26.03.20	1
5	Выбор программного средства для разработки	4 дня	Пн 30.03.20	Чт 02.04.20	4;3
6	Анализ требований завершен	0 дней	Чт 02.04.20	Чт 02.04.20	5
7	Проектирование для проекта разработки приложения	9 дней	Пт 03.04.20	Ср 15.04.20	
8	Проектирование интерфейса приложения	3 дня	Пт 03.04.20	Вт 07.04.20	6
9	Проектирование структур данных	5 дней	Пт 03.04.20	Чт 09.04.20	6
10	Выбор архитектуры программного обеспечения	4 дня	Пт 10.04.20	Ср 15.04.20	9
11	Проектирование завершенно	0 дней	Ср 15.04.20	Ср 15.04.20	10;8
12	Реализация	12 дней	Чт 16.04.20	Пт 01.05.20	
13	Программирование классов приложения	8 дней	Чт 16.04.20	Пн 27.04.20	11
14	Создание базы данных	5 дней	Чт 16.04.20	Ср 22.04.20	11
15	Интеграция клиентской и серверной частей	4 дня	Вт 28.04.20	Пт 01.05.20	13;14
16	Реализация завершенна	0 дней	Пт 01.05.20	Пт 01.05.20	15
17	Отладка	15 дней	Пн 04.05.20	Пт 22.05.20	
18	Тестирование работы приложения	8 дней	Пн 04.05.20	Ср 13.05.20	16
19	Исправление ошибок	7 дней	Чт 14.05.20	Пт 22.05.20	18
20	Отладка завершенна	0 дней	Пт 22.05.20	Пт 22.05.20	19
21	Конец проекта	0 дней	Пт 22.05.20	Пт 22.05.20	20

Рисунок 19 – Планирование задач в Microsoft Project

Microsoft Project по задачам, введенным в системе автоматически выстроил диаграмму Ганта (рисунок 20).



Рисунок 20 – Диаграмма Ганта по проекту

На рисунке 21 изображен отчет проекта, на котором можно увидеть подсчитанную Microsoft Project длительность реализации проекта – 45 дней и предполагаемые затраты – 57 480 руб.

Даты			
Начало:	Пн 23.03.20	Окончание:	Пт 22.05.20
Баз. начало:	НД	Баз. окончание:	НД
Факт. начало:	НД	Факт. окончание:	НД
Отклон. начала:	0 дней	Отклон. окончания:	0 дней
Длительность			
Текущий план:	45 дней	Оставшаяся:	45 дней
Базовый план:	0 дней?	Фактическая:	0 дней
Отклонение:	45 дней	Процент завершения	0%
Трудозатраты			
Текущий план:	672 ч	Оставшиеся:	672 ч
Базовый план:	0 ч	Фактические:	0 ч
Отклонение:	672 ч	Процент завершения	0%
Затраты			
Текущий план:	57 480,00 Р	Оставшиеся:	57 480,00 Р
Базовый план:	0,00 Р	Фактические:	0,00 Р
Отклонение:	57 480,00 Р		
Состояние задачи		Состояние ресурса	
Еще не начатые задачи:	21	Трудовые ресурсы:	1
Выполняющиеся задачи:	0	Трудовые ресурсы с прев. дост.	2
Завершенные задачи:	0	Материальные ресурсы:	2
Итого задач:	21	Итого ресурсов:	5

Рисунок 21 – Отчет проекта

Таким образом, было проведено грамотное планирование проекта, а также оценена его стоимость. Полученные результаты свидетельствуют об экономической целесообразности данного проекта.

2 Проектирование мобильного приложения

2.1 Выбор технологий и инструментальных средств разработки

Выделяют несколько технологий разработки программного продукта - каскадная модель, спиральная модель, технологии XP, RUP, RAD. Некоторые технологии не подходят под разработку приложения, например, каскадная, предполагающая придерживаться строгой последовательности всех этапов разработки ПО и не имеющая возможности возвращения с текущего этапа на предыдущий, в настоящее время данная технология практически не используется; технология RAD, предусматривающая активное привлечение заказчика уже на ранних стадиях для выработки требований к системе. Явной проблемой спиральной модели является определение момента перехода на следующий этап разработки – в разрабатываемом приложении есть определенные сроки, когда программный продукт должен быть реализован, поэтому спиральной моделью было принято решение так же пренебречь.

Приложение разрабатывалось при помощи методологии разработки ПО RUP. Данная технология была выбрана не случайно – она является одной из самых популярных методологий разработки ПО в настоящее время. Данная модель разработки соответствует в значительной степени всем нормативным документам, связанными с процессами жизненного цикла программного обеспечения. Большим плюсом разработки программного обеспечения в RUP является разделение проекта на несколько мелких проектов, которые выполняются последовательно, и каждая итерация разработки четко определена набором целей, которые должны быть достигнуты в конце итерации. Конечная итерация предполагает, что набор целей итерации должен в точности совпадать с набором целей, указанных заказчиком продукта, то есть все требования должны быть выполнены. Таким образом, уже на стадии планирования целей и задач проекта, можно разделить команду разработчиков на конкретные мелкие проекты [23].

Мобильное приложение разрабатывалось с помощью технологии Xamarin, ориентированную на ОС Android. Для разработки мобильного приложения использовалась среда разработки Microsoft Visual Studio 2019, язык программирования C#.

Серверная часть приложения разрабатывалась на языке программирования C# в среде разработки Microsoft Visual Studio 2019. Серверная часть включает в себя API – интерфейс прикладного программирования [24]. Для хранения данных использовалась СУБД Microsoft SQL Server 2016, языком запросов для базы данных был выбран Transact-SQL. Для тестирования серверной части использовалось инструментальное средство Postman.

В качестве языка моделирования методология RUP использует язык UML – язык графического описания для объектного моделирования в области разработки программного обеспечения. Для проектирования UML-диаграмм использовался векторный графический редактор Microsoft Visio. Планирование проекта и оценка бюджета были выполнены при помощи программы Microsoft Project.

2.2 Проектирование архитектуры программного обеспечения

При разработке мобильного приложения для управления спортивными мероприятиями была выбрана двухуровневая архитектура – клиент-сервер (рисунок 22).

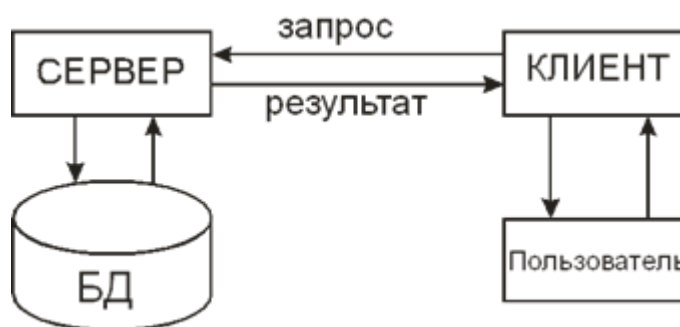


Рисунок 22 – Клиент-серверная архитектура

При таком подходе на сервере находится база данных вместе с СУБД и прикладными программами, клиентская часть содержит только уровень представления данных [24]. Применение такой архитектуры имеет ряд сильных сторон:

- все данные хранятся на сервере, который защищен гораздо лучше, чем клиенты;
- снижается нагрузка на клиентские приложения;
- программный код серверной и клиентской частей приложения разделен, это упрощает дальнейшую работу по доработке или изменению программного продукта.

Взаимодействие между клиентом и сервером осуществляется следующим образом: клиент посылает сервису API сервера HTTP-запросы, интерфейс прикладного программирования API обрабатывает этот запрос и посылает запрос на языке SQL базе данных Microsoft SQL Server. В случае успешной обработки запроса клиенту возвращаются данные в формате JSON – текстовый формат обмена данными [25]. После этого клиентская часть отображает полученные данные в удобной форме.

2.3 Проектирование пользовательского интерфейса

На основе выявленных функциональных требований был спроектирован пользовательский интерфейс мобильного приложения. К основным страницам приложения относятся следующие: страница авторизации, страница организатора, страница добавления турнира, страница добавления команды, страница добавления игрока, страница генерации календаря турнира, страница редактирования календаря турнира, страница секретаря, главная страница обычного пользователя, страница с турнирами, информационная страница по турниру, информационная страница по команде, информационная страница по игроку и информационная страница по матчу.

Рассмотрим элементы, расположенные на странице авторизации (рисунок 23).

Цифрами на рисунке отмечены следующие компоненты страницы:
1 – текстовое поле для ввода логина, 2 – текстовое поле для ввода пароля,
3 – кнопка подтверждения авторизации, 4 – кнопка входа в приложение в качестве пользователя.

Данная страница содержит 2 кнопки, предназначенные для перехода к функциональным возможностям приложения. Если приложением пользуется обычный пользователь, желающий просматривать информацию о турнирах, то ему достаточно войти в приложение как обычный пользователь. Если же в приложение входит организатор турнира, отвечающий за организацию соревнований или секретарь матча, то им необходимо предварительно пройти авторизацию, заполнив 2 текстовых поля для ввода логина и пароля и подтвердить авторизацию, нажав кнопку, отмеченную на рисунке 23 цифрой 3.



Рисунок 23 – Страница авторизации

При успешном прохождении авторизации, пользователь попадает на страницу организатора (рисунок 24), содержащую следующие компоненты:
1 – кнопка перехода на страницу добавления турнира, 2 – кнопка перехода на страницу добавления команды, 3 – кнопка перехода на страницу добавления

игрока, 4 – кнопка перехода на страницу генерации календаря турнира, 5 – кнопка перехода на страницу редактирования календаря турнира. Данная страница состоит из 5 кнопок, при нажатии на каждую из которых пользователь переходит на соответствующие страницы.

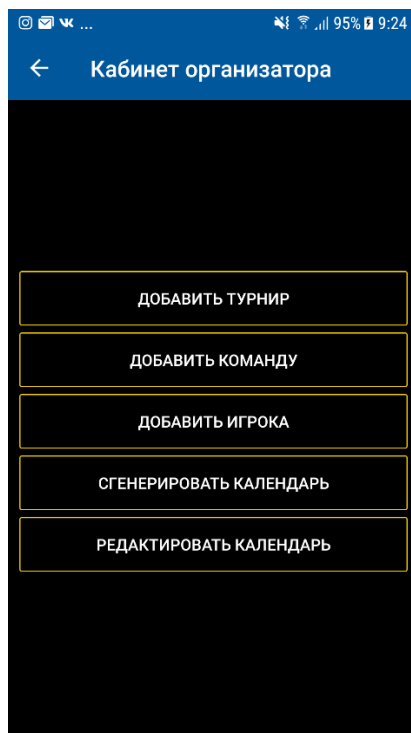


Рисунок 24 – Страница организатора

На рисунке 25 представлена страница добавления турнира, состоящая из следующих компонентов: 1 – переключатель видов спорта, 2 – текстовое поле для ввода названия турнира, 3 – текстовое поле для краткого описания турнира, 4 – переключатель форматов турнира, 5 – текстовое поле для указания временного промежутка игры, 6 – переключатель количества команд, 7 – кнопка, добавляющая в базу данных информацию по турниру.

Страница состоит из 7 элементов. Переключатель, отмеченный на рисунке 25 цифрой 1, включает в себя значения тех видов спорта, турниры которых возможно создать в приложении – это футбол, хоккей и регби. Текстовые поля, отмеченные цифрами 2 и 3 на рисунке 25 предназначены для ввода названия турнира и краткого описания по нему, соответственно. Переключатель, отмеченный на рисунке 25 цифрой 4, содержит значения

форматов турнира, – круговой турнир и формат плей-офф. Каждый матч каждой спортивной дисциплины длится определенное количество времени, в футболе и регби матч состоит из 2 таймов, в хоккее игра состоит из 3 периодов. Текстовое поле, отмеченное на рисунке 25 цифрой 5, предназначено для ввода времени, отведенного на один тайм, или один период. Переключатель, отмеченный на рисунке 25 цифрой 6, предназначен для установления количества команд, участвующих в турнире. При нажатии на кнопку, отмеченную цифрой 7 на рисунке 25, при том условии, что все остальные поля заполнены, происходит передача введенных данных серверу и дальнейшее внесение информации о турнире в базу данных.



Рисунок 25 – Страница добавления турнира

На рисунке 26 представлена страница добавления команды, состоящая из следующих элементов: 1 – переключатель турниров, 2 – текстовое поле, отображающее вид спорта, 3 – текстовое поле для ввода названия команды, 4 – кнопка, добавляющая в базу данных информацию по команде.

Страница состоит из 4 элементов. Переключатель, отмеченный цифрой 1 на рисунке 26, включает в себя названия всех турниров, куда возможно

добавить команду. Текстовое поле, отмеченное на рисунке 26 цифрой 2 отображает вид спорта, турнир которого установлен в переключателе. Текстовое поле, отмеченное на рисунке 26 цифрой 3 предназначено для ввода названия турнира. При нажатии на кнопку, отмеченную на рисунке 26 цифрой 4, при том условии, что все остальные поля заполнены, происходит передача введенных данных серверу и дальнейшее внесение информации о команде в базу данных.



Рисунок 26 – Страница добавления команды

На рисунке 27 представлена страница добавления игрока в команду, состоящая из следующих компонентов: 1 – текстовое поле для ввода имени и фамилии игрока, 2 – компонент выбора даты рождения игрока, 3 – переключатель турниров, 4 – переключатель команд турнира, 5 – переключатель амплуа игрока, 6 – переключатель свободных номеров в команде, 7 – кнопка, добавляющая в базу данных информацию об игроке.

Страница состоит из 7 элементов. Текстовое поле, отмеченное цифрой 1 на рисунке 27 предназначено для ввода имени и фамилии игрока. Компонент выбора даты рождения, отмеченный на рисунке 27 цифрой 2 предназначен для

указания даты рождения добавляемого спортсмена. Переключатель турниров, отмеченный цифрой 3 на рисунке 27, содержит в себя значения всех турниров. Переключатель команд, отмеченный на рисунке 27 цифрой 4, содержит все команды из турнира, выбранного в предыдущем переключателе, переключатель, отмеченный на рисунке 27 цифрой 5 содержит значения игровых амплуа, которые могут быть установлены для игрока. Переключатель, отмеченный на рисунке 27 цифрой 6, содержит значения всех доступных игровых номеров в команде. В одной команде не может существовать игроков с одинаковыми игровыми номерами. При нажатии на кнопку, отмеченную на рисунке 27 цифрой 7, при том условии, что все остальные поля заполнены, происходит передача введенных данных серверу и дальнейшее внесение информации об игроке в базу данных.

Рисунок 27 – Страница добавления игрока

На рисунке 28 представлена страница генерации календаря турнира, которая состоит из двух компонентов – переключателя турниров, который содержит в качестве элементов названия тех турниров, календарь для которых не был сформирован ранее (на рисунке 28 отмечен цифрой 1) и кнопку,

генерирующую календарь для выбранного турнира (на рисунке 28 отмечен цифрой 2). При нажатии на кнопку осуществляется запрос на сервер, где происходит вызов метода, генерирующего календарь турнира, после его генерации происходит внесение сгенерированных данных в базу данных.

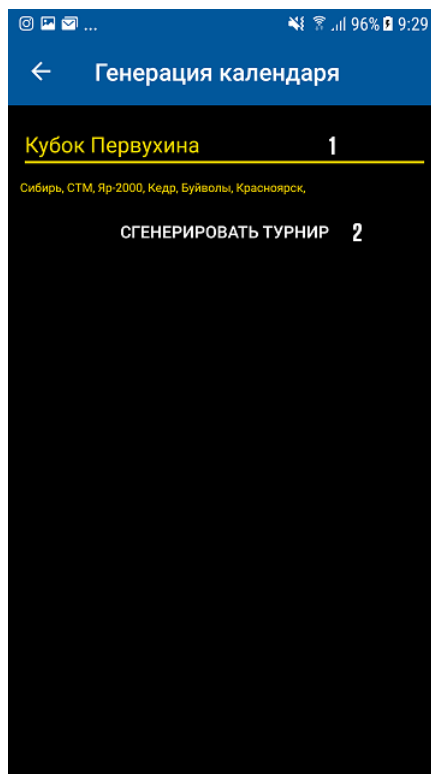


Рисунок 28 – Страница генерации календаря турнира

На рисунке 29 и 30 представлена страница редактирования календаря турнира, страница состоит из двух компонентов – переключателя турниров, элементами которого являются названия турниров (отмечен на рисунке 29 цифрой 1) и списка игр, при нажатии на каждую строку которого происходит переход к редактированию даты и времени конкретного матча (рисунок 30).

При помощи данной страницы можно изменить дату и время конкретного матча в каком-либо турнире. Страница содержит 4 компонента: 1 – надпись, содержащая названия команд, участвующих в матче, 2 – компонент установления даты матча, 3 – компонент установления времени матча, 4 – кнопку, при нажатии на которую изменяется дата и время. После нажатия на кнопку, отмеченную на рисунке 30 цифрой 4, происходит передача

установленных данных серверу и дальнейшее внесение информации о дате и времени матча в базу данных.



Рисунок 29 – Страница редактирования календаря турнира (первая часть)



Рисунок 30 – Страница редактирования календаря турнира (вторая часть)

На рисунке 31 представлена страница секретаря, состоящая из следующих компонентов: 1 – кнопка, позволяющая заполнить протокол матча, 2 – кнопка, позволяющая заполнить состав матча.



Рисунок 31 – Страница секретаря

На рисунке 32 представлена главная страница обычного пользователя. Она содержит единственный компонент – список, отображающий 3 спортивные дисциплины, турниры которых имеются в системе. Список обозначен на рисунке 32 цифрой 1.

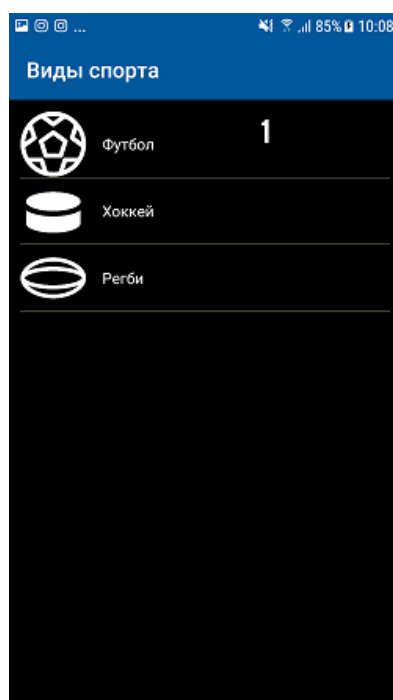


Рисунок 32 – Главная страница обычного пользователя

На рисунке 33 представлена страница с турнирами. Она содержит единственный компонент – список, отображающий все турниры по ранее выбранной дисциплине. Список обозначен на рисунке 33 цифрой 1.



Рисунок 33 – Страница с турнирами

На рисунке 34 представлена информационная страница по турниру. Она состоит из следующих компонентов: 1 – вкладка, отображающая таблицу турнира, 2 – вкладка, отображающая календарь турнира, 3 – вкладка, отображающая результаты турнира, 4 – вкладка, отображающая снайперов турнира, 5 – вкладка, отображающая ассистентов турнира, 6 – вкладка, отображающая бомбардиров турнира.

Страница состоит из 6 вкладок, при нажатии на ту или иную перед пользователем появляется соответствующая информация по турниру – таблица, календарь, результаты или список лучших игроков турнира. На рисунке 34 представлен тот вариант, когда открыта вкладка с таблицей турнира, на ней отображено турнирное положение команд, количество сыгранных игр для каждой команды, число побед и количество набранных очков. Так же на рисунке представлен компонент, отмеченный цифрой 7, при нажатии на какую-либо строку которого, происходит переход на информационную страницу команды.

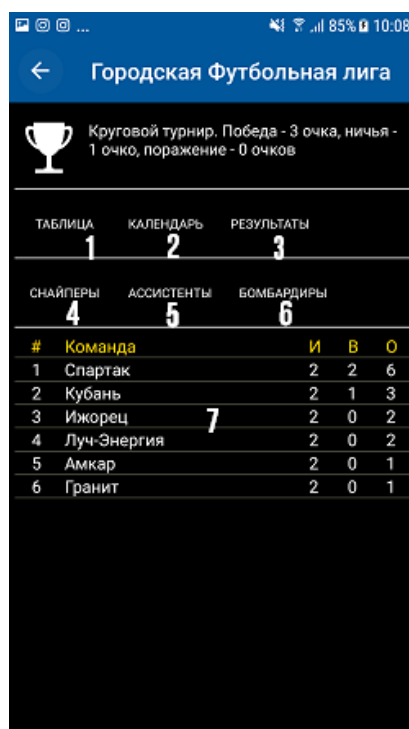


Рисунок 34 – Информационная страница по турниру

Информационная страница команды (рисунок 35) состоит из следующих компонентов: 1 – вкладка, отображающая результаты команды в турнире, 2 – вкладка, отображающая календарь команды в турнире, 3 – вкладка, отображающая состав команды. Страница состоит из 3 вкладок, при нажатии на ту или иную перед пользователем появляется соответствующая информация по команде – результаты, календарь или состав. В данном случае существует возможность нажатия на строку компонента, отмеченного цифрой 4, и происходит переход на информационную страницу игрока (рисунок 36), состоящую из 1 компонента, который позволяет переходить к протоколам сыгранных матчей. На рисунке 36 этот компонент отмечен цифрой 1. Страница содержит информацию об игровом амплуа игрока, его возраст, название команды, а так же игровой номер. Так же страница отображает игровую статистику спортсмена в играх, в которых он принимал участие, демонстрируется число голов и ассистентских передач в каждой такой игре.

#	Игрок	Позиция
28	Сергей Веремко	Вр.
44	Михаил Спиридонов	Вр.
33	Сергей Агеев	Зщ.
3	Михаил Шукаев	Зщ.
17	Эдуард Шусе	Зщ.
22	Александр Петров	Зщ.
29	Михаил Якунин	Зщ.
55	Дмитрий Рогозин	Зщ.
14	Лев Петров	Зщ.
4	Пётр Кочетков	Пзщ.
25	Борис Николаев	Пзщ.
10	Николай Зацепин	Пзщ.
16	Ильнур Маметов	Пзщ.
90	Илья Устинов	Пзщ.
98	Давид Пашинян	Пзщ.
27	Клим Рогов	Пзщ.
68	Александр Пахутов	Пзщ.
87	Кирилл Журавлев	Нап.
7	Ярослав Брыкин	Нап.

Рисунок 35 – Информационная страница команды

Дата	Команда	Г	А	Счет
08.05	Гранит	0	0	0-0
12.05	Луч-Энергия	1	1	2-2

Рисунок 36 – Информационная страница игрока

Информационная страница матча (рисунок 37) содержит 2 вкладки: 1 – вкладка с протоколом игры, 2 – вкладка с составами команд.

На вкладке с протоколом матча можно посмотреть основные действия матча – голы, удаления. Рядом с действиями матча отображаются игроки,

причастные к тому или иному действию. Вкладка с составом команд содержит составы команд.



Рисунок 37 – Информационная страница матча

Таким образом, было проведено проектирование интерфейса мобильного приложения.

2.4 Проектирование серверной части приложения

2.4.1 Логическое проектирование базы данных

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

Типы сущностей предметной области представлены в таблице 7.

Таблица 7 – Типы сущностей

№	Название	Описание	Категория	Количество экземпляров	Режим обновления
1	Команда	Список всех команд	Стержневая	1000	25% в год
2	Вид спорта	Список спортивных дисциплин	Характеристическая	3	

Продолжение таблицы 7

№	Название	Описание	Категория	Количество экземпляров	Режим обновления
3	Амплуа	Позиция спортсмена в команде	Характеристическая	9	
4	Результаты	Список сыгранных матчей	Стержневая	2000	25 % в год
5	Игра Плей-офф	Результаты сыгранных матчей серии плей-офф	Стержневая	1000	10 % в год
6	Турнирная таблица	Характеризует результаты выступления команды в турнире	Стержневая	1000	250 в месяц
7	Турнир	Вид соревнований	Стержневая	1500	500 в месяц
8	Календарь	Список матчей, которые должны быть сыграны	Стержневая	2000	500 в месяц
9	Составы команд	Список игроков, принимающих участие в матче	Стержневая	5000	20 % в год
10	Участие в игре	Информация о том играл игрок или пропускал игру	Характеристическая	2	
11	Игрок	Список всех игроков	Стержневая	2000	500 в месяц
12	Протокол матча	Список событий, происходящих в матче	Стержневая	5000	1500 в месяц
13	Спортивное событие	Виды событий в определенном виде спорта	Характеристическая	20	
14	Ассистенты	Список игроков, отдавших голевую передачу	Стержневая	2000	500 в месяц
15	Снайперы	Список игроков, забивших гол	Стержневая	2000	500 в месяц

Окончание таблицы 7

№	Название	Описание	Категория	Количество экземпляров	Режим обновления
16	Бомбардиры	Список игроков, отдавших голевую передачу или забивших гол	Стержневая	2000	500 в месяц

Между типами сущностей установлены определенные связи. Таблица 8 отображает связи между типами сущностей.

Таблица 8 – Типы связей

№	Тип сущности 1	Тип сущности 2	Мощность	Обязательность
1	Команда	Вид спорта	1:1	Обязательно
2	Команда	Турнир	1:1	Необязательно (При первоначальном добавлении команды ей не назначается турнир, он выбирается позже)
3	Команда	Игрок	1:M	Необязательно (При первоначальном добавлении команды ей не назначаются определенные игроки, они выбираются позже)
4	Команда	Составы команд	1:M	Необязательно (Составы команд заполняются после добавления команды)
5	Команда	Протокол матча	1:M	Необязательно (Протоколы матча заполняются после добавления команды)
6	Команда	Результаты	1:M	Необязательно (Результаты матчей добавляются после добавления команды)

Продолжение таблицы 8

№	Тип сущности 1	Тип сущности 2	Мощность	Обязательность
7	Команда	Игра Плей-офф	1:M	Необязательно (Игры плей-офф формируются после добавления команды)
8	Команда	Турнирная таблица	1:1	Необязательно (Турнир для команды выбирается позже, после ее добавления)
9	Команда	Календарь	1:1	Необязательно (Календарь формируется после добавления команды)
10	Игра Плей-офф	Турнир	M:1	Обязательно
11	Вид спорта	Турнир	1:M	Обязательно
12	Вид спорта	Амплуа	1:M	Обязательно
13	Вид спорта	Спортивное событие	1:M	Обязательно
14	Амплуа	Игрок	1:1	Обязательно
15	Результаты	Составы команд	1:M	Обязательно
16	Результаты	Протокол	1:M	Обязательно
17	Результаты	Турнир	M:1	Необязательно (Результаты турниров формируется после добавления турнира)
18	Турнирная таблица	Турнир	1:1	Обязательно
19	Календарь	Турнир	1:1	Обязательно
20	Турнир	Составы команд	1:M	Необязательно (Составы команд формируются после создания турнира)
21	Турнир	Ассистенты	1:M	Необязательно (Список ассистентов формируется после создания турнира)

Окончание таблицы 8

№	Тип сущности 1	Тип сущности 2	Мощность	Обязательность
22	Турнир	Снайперы	1:M	Необязательно (Список снайперов формируется после создания турнира)
23	Турнир	Бомбардиры	1:M	Необязательно (Список бомбардиров формируется после создания турнира)
24	Составы команд	Участие в игре	1:1	Обязательно
25	Составы команд	Игрок	1:M	Обязательно
26	Игрок	Протокол	M:1	Необязательно (Протокол формируется после проведенного матча)
27	Игрок	Ассистенты	1:1	Необязательно (Список ассистентов формируется после проведенного матча)
28	Игрок	Снайперы	1:1	Необязательно (Список снайперов формируется после проведенного матча)
29	Игрок	Бомбардиры	1:1	Необязательно (Список бомбардиров формируется после проведенного матча)
30	Протокол	Спортивное событие	1:M	Обязательно

Логическая модель базы данных, представленная в нотации IDEF1X, отображена на рисунках 38 и 39.

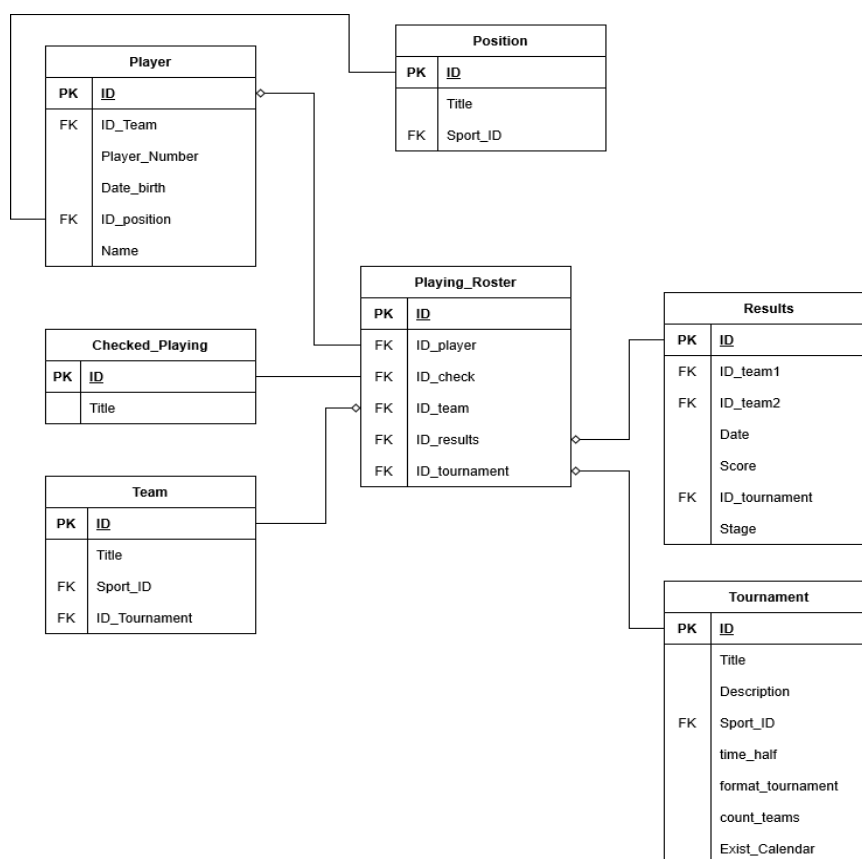


Рисунок 38 – Логическая модель базы данных (первая часть)

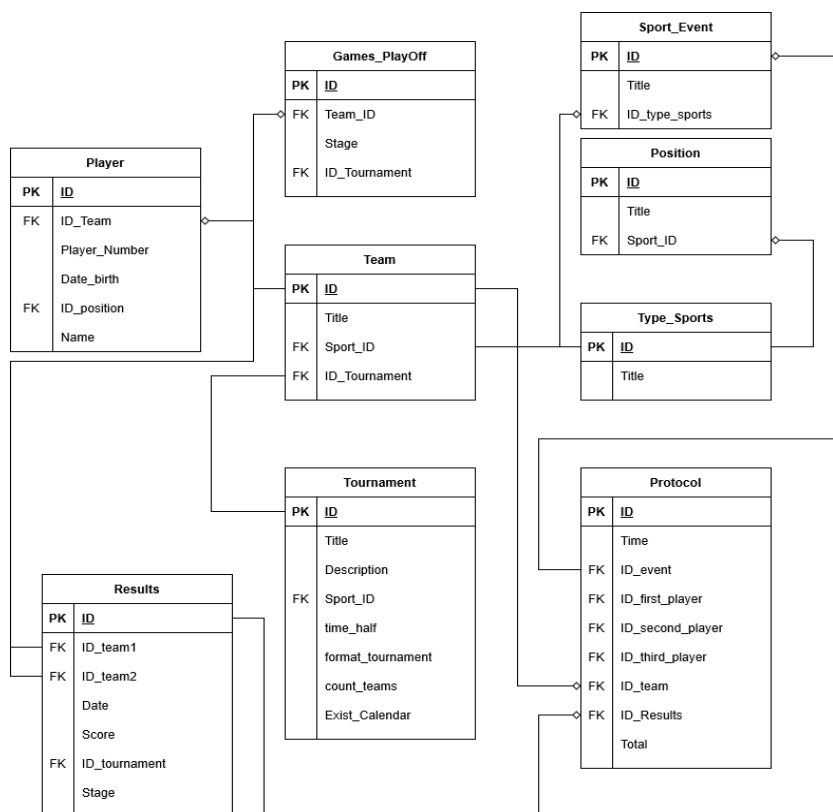


Рисунок 39 – Логическая модель базы данных (вторая часть)

Продолжение логической модели базы данных представлено на рисунках 40 и 41.

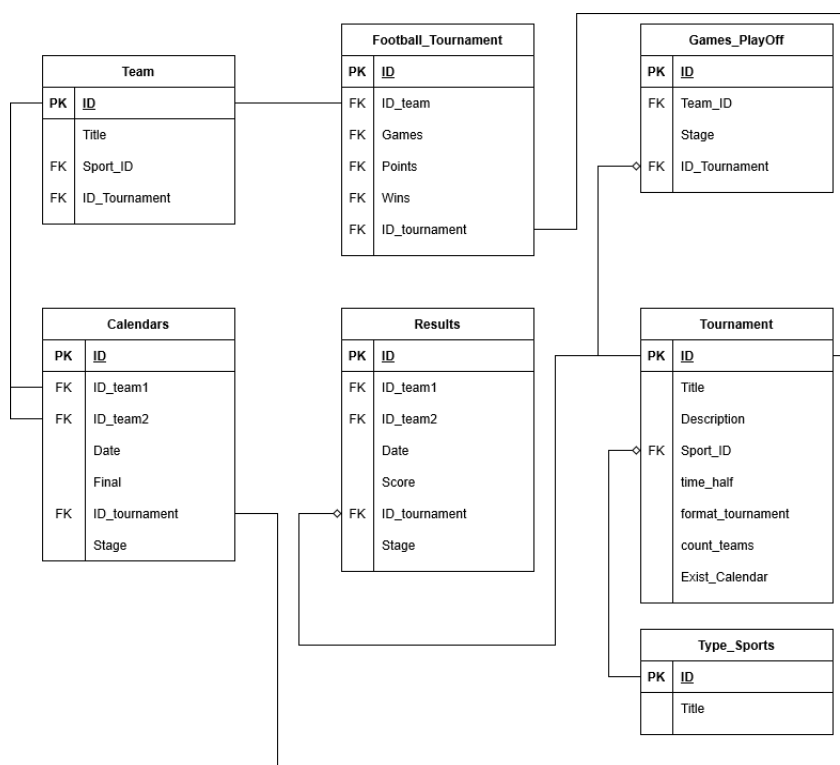


Рисунок 40 – Логическая модель базы данных (третья часть)

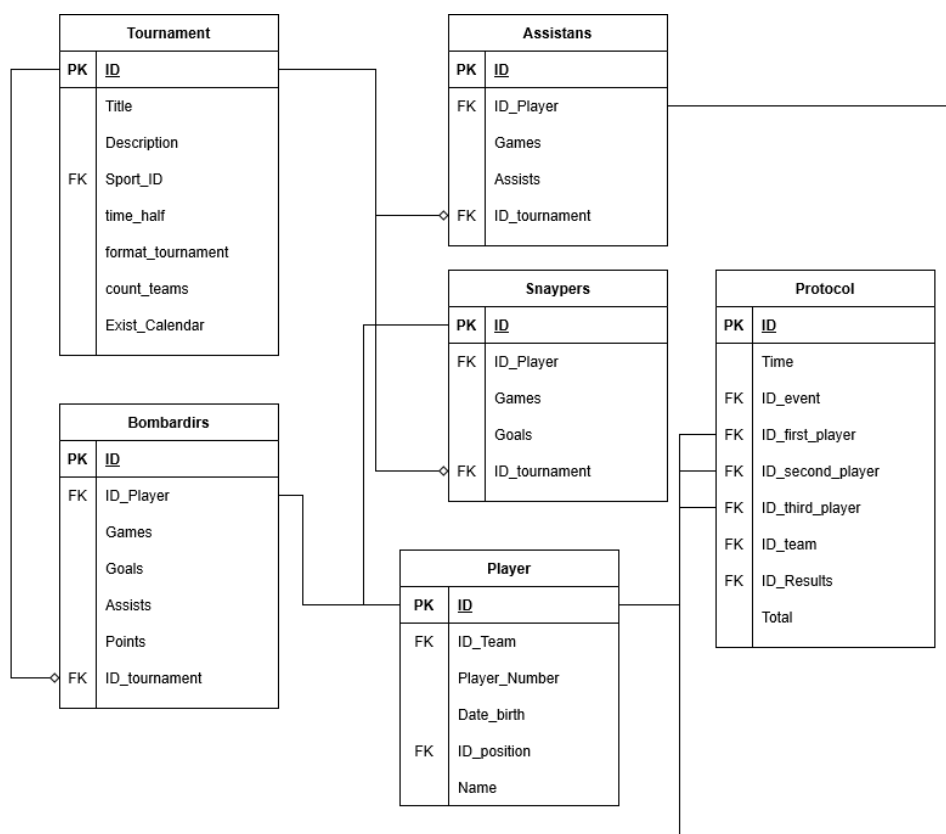


Рисунок 41 – Логическая модель базы данных (четвертая часть)

Логическая модель базы данных – концептуальная схема на основе представлений о предметной области, важной особенностью логической модели является то, что ее проектирование должно являться независимым от требований платформы и языка реализации или способа дальнейшего использования данных.

2.5 Проектирование API

Серверная часть приложения предоставляет прикладной программный интерфейс для работы с информацией, находящейся на сервере. Передача данных осуществляется при помощи HTTP-протокола, формат данных – JSON.

Веб-сервис разрабатывался на платформе .NET с помощью языка C#. При разработке сервиса использовалась архитектура MVC (Model-View-Controller). Архитектура представлена на рисунке 42.

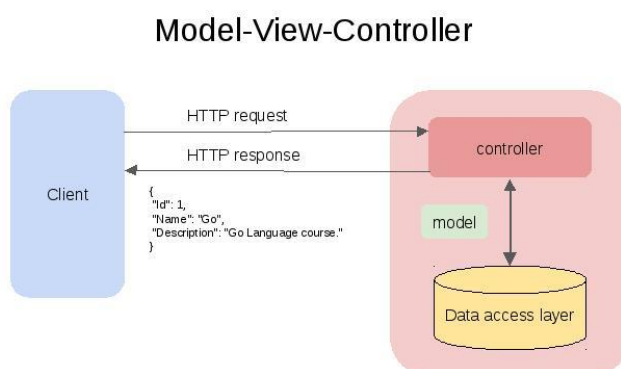


Рисунок 42 – Модель MVC

Принцип работы такой архитектуры заключается в получении контроллером запроса от клиентской части (view), интерпретации заданных действий, изменения состояния модели (model). После этого контроллер отправляет клиентской стороне необходимые данные, на стороне которой происходит их отображение.

Таким образом, разрабатываемая веб-служба содержит классы представления модели данных и класс контроллера, в состав которого входят

методы, реализующие все функциональные возможности приложения. Именно из методов контроллера происходит отправка запросов в базу данных и получение информации из нее.

3 Реализация мобильного приложения

3.1 Реализация серверной части

3.1.1 Реализация базы данных

На основе логической модели базы данных, была реализована физическая модель. Физическая модель строится на основе логической с учетом ограничений, накладываемых выбранной СУБД. В качестве СУБД выступал Microsoft SQL Server. Полученная физическая модель данных представлена на рисунках 43 – 45.

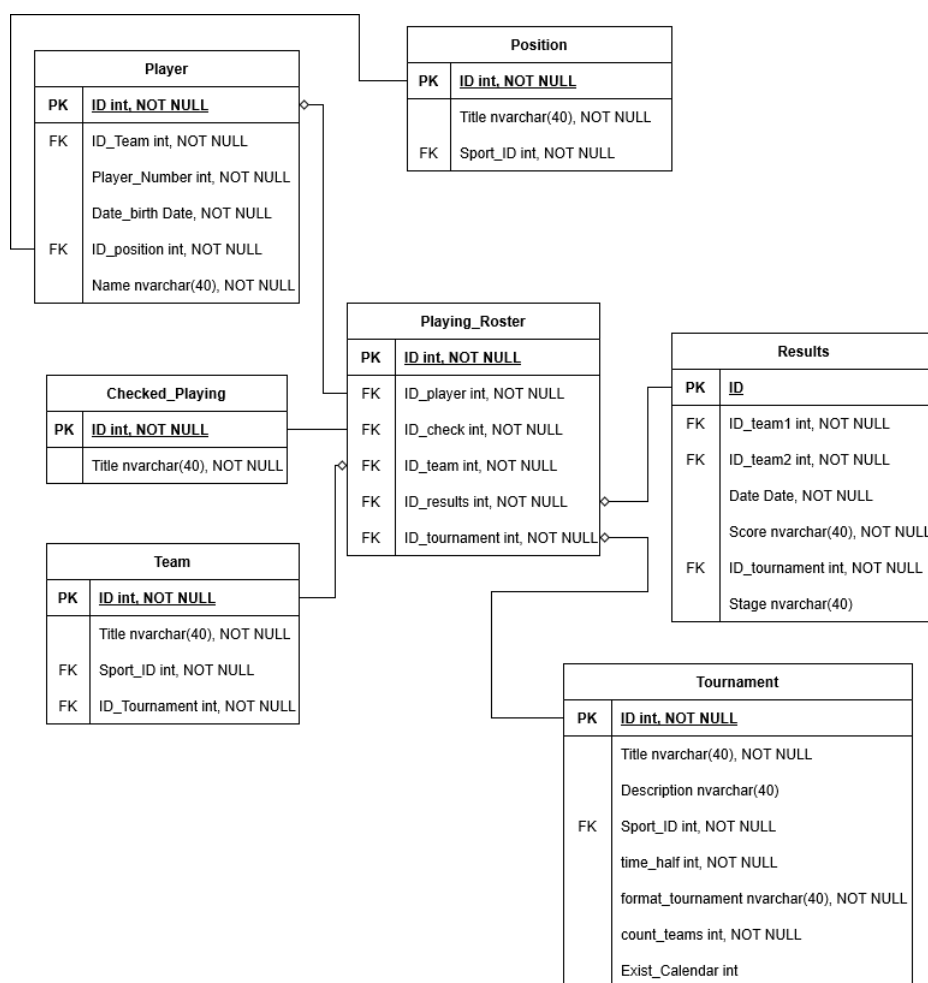


Рисунок 43 – Физическая модель базы данных (первая часть)

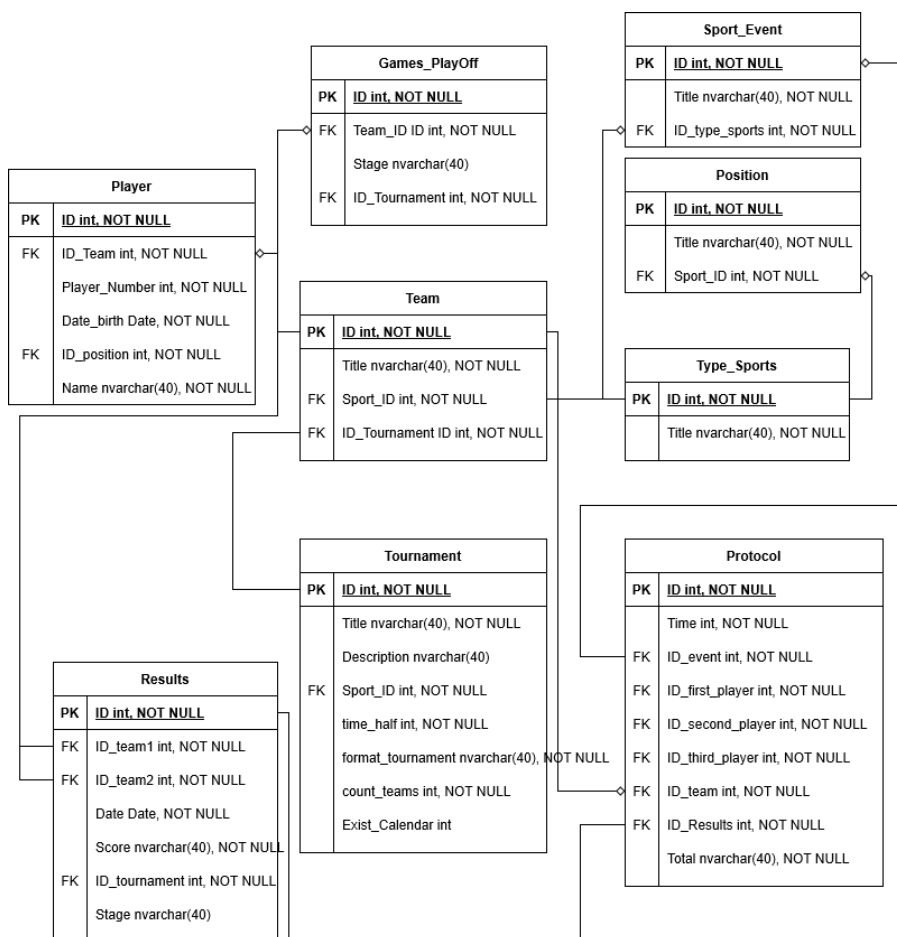


Рисунок 44 – Физическая модель базы данных (вторая часть)

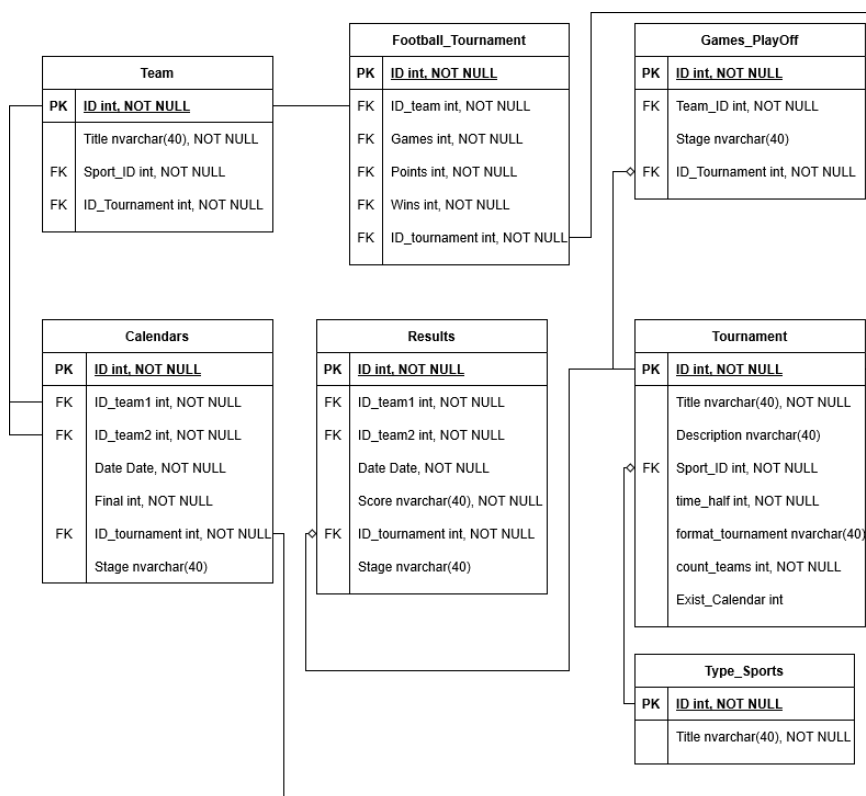


Рисунок 45 – Физическая модель базы данных (третья часть)

Продолжение физической модели базы данных представлено на рисунке 46.

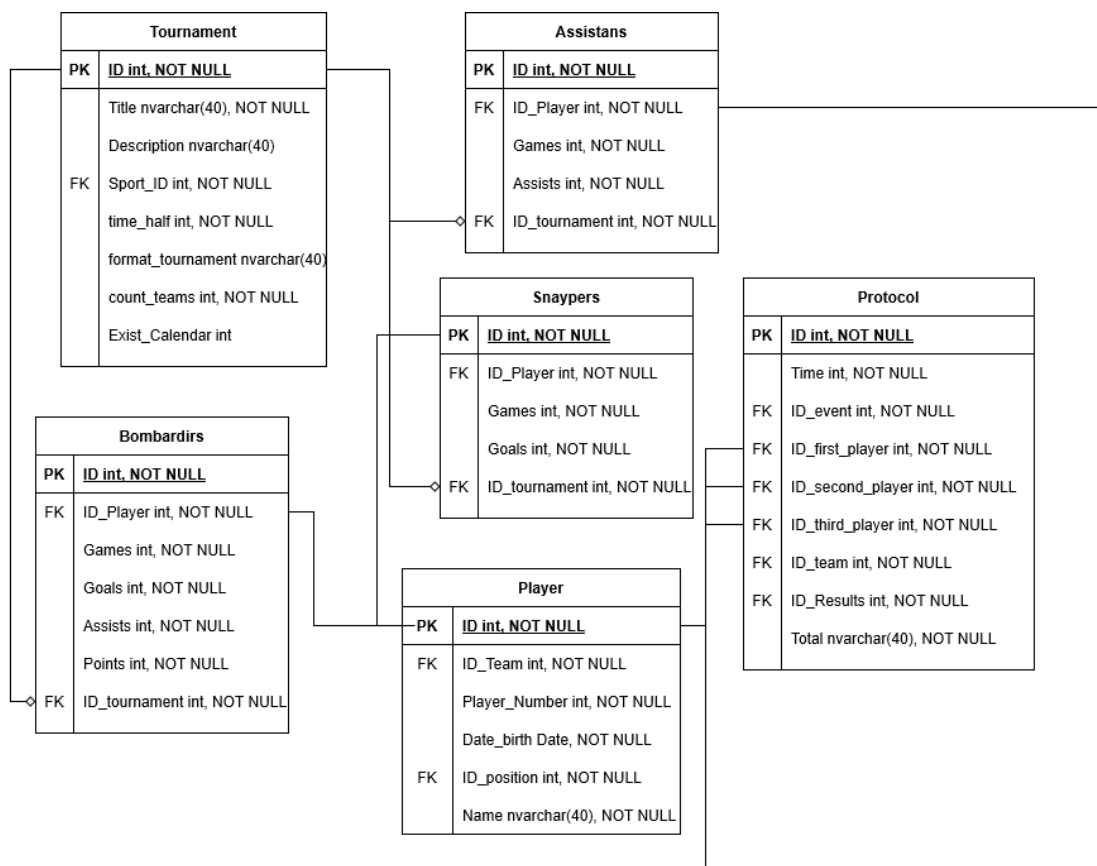


Рисунок 46 – Физическая модель базы данных (четвертая часть)

Таким образом была реализована база данных при помощи СУБД Microsoft SQL Server 2016.

3.1.2 Реализация серверной части приложения

Разработка API проводилась в среде программирования Microsoft Visual Studio 2019 с использованием языка программирования C#.

Серверная часть включает в себя классы, участвующие в процессах сериализации и десериализации данных (рисунок 47, 48), эти процессы участвуют при взаимодействии клиентской части приложения и разрабатываемого API, форматом обмена данными между ними является JSON. Для работы с JSON была подключена библиотека Newtonsoft.JSON. Библиотека предлагает хороший инструментарий, содержащий в себе все необходимое, что может понадобиться для разработки приложения [25, 26].

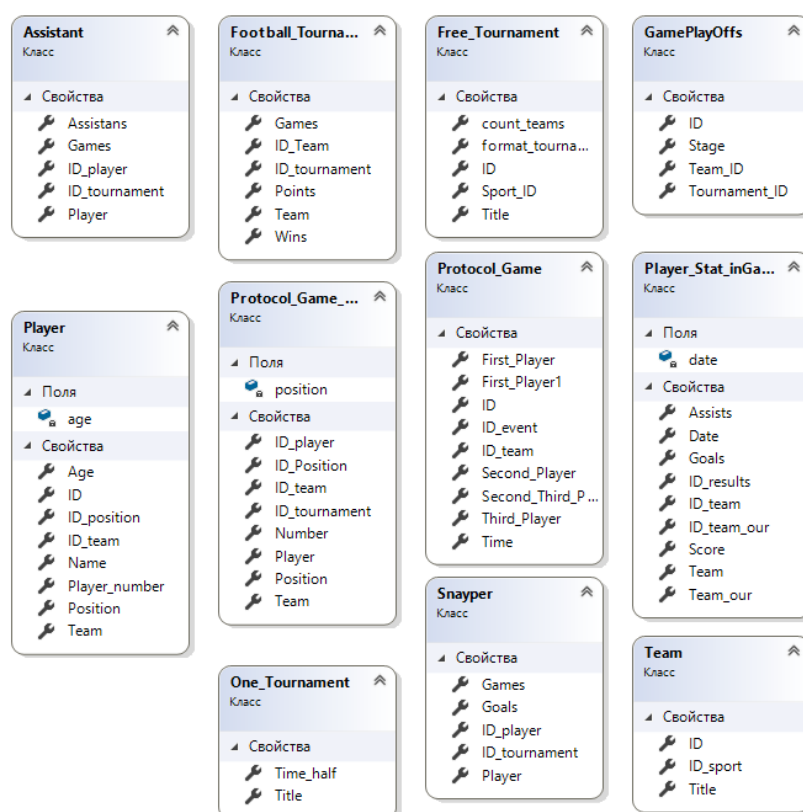


Рисунок 47 – Диаграмма классов серверной части (первая часть)

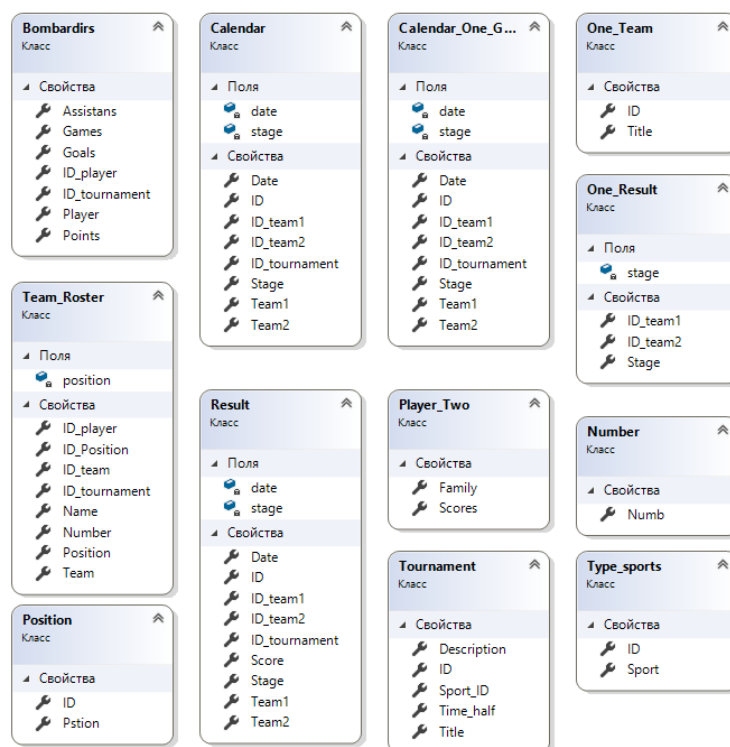


Рисунок 48 – Диаграмма классов серверной части (вторая часть)

При отправке ответа клиентской стороне серверная часть сериализует данные – преобразует определенную структуру данных в последовательность битов. Данную операцию можно выполнить при помощи статического метода `SerializeObject` класса `JsonConvert`, метод принимает в качестве входного параметра 1 аргумент – объект структуры данных (класса). На клиентской стороне происходит десериализация данных.

Серверная часть реализовывалась при помощи архитектуры MVC (Model-View-Controller). Класс контроллера выступает в роли посредника между клиентской частью приложения и серверной, он принимает запросы и посылает ответы клиенту. В состав класса контроллера входят методы, которые и принимают HTTP-запросы и посылают HTTP-ответы. Пример одного из таких методов приведен на рисунке 49.

```
public string Select_Tournaments()
{
    string conn = "Data Source =DESKTOP-SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated Security=True;";
    SqlConnection sqlConnection;
    SqlCommand command;
    SqlDataReader reader;
    sqlConnection = new SqlConnection(conn);
    sqlConnection.Open();
    command = new SqlCommand();
    command.CommandText = "Select * from [Tournament] ";
    command.Connection = sqlConnection;

    reader = command.ExecuteReader();
    MyTournaments myTournaments = new MyTournaments();
    Tournament tournament = new Tournament();
    while (reader.Read())
    {
        tournament.ID = Convert.ToInt32(reader["ID"]);
        tournament.Title = reader["Title"].ToString();
        tournament.Sport_ID = Convert.ToInt32(reader["Sport_ID"]);
        tournament.Time_half = Convert.ToInt32(reader["time_half"].ToString());
    }
    string serialized = JsonConvert.SerializeObject(tournament);
    sqlConnection.Close();
    return serialized;
}
```

Рисунок 49 – Метод `Select_Tournaments`

Данный метод отправляет клиентской части информацию о спортивных турнирах, имеющуюся в базе данных. Для работы с СУБД Microsoft SQL Server было подключено пространство имен `System.Data.SqlClient`. Серверная часть приложения формирует запрос на выборку всех данных из таблицы `Tournament`, после этого происходит считывание полученной информации в созданный объект класса `Tournament`, который подвергается сериализации, последовательность битов полученная при данном процессе

записывается в строку и отправляется на клиентскую сторону приложения, где в последствии десериализуется.

3.2 Реализация клиентской части мобильного приложения

Реализация клиентской части мобильного приложения происходила в среде программирования Microsoft Visual Studio 2019 на языке программирования C# с использованием технологии Xamarin.Forms.

Клиентская часть приложения включает в себя классы, участвующие в процессах сериализации и десериализации данных в серверной части приложения. Как было сказано ранее – эти процессы взаимодействуют между клиентской частью мобильного приложения и серверной. Стоит отметить, что классы, участвующие в этих процессах идентичны классам, участвующим в этих же процессах на серверной части мобильного приложения. Клиентская часть принимает ответ от серверной части и при помощи статического метода `DeserializeObject` класса `JsonConvert`, преобразует последовательность битов переданных серверной частью в определенную нужную структуру данных (класс) [26]. В качестве примера рассмотрим один из методов (рисунок 50), расположенных на клиентской стороне приложения, который отправляет HTTP-запрос типа POST на сервер и принимает от него ответ.

```
private void Get_Title_Team(string id)
{
    WebRequest request = WebRequest.Create("http://192.168.0.104/WebApplication1/Home/Return Title Team");
    request.Method = "POST"; // для отправки используется метод Post
                           // данные для отправки
    string data = "id=" + id;
    // преобразуем данные в массив байтов
    byte[] byteArray = System.Text.Encoding.UTF8.GetBytes(data);
    // устанавливаем тип содержимого - параметр ContentType
    request.ContentType = "application/x-www-form-urlencoded";
    // Устанавливаем заголовок Content-Length запроса - свойство ContentLength
    request.ContentLength = byteArray.Length;
    //записываем данные в поток запроса
    using (Stream dataStream = request.GetRequestStream())
    {
        dataStream.Write(byteArray, 0, byteArray.Length);
    }
    WebResponse response = request.GetResponse();
    string json_string = "";
    using (Stream stream = response.GetResponseStream())
    {
        using (StreamReader reader = new StreamReader(stream))
        {
            json_string += reader.ReadToEnd();
        }
    }
    my_One_Team = JsonConvert.DeserializeObject<My_One_Team>(json_string);
    response.Close();
}
```

Рисунок 50 – Метод `Get_Title_Team`

Метод позволяет получить название команды по ее уникальному идентификатору, в качестве единственного параметра он принимает идентификатор искомой команды. После этого формируется HTTP-запрос типа POST на серверную сторону мобильного приложения, после его обработки, сервер возвращает ответ в виде последовательности битов, которую клиентская сторона принимает в строковую переменную, а после подвергает операции десериализации при помощи ранее описанного метода `DeserializeObject()`. Итогом такого алгоритма действий является получение названия команды, что и требовалось получить.

Так же клиентская часть мобильного приложения содержит в своем составе классы, задачей которых является отображение содержимого страниц мобильного приложения:

- `Page_Autorization.xaml.cs` – страница авторизации приложения;
- `Page_Organizator.xaml.cs` – страница организатора;
- `Page_Organizator_Tournament.xaml.cs` – страница добавления турнира;
- `Page_Organizator_Team.xaml.cs` – страница добавления команды;
- `Page_Organizator_Player.xaml.cs` – страница добавления игрока;
- `Page_Organizator_Generic_Calendar.xaml.cs` – страница генерации календаря турнира;
- `Page_Edit_Date_Calendar.xaml.cs` – страница редактирования календаря турнира;
- `MainPage.xaml.cs` – главная страница обычного пользователя;
- `Page_FTournaments.xaml.cs` – страница с турнирами;
- `Page_Table.xaml.cs` – информационная страница по турниру;
- `Page_Team.xaml.cs` – информационная страница по команде;
- `Page_Player.xaml.cs` – информационная страница по игроку;
- `Page_Game.xaml.cs` – информационная страница по матчу.

На рисунке 51 представлена диаграмма компонентов мобильного приложения.

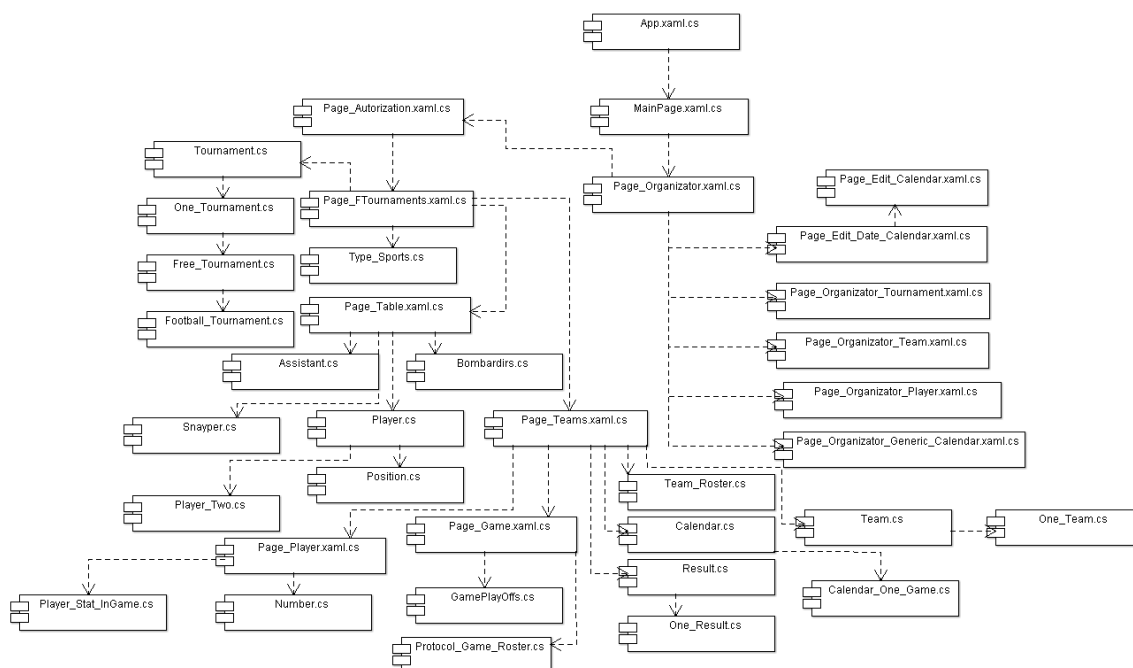


Рисунок 51 – Диаграмма компонентов мобильного приложения

На рисунке 52 представлена диаграмма развертывания, отображающая физическое расположение компонентов системы.

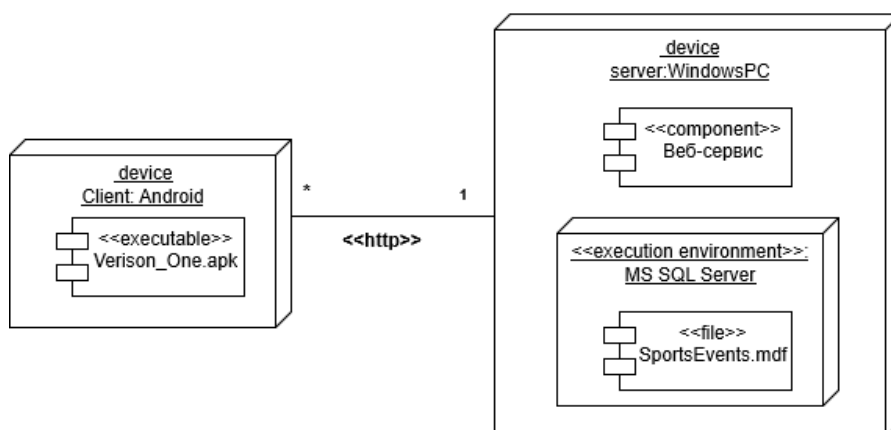


Рисунок 52 – Диаграмма развертывания мобильного приложения

Диаграмма отображает, что клиентское приложение располагается на Android-устройстве. API приложения и база данных располагаются на сервере приложения. Стоит отметить, что существует возможность множественного подключения клиентских приложений с разных Android-устройств к одному серверу.

4 Тестирование программного продукта

4.1 Тестирование серверной части приложения

При помощи инструментального средства Postman убедимся в корректности работы разработанного веб-сервиса.

Рассмотрим примеры некоторых возможностей веб-сервиса. Воспользуемся функцией получения информации о турнирах по конкретному виду спорта. В качестве передаваемых данных в запрос передадим идентификатор вида спорта (1-Футбол). Разработанный веб-сервис посылает ответ, содержащий список всех турниров по спортивной дисциплине «футбол» в формате JSON. Код ответа – 200 OK, что означает что запросом выполнен успешно (рисунок Г.1).

Воспользуемся возможностью веб-сервиса предоставить информацию о тех турнирах, для которых еще не сгенерирован календарь. Веб-сервис возвращает ответ, содержащий список таких турниров. Формат данных – JSON, код ответа – 200 OK (рисунок Г.2).

При помощи функции веб-сервиса «Создание нового турнира» добавим информацию о новом турнире. В качестве передаваемых данных в запрос передадим: название турнира – Кубок Быстрова, описание турнира – Ежегодный турнир среди детей 10-11 лет, идентификатор спорта – 1, формат турнира – Таблица, время одного тайма – 10 минут, количество команд – 5. Веб-сервис успешно добавляет данные о чем свидетельствует код ответа – 201, сообщающий о том, что создан новый ресурс. (рисунок Г.3).

Используя метод получения составов команд в каком-либо турнире разработанного веб-сервиса, отправим запрос на получение составов определенных команд. Передаваемыми параметрами будут выступать: идентификатор первой команды – 1, идентификатор второй команды – 2, идентификатор турнира – 1. Веб-сервис возвращает данные в формате JSON, код ответа – 200 OK (рисунок Г.4).

Получим название команды по ее идентификатору. Единственным передаваемым параметром будет являться идентификатор команды – 20, ответ веб-сервиса – 200 ОК. Данные успешно приняты клиентом в формате JSON (рисунок Г.5).

Полное тестирование всех функциональных возможностей разработанного веб-сервиса представлено в таблице 9.

Таблица 9 – Результаты тестирования веб-сервиса

№	Описание теста	Запрос		Результат
		Метод	Параметры	
1	Получение списка игр плей-офф первой стадии по турниру с идентификатором 21	POST	Stage=1, ID_Tournament = 21	Тест пройден. Получен список игр плей-офф первой стадии. Код ответа HTTP – 200.
2	Добавление команды для участия в серии плей-офф в турнир с идентификатором 21	POST	Stage = 1, ID_Team = 20, ID_Tournament = 21	Тест пройден. Создан новый ресурс. Код ответа HTTP – 201.
3	Обновление даты проведения матча у игры с идентификатором 18	POST	Date1 = 12.06.2020, Date2 = 18:30, ID_Game = 18	Тест пройден. Данные обновлены. Код ответа HTTP – 200.
4	Получение даты матча с идентификатором 20	POST	ID_Game = 20	Тест пройден. Получена дата матча. Код ответа HTTP – 200.
5	Получение турниров, в которых возможно редактировать календарь	POST		Тест пройден. Получен список турниров. Код ответа HTTP – 200.

Продолжение таблицы 9

№	Описание теста	Запрос		Результат
		Метод	Параметры	
6	Получение идентификатора команды	POST		Тест пройден. Получен идентификатор команды. Код ответа HTTP – 200.
7	Добавление новой игры плей-офф для команд с идентификаторами 20 и 21	POST	Id_team1 = 20, Id_team2 = 21, Date = 13.06.2020, Stage = 1, ID_Tournament = 21	Тест пройден. Создан новый ресурс. Код ответа HTTP – 201.
8	Добавление нового матча в календарь для команд с идентификаторами 1 и 5	POST	Id_team1 = 1, Id_team2 = 5, Date = 13.06.2020 18:30:00, ID_Tournament = 1	Тест пройден. Создан новый ресурс. Код ответа HTTP – 201.
9	Получение списка турниров, для которых не составлены календари	POST		Тест пройден. Получен список турниров. Код ответа HTTP – 200.
10	Добавление нового игрока Алексея Егорова в команду с идентификатором 30	POST	Id_Team = 30, Player_number = 98, date_birth = 12.10.1998, name = Алексей Егоров, id_position = 5	Тест пройден. Создан новый ресурс. Код ответа HTTP – 201.
11	Получение списка доступных номеров в команде с идентификатором 30	POST	ID_Team = 30	Тест пройден. Получен список свободных номеров в команде. Код ответа HTTP – 200.

Продолжение таблицы 9

№	Описание теста	Запрос		Результат
		Метод	Параметры	
12	Получение списка позиций игрока по виду спорта с идентификатором 3	POST	ID_sport = 3	Тест пройден. Получен список позиций. Код ответа HTTP – 200.
13	Получение списка команд по турниру с идентификатором 3	POST	ID_Tournament = 3	Тест пройден. Получен список команд. Код ответа HTTP – 200.
14	Получение списка турниров	POST		Тест пройден. Получен список турниров. Код ответа HTTP – 200.
15	Добавление новой команды Кристалл в турнир с идентификатором 18	POST	Title = Кристалл, ID_sport = 2, ID_Tournament = 18	Тест пройден. Создан новый ресурс. Код ответа HTTP – 201.
16	Получение списка турниров, в которые можно добавить команды	POST		Тест пройден. Получен список турниров. Код ответа HTTP – 200.
17	Добавление нового турнира Кубок пяти	POST	Title = Кубок пяти, description = ежегодный турнир, ID_sport = 2, time_half = 11, format = Плей-офф, count_teams = 5	Тест пройден. Создан новый ресурс. Код ответа HTTP – 201.
18	Получение списка снайперов по турниру с идентификатором 2	POST	ID_Tournament = 2	Тест пройден. Получен список снайперов.

Продолжение таблицы 9

№	Описание теста	Запрос		Результат
		Метод	Параметры	
19	Получение списка ассистентов турниру с идентификатором 3	POST	ID_Tournament = 3	Тест пройден. Получен список ассистентов. Код ответа HTTP – 200.
20	Получение списка бомбардиров по турниру с идентификатором 4	POST	ID_Tournament = 4	Тест пройден. Получен список ассистентов. Код ответа HTTP – 200.
21	Обновление статистических данных у игрока с идентификатором 30	POST	Id_player = 30, ID_tournament = 3	Тест пройден. Данные обновлены. Код ответа HTTP – 200.
22	Добавление результата игры между командами с идентификаторами 1 и 2	POST	Id_one_team = 1, Id_two_team = 2, Date = 10.05.2020, Score = 2-1	Тест пройден. Создан новый ресурс. Код ответа HTTP – 201.
23	Добавление события матча в игру с идентификатором 33	POST	Time = 3, ID_event = 5, Id_first = 33, Id_second = 70, Id_third = 66, Id_team = 30, Id_results = 33, Total = 2-0	Тест пройден. Создан новый ресурс. Код ответа HTTP – 201.
24	Добавление стартовых составов в игре с идентификатором 55 между командами с идентификаторами 19 и 33	POST	Id_player = 40, id_position = 3, Id_team = 19, Id_result = 55, Id_tournament = 2	Тест пройден. Создан новый ресурс. Код ответа HTTP – 201.
25	Получение статуса авторизации пользователя	POST	Login_ = Organizer, password_ = o	Тест пройден. Пользователь зарегистрирован в системе.

Продолжение таблицы 9

№	Описание теста	Запрос		Результат
		Метод	Параметры	
26	Получение составов команд с идентификаторами 30 и 40	POST	Id_team1 = 30, Id_team2 = 40, ID_Tournament = 6	Тест пройден. Получены списки с составами. Код ответа HTTP – 200.
27	Получение названия команды по идентификатору 1	POST	Id = 1	Тест пройден. Получено название команды. Код ответа HTTP – 200.
28	Получение результата игры по идентификатору 33	POST	Id = 33	Тест пройден. Получена информация о результате встречи. Код ответа HTTP – 200.
29	Получение названия турнира по идентификатору 3	POST	Id = 3	Тест пройден. Получено название турнира. Код ответа HTTP – 200.
30	Получение протокола матча по идентификатору 33	POST	Id = 33	Тест пройден. Получен протокол матча. Код ответа HTTP – 200.
31	Получение списка играющих составов в игре с идентификатором 33	POST	Id = 33	Тест пройден. Получен список играющих команд. Код ответа HTTP – 200.

Продолжение таблицы 9

№	Описание теста	Запрос		Результат
		Метод	Параметры	
32	Получение списка последних матчей у игрока с идентификатором 100	POST	ID_player = 100, ID_Tournament = 6	Тест пройден. Получен список последних матчей. Код ответа HTTP – 200.
33	Получение информации об игроке с идентификатором 100	POST	ID_player = 100	Тест пройден. Получена информация об игроке. Код ответа HTTP – 200.
34	Получение результатов команды с идентификатором 55	POST	ID_Team = 55, ID_Tournament = 6	Тест пройден. Получены результаты команды. Код ответа HTTP – 200.
35	Получение календаря команды с идентификатором 55	POST	ID_Team = 55, ID_Tournament = 6	Тест пройден. Получен календарь команды. Код ответа HTTP – 200.
36	Получение состава команды по идентификатору 40	POST	ID_Team = 40	Тест пройден. Получен список с составом команды. Код ответа HTTP – 200.
37	Получение списка бомбардиров по идентификатору турнира 1	POST	ID_Tournament = 1	Тест пройден. Получен список бомбардиров. Код ответа HTTP – 200.

Продолжение таблицы 9

№	Описание теста	Запрос		Результат
		Метод	Параметры	
38	Получение списка ассистентов по идентификатору турнира 1	POST	ID_Tournament = 1	Тест пройден. Получен список ассистентов. Код ответа – 200.
39	Получение списка снайперов по идентификатору турнира 1	POST	ID_Tournament = 1	Тест пройден. Получен список снайперов. Код ответа – 200.
40	Получение списка результатов турнира с идентификатором 1	POST	ID_Tournament = 1	Тест пройден. Получен список результатов игр. Код ответа HTTP – 200.
41	Получение календаря турнира с идентификатором 1	POST	ID_Tournament = 1	Тест пройден. Получен список с календарем турнира. Код ответа HTTP – 200.
42	Получение турнирной таблицы турнира с идентификатором 1	POST	ID_Tournament= 1	Тест пройден. Получен список с турнирной таблицей турнира. Код ответа HTTP – 200.
43	Получение списка турниров с идентификатором вида спорта 1	POST	ID = 1	Тест пройден. Получен список турниров. Код ответа HTTP – 200.

Окончание таблицы 9

№	Описание теста	Запрос		Результат
		Метод	Параметры	
44	Получение списка видов спорта	POST		Тест пройден. Получен список видов спорта. Код ответа HTTP – 200.

Все тесты API серверной части были пройдены успешно, что свидетельствует о корректности его работы.

4.2 Тестирование клиентской части приложения

4.2.1 Функциональное тестирование

Для проверки корректности работы клиентской части приложения было проведено функциональное тестирование. Тестирование проводилось с помощью Android-устройства Samsung Galaxy A5. Результаты функциональных тестов представлены в таблице 10.

Таблица 10 – Результаты тестирования клиентской части приложения

№ п/п	Вариант использования	Описание теста	Результаты теста
1	Просмотреть таблицу турнира	Перейти на вкладку Таблица в открытом окне Турнира «Городская футбольная лига»	Тест выполнен. Таблица выведена на экран (рисунок Д.1).
2	Просмотреть календарь турнира	Перейти на вкладку Календарь в открытом окне Турнира «Городская футбольная лига»	Тест выполнен. Календарь турнира выведен на экран (рисунок Д.2).
3	Просмотреть результаты матчей турнира	Перейти на вкладку Результаты в открытом окне Турнира «Городская футбольная лига»	Тест выполнен. Результаты матчей турнира выведены на экран (рисунок Д.3).
4	Просмотреть списки лучших игроков турнира	Перейти на вкладку Снайперы в открытом окне Турнира «Городская футбольная лига»	Тест выполнен. Список снайперов турнира выведен на экран (рисунок Д.4).
		Перейти на вкладку Ассистенты в открытом окне Турнира «Городская футбольная лига»	Тест выполнен. Список ассистентов выведен на экран (рисунок Д.5).

Продолжение таблицы 10

№ п/п	Вариант использования	Описание теста	Результаты теста
	Просмотреть списки лучших игроков турнира	Перейти на вкладку Бомбардиры в открытом окне Турнира «Городская футбольная лига»	Тест выполнен. Список бомбардиров выведен на экран (рисунок Д.6).
5	Просмотреть информацию о команде	В открытом окне Турнира выбрать строку с командой «Гранит»	Тест выполнен. Открыта информационная страница по команде «Гранит». На экране отображается состав команды (рисунок Д.7).
		В открытом окне информационная страница по команде «Гранит» открыть вкладку Календарь	Тест выполнен. Открыт календарь на оставшиеся матчи для команды «Гранит» (рисунок Д.8).
		В открытом окне информационная страница по команде «Гранит» открыть вкладку Результаты	Тест выполнен. Открыты результаты матчей команды «Гранит» (рисунок Д.9).
6	Просмотреть протокол матча	В открытом окне Турнира на вкладке Результаты выбрать строку с матчем «Гранит» - «Кубань»	Тест выполнен. Открыта информационная страница матча, отображающая основные события матча (рисунок Д.10).
		В открытом окне информационной страницы матча «Гранит» - «Кубань» выбрать вкладку составы	Тест выполнен. Открыты составы команд на матч «Гранит» - «Кубань» (рисунок Д.11).
7	Добавить турнир	В открытом окне Организатора турниров выбрать опцию «Добавить турнир». Заполнить информацию о добавляемом турнире	Тест выполнен. Турнир добавлен (рисунок Д.12).
8	Добавить команду в турнир	В открытом окне Организатора турниров выбрать опцию «Добавить команду». Ввести название команды – «Отдел 22», указать турнир – «Хоккейный турнир»	Тест выполнен. Команда «Отдел 22» добавлена в турнир «Хоккейный турнир» (рисунок Д.13, рисунок Д.14).
9	Добавить игрока в команду	В открытом окне Организатора турниров выбрать опцию «Добавить игрока». Ввести имя и фамилию игрока – «Васин Егор», указать команду – «Отдел 22»	Тест выполнен. Игрок Егор Васин добавлен в команду Отдел 22 (рисунок Д.15, рисунок Д.16).

Окончание таблицы 10

10	Редактировать календарь турнира	В открытом окне Организатора турниров выбрать опцию «Редактировать календарь турнира», в качестве турнира выбрать Городскую футбольную лигу, редактируемый матч – «Луч-Энергия» - «Кубань». Изменить время матча на 19:40.	Тест выполнен. Время матча изменено (рисунок Д.17, рисунок Д.18, рисунок Д.19).
11	Авторизоваться в приложении	В главном окне приложения ввести Логин – «Org» и пароль «о»	Тест выполнен. Открыт личный кабинет организатора турниров (рисунок Д.20, рисунок Д.21).
12	Заполнить составы команд на матч	Авторизоваться в приложении как секретарь, выбрать турнир «Кубок Петрова», на вкладке Календарь выбрать матч «Кристалл» - «Варяги», при помощи переключателей указать составы команд	Тест выполнен. Составы команд указаны, открыта возможность заполнения протокола матча (рисунок Д.22).
13	Заполнить протокол матча	После заполнения составов команд на матч, выбрать вкладку «Заполнить протокол», указывать события матча	Тест выполнен, события матча сохранены, результаты сохранены (рисунок Д.23, рисунок Д.24, рисунок Д.25, рисунок Д.26).

Функциональное тестирование показало корректность работы клиентской части мобильного приложения.

4.3 Метрики кода

Метрикой программного кода называется количественная мера, позволяющая определить, в какой степени система, компоненты или процесс, обладают заданным атрибутом. Все метрики программного кода можно разделить на группы:

- метрики размера программы (SLOC-метрики);
- метрики сложности потока управления программ (метрики Холстеда и цикломатическая сложность).

Для программного кода, изображенного на рисунке 53 рассчитаем метрики программного кода.

```

//Получение турнирной таблицы.
Ссылка 0
public string Return_Table_Tournament(string id)
{
    List<Football_Tournament> list_table = new List<Football_Tournament>();
    string conn = "Data Source =DESKTOP-SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated Security=True;";
    SqlConnection sqlConnection;
    SqlCommand command;
    SqlDataReader reader;
    sqlConnection = new SqlConnection(conn);
    sqlConnection.Open();
    command = new SqlCommand();
    command.CommandText = "SELECT ID_Team, Games, Wins, Points, ID_tournament from [Football_Tournament] where ID_tournament = " + id +
        " order by Points DESC, Wins ASC, Games ASC";
    command.Connection = sqlConnection;
    reader = null;
    reader = command.ExecuteReader();
    while (reader.Read())
    {
        Football_Tournament football_Tournament = new Football_Tournament();
        football_Tournament.ID_Team = Convert.ToInt32(reader["ID_Team"].ToString());
        football_Tournament.Games = Convert.ToInt32(reader["Games"].ToString());
        football_Tournament.Wins = Convert.ToInt32(reader["Wins"].ToString());
        football_Tournament.Points = Convert.ToInt32(reader["Points"].ToString());
        football_Tournament.ID_tournament = Convert.ToInt32(reader["ID_tournament"].ToString());
        list_table.Add(football_Tournament);
    }
    reader.Close();
    command = null;
    MyFootballTournament myFootballTournament = new MyFootballTournament();
    myFootballTournament.F_Tournaments = new List<Football_Tournament>();
    foreach (var team in list_table)
    {
        command = new SqlCommand();
        command.CommandText = "SELECT Title from [Team] where ID = " + team.ID_Team + ";";
        command.Connection = sqlConnection;
        reader.Close();
        reader = command.ExecuteReader();
        while (reader.Read())
        {
            team.Team = reader["Title"].ToString();
        }

        myFootballTournament.F_Tournaments.Add(team);
    }
    sqlConnection.Close();
    string serialized = JsonConvert.SerializeObject(myFootballTournament);
    return serialized;
}

```

Рисунок 53 – Метод получения информации о турнирной таблице

Прежде всего, стоит сказать, что данный метод расположен на серверной части мобильного приложения и позволяет получить информацию каждой команды в турнирной таблице – количество сыгранных матчей, число побед, количество набранных очков. В качестве входного параметра метод принимает идентификатор турнира, информацию о турнирной таблице которого необходимо получить. Далее формируется запросы к базе данных для получения необходимых данных.

4.3.1 SLOC-метрика

SLOC-метрика предназначена для подсчета физических и логических строк программного кода, к физическим строкам относят все непустые строки программного файла, к логическим – число строк с командами, используемыми в программном коде. Так же SLOC-метрика предполагает вычисление строк с комментариями.

Вычислим SLOC-метрику для программного кода, приведенного на рисунке 50. Результаты вычисления приведены в таблице 11.

Таблица 11 – Вычисление SLOC-метрики

Вид строк	Количество
Физические строки	48
Логические строки	27
Строки-комментарии	1

Количество логических строк у одной и той же программы, выполненной на разных языках программирования, будет различным не только потому что размер файла будет отличаться, но и потому, что определение команды зависит от конкретного языка программирования.

4.3.2 Метрики Холстеда

Метрики Холстеда включают четыре измеряемые характеристики программы:

- число уникальных операторов программы;
- число уникальных операндов программы;
- общее число операторов программы;
- общее число операндов программы.

На основе метрик Холстеда можно получить следующие количественные оценки:

- словарь программы = число уникальных операторов + число уникальных операндов;
- длина программы = общее число операторов + общее число операндов;
- объем программы = длина программы * $\log_2$ словарь программы;
- сложность программы = $(\text{число уникальных операторов}/2) * (\text{общее число операндов}/\text{число уникальных операндов})$.

Получим метрики Холстеда для программного кода, изображенного на рисунке 50. В таблицах 12 и 13 отображены уникальные операторы программного кода и уникальные операнды.

Таблицы 12 и 13 показывают, что общее число операторов программного кода = 7, а общее число операндов = 12.

Таблица 12 – Уникальные операторы

Номер	Оператор	Количество
1	=	21
2	+	4
3	;	33
4	(	33
5	)	33
6	new	7
7	Convert.ToInt32	5
8	foreach	1

Таблица 13 – Уникальные операнды

1	list_table	3
2	conn	2
3	sqlConnection	6
4	command	10
5	reader	14
6	command.CommandText	2
7	command.Connection	2
8	myFootballTournament.F_Tournaments	2
9	team	1
10	team.Team	1
11	myFootballTournament	2

Окончание таблицы 13

Номер	Оператор	Количество
12	serialized	2
13	Team.ID_Team	1

Рассчитаем словарь программы, который вычисляется как сумма уникальных операторов и уникальных операндов программы: словарь программы = $8+13 = 21$.

Рассчитаем длину программы, равную сумме всех операторов программы и всех операндов: длина программы = 185.

Рассчитаем объем программы: $185 * \log_2 21 = 812,52$.

Рассчитаем сложность программы: $(8/2) * (48/13) = 14,76$.

Полученные характеристики показывают, что программный код имеет высокий уровень качества и позволяет с ним работать в дальнейшем. Показатель сложности программы находится в допустимых границах, что означает, что программный код имеют среднюю сложность поддержки.

Заключение

В рамках выполнения выпускной квалификационной работы было разработано мобильное приложение для управления спортивными мероприятиями.

На этапе анализа требований был проведен сравнительный анализ мобильных приложений-аналогов и выделены их слабые и сильные стороны, выделены функциональные и нефункциональные требования к разрабатываемому приложению, проведен анализ целевой аудитории разрабатываемого приложения.

Было осуществлено планирование разработки мобильного приложения, оценена стоимость такой разработки. Планирование показало, что срок реализации проекта составит 45 дней, а оценка стоимости проекта составила 57 480 рублей.

На этапе проектирования приложения была составлена логическая модель базы данных, выстроена архитектура разрабатываемого приложения. Серверная часть приложения была разработана в среде Microsoft Visual Studio 2019 на языке программирования C#. Клиентская часть была разработана так же в среде программирования Microsoft Visual Studio 2019 на языке программирования C# при помощи технологии Xamarin.Forms. Для реализации базы данных использовалась СУБД MS SQL Server 2016.

Было проведено тестирование серверной и клиентской части разработанного программного продукта, которое показало корректность работы всего приложения.

В дальнейшем планируется добавление новых видов спорта, размещение серверной части в облачном сервисе Heroku и размещение разработанного программного продукта в магазине приложений Google Play.

Таким образом, поставленная задача по реализации мобильного приложения для управления спортивными мероприятиями была выполнена в полном объеме.

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: 23.03.2020).
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: 23.03.2020).
3. Власов, А.Е. Развитие массового спорта в современной России. Монография / Власов А.Е., Зеленков Ю.А., Солнцев И.В. – Проспект, 2018. – 127 с.
4. Спорт как инструмент развития личности ребенка [Электронный ресурс]. – URL: <https://moluch.ru/th/2/archive/104/3645/> (дата обращения: 19.05.2020)
5. Школьный спорт. Как правильно его организовывать? [Электронный ресурс]. – URL: <http://www.ug.ru/article/217> (дата обращения: 19.05.2020)
6. Корпоративный спорт на промышленном предприятии [Электронный ресурс]. – URL: <https://allrefrs.ru/2-9068.html> (дата обращения: 19.05.2020)
7. Распоряжение правительства РФ от 03.06.2019 «Об утверждении Стратегии развития спортивной индустрии до 2035 года» № 1188 [Электронный ресурс]. – URL: http://www.consultant.ru/document/cons_doc_LAW_326728/ (дата обращения: 23.03.2020).
8. О состоянии спортивной инфраструктуры в России // Сайт спортивной направленности [Электронный ресурс]. – URL:

<http://sportfiction.ru/articles/o-sostoyanii-sportivnoy-infrastruktury-v-rossii/> (дата обращения: 25.03.2020)

9. Значимость спортивных физкультурно-массовых мероприятий в формировании здорового образа жизни студентов [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/znachimost-sportivnyh-fizkulturno-massovyh-meropriyatiy-v-formirovanii-zdorovogo-obraza-zhizni-studentov> (дата обращения: 19.05.2020)

10. Winner-Создание турниров // Магазин приложений PlayMarket [Электронный ресурс]. – URL: <https://play.google.com/store/apps/details?id=il.talent.winner&hl=ru> (дата обращения: 23.03.2020)

11. Мобильное приложение League App // Магазин приложений PlayMarket [Электронный ресурс]. – URL: <https://play.google.com/store/apps/details?id=com.mileyenda.manager&hl=ru> (дата обращения 20.03.2020)

12. Международный симулятор // Магазин приложений PlayMarket [Электронный ресурс]. – URL: <https://play.google.com/store/apps/details?id=com.Senapp.FC2018S&hl=ru> (дата обращения: 22.03.2020)

13. В каких странах IOs популярнее Android // Новостной сайт о мобильных разработках [Электронный ресурс]. – URL: <https://www.iphones.ru/iNotes/v-kakih-stranah-ios-populyarnее-android-09-13-2019> (дата обращения: 27.03.2020)

14. Статистика рынка смартфонов // Новостной сайт об информационных технологиях [Электронный ресурс]. – URL: https://www.tadviser.ru/index.php/Продукт:Google_Chrome?cache=no&pptype=news (дата обращения: 27.03.2020)

15. Статистика версий Android // Сайт о технике, об информационных технологиях [Электронный ресурс]. – URL: <https://4pda.ru/2020/04/11/370395/> (дата обращения: 28.03.2020)

16. Сравниваем цены на устройства IOs и Android // Сайт о мобильных телефонах и мобильной связи [Электронный ресурс]. – URL: <http://mobinfo.uz/12834-sravnivaem-srednie-ceny-na-ustrojjstva-pod-2.html> (дата обращения: 30.03.2020)
17. Что лучше Android или IOs – ставим точку в извечном вопросе // Портал о технике [Электронный ресурс]. – URL: <http://mobinfo.uz/12834-sravnivaem-srednie-ceny-na-ustrojjstva-pod-2.html> (дата обращения: 30.03.2020)
18. Арлоу, Д. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2е издание / Д. Арлоу, И. Нейштадт – СПб: Символ Плюс, 2007. – 624 с.
19. Статья на Wikipedia о UML // Электронная энциклопедия [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/UML> (дата обращения: 31.03.2020)
20. Основы управления проектами : [учеб. пособие] / Л. Н. Боронина, З. В. Сенук ; М-во образования и науки рос. Федерации, Урал. Федер. Ун-т. – Екатеринбург : изд-во Урал. ун-та, 2015. — 112 с.
21. Управление проектами и рисками // Ю.Ш. Сиразитдинова, Томский Политехнический Университет – 52 с.
22. Шкрыль А.А. MS Project 2007. Современное управление проектами. БХВ-Петербург, 2008. – 256 с.
23. Технология RUP // Новостной сайт об информационных системах [Электронный ресурс]. – URL: https://wiki.clan.su/publ/primery_ts_po_tekhnologija_rup_rational_unified_process/1-1-0-39 (дата обращения: 30.03.2020)
24. Статья на Wikipedia об архитектуре клиент-сервер // Электронная энциклопедия [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Клиент_—_сервер (дата обращения: 31.03.2020)
25. Статья на Wikipedia о формате данных JSON // Электронная энциклопедия [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/JSON> (дата обращения: 10.04.2020)

26. Сериализация-десериализация JSON в C# // Учебный портал, направленный на обучение языкам программирования [Электронный ресурс].
- URL: https://котодомик.рф/2015/02/18/json_csharp/ (дата обращения: 12.04.2020)

Приложение А.
ГЛОССАРИЙ
(справочное)

iOS – мобильная операционная система для смартфонов, электронных планшетов и других устройств, разрабатываемая и выпускаемая американской компанией Apple.

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

RUP (Rational Unified Process) – методология разработки программного обеспечения, которая использует итеративную модель разработки.

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».

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

Приложение Б.
ИСХОДНЫЙ КОД СЕРВЕРНОЙ ЧАСТИ
(рекомендуемое)

Листинг Б.1 – Файл Assistant.cs

```
public class Assistant
{
    public int ID_player { get; set; }
    public string Player { get; set; }
    public int Games { get; set; }
    public int Assistans { get; set; }
    public int ID_tournament { get; set; }
}
```

Листинг Б.2 – Файл Bombardirs.cs

```
public class Bombardirs
{
    public int ID_player { get; set; }
    public string Player { get; set; }
    public int Games { get; set; }
    public int Goals { get; set; }
    public int Assistans { get; set; }
    public int Points { get; set; }
    public int ID_tournament { get; set; }
}
```

Листинг Б.3 – Файл Calendar.cs

```
public class Calendar
{
    public int ID { get; set; }
    public int ID_team1 { get; set; }
    public string Team1 { get; set; }
    public int ID_team2 { get; set; }
    public string Team2 { get; set; }
    private string date;
    private string stage;
    public string Date
```

```

    {
        get
        {
            return date;
        }
        set
        {
            date = value.Substring(0, 5);
        }
    }
    public string Stage
    {
        get
        {
            return stage;
        }
        set
        {
            if (value == "1")
                stage = "1/8 финала";
            else if (value == "2")
                stage = "1/4 финала";
            else if (value == "3")
                stage = "1/2 финала";
            else if (value == "4")
                stage = "Финал";
        }
    }
    public int ID_tournament { get; set; }
}

```

Листинг Б.4 – Файл Calendar_One_Game.cs

```

public class Calendar_One_Game

```

```

{
    public int ID { get; set; }
    public int ID_team1 { get; set; }
    public string Team1 { get; set; }
    public int ID_team2 { get; set; }
    public string Team2 { get; set; }
    private string date;
    private string stage;
    public string Date
    {
        get
        {
            return date;
        }
        set
        {
            date = value.Substring(0, value.Length - 3);
        }
    }
    public string Stage
    {
        get
        {
            return stage;
        }
        set
        {
            if (value == "1")
                stage = "1/8 финала";
            else if (value == "2")
                stage = "1/4 финала";
            else if (value == "3")

```

```

        stage = "1/2 финала";
    else if (value == "4")
        stage = "Финал";
    }
}

public int ID_tournament { get; set; }
}

```

Листинг Б.5 – Файл Football_Tournament.cs

```

public class Football_Tournament
{
    public int ID_Team { get; set; }
    public string Team { get; set; }
    public int Games { get; set; }
    public int Wins { get; set; }
    public int Points { get; set; }
    public int ID_tournament { get; set; }

}

```

Листинг Б.6 – Файл Free_Tournament.cs

```

public class Free_Tournament
{
    public int ID { get; set; }
    public string Title { get; set; }
    public int Sport_ID { get; set; }
    public string format_tournament { get; set; }
    public int count_teams { get; set; }

}

```

Листинг Б.7 – Файл GamePLayOffs.cs

```

public class GamePlayOffs
{
    public int ID { get; set; }
}

```

```

        public int Team_ID { get; set; }
        public int Stage { get; set; }
        public int Tournament_ID { get; set; }
    }

```

Листинг Б.8 – Файл Number.cs

```

public class Number
{
    public int Numb { get; set; }
}

```

Листинг Б.9 – Файл One_Result.cs

```

public class One_Result
{
    public string ID_team1 { get; set; }
    public string ID_team2 { get; set; }
    private string stage;
    public string Stage
    {
        get
        {
            return stage;
        }
        set
        {
            if (value == "1")
                stage = "1/8 финала";
            else if (value == "2")
                stage = "1/4 финала";
            else if (value == "3")
                stage = "1/2 финала";
            else if (value == "4")
                stage = "Финал";
        }
    }
}

```

```

    }
}

```

Листинг Б.10 – Файл One_Team.cs

```

public class One_Team
{
    public string Title { get; set; }
    public int ID { get; set; }
}

```

Листинг Б.11 – Файл One_Tournament.cs

```

public class One_Tournament
{
    public string Title { get; set; }
    public int Time_half { get; set; }
}

```

Листинг Б.12 – Файл Player.cs

```

public class Player
{
    public int ID { get; set; }
    public int ID_team { get; set; }
    public string Team { get; set; }
    public string Name { get; set; }
    public int ID_position { get; set; }
    public string Position { get; set; }
    private string age;
    public string Age
    {
        get
        {
            return age;
        }
        set

```

```

        {
            var current_Date = DateTime.Now;
            var player_Date = Convert.ToDateTime(value);
            int n = current_Date.Year - player_Date.Year;
            if (player_Date.DayOfYear >
current_Date.DayOfYear)
                --n;
            age = n.ToString();
        }
    }
    public int Player_number { get; set; }
}

```

Листинг Б.13 – Файл Player_Stat_inGame.cs

```

public class Player_Stat_inGame
{
    private string date;
    public string Date
    {
        get
        {
            return date;
        }
        set
        {
            string d = value.Substring(0, 5);
            date = d;
        }
    }

    public int ID_team_our { get; set; }
    public string Team_our { get; set; }
    public int ID_team { get; set; }
    public string Team { get; set; }
}

```

```

        public int Goals { get; set; }
        public int Assists { get; set; }
        public string Score { get; set; }
        public int ID_results { get; set; }
    }

```

Листинг Б.14 – Файл Position.cs

```

public class Position
{
    public int ID { get; set; }
    public string Pstion { get; set; }
}

```

Листинг Б.15 – Файл Protocol_Game.cs

```

public class Protocol_Game
{
    public int ID { get; set; }
    public int First_Player { get; set; }
    public int Second_Player { get; set; }
    public int Third_Player { get; set; }
    public string First_Player1 { get; set; }
    public string Second_Third_Player { get; set; }
    public int ID_event { get; set; }
    public int Time { get; set; }
    public int ID_team { get; set; }
}

```

Листинг Б.16 – Файл Protocol_Game_Roster.cs

```

public class Protocol_Game_Roster
{
    public int ID_player { get; set; }
    public string Player { get; set; }
    public int Number { get; set; }
    private string position;
}

```

```

public int ID_Position { get; set; }
public string Position
{
    get
    {
        return position;
    }
    set
    {
        if (value == "Вратарь")
            position = "Вр";
        else if (value == "Защитник")
            position = "Зщ";
        else if (value == "Полузащитник")
            position = "Пзщ";
        else if (value == "Нападающий")
            position = "Нап";
    }
}

public int ID_team { get; set; }
public string Team { get; set; }
//Для перехода на страницу игрока (Страница 5)
public int ID_tournament { get; set; }
}

```

Листинг Б.17 – Файл Result.cs

```

public class Result
{
    public int ID { get; set; }
    public int ID_team1 { get; set; }
    public string Team1 { get; set; }
    public int ID_team2 { get; set; }
    public string Team2 { get; set; }
}

```

```
private string date;
public string Date
{
    get
    {
        return date;
    }
    set
    {
        date = value.Substring(0, value.Length - 3);
    }
}
public string Score { get; set; }
private string stage;
public string Stage
{
    get
    {
        return stage;
    }
    set
    {
        if (value == "1")
            stage = "1/8 финала";
        else if (value == "2")
            stage = "1/4 финала";
        else if (value == "3")
            stage = "1/2 финала";
        else if (value == "4")
            stage = "Финал";
    }
}
```

```

        public int ID_tournament { get; set; }
    }

```

Листинг Б.18 – Файл Snayper.cs

```

public class Snayper
{
    public int ID_player { get; set; }
    public string Player { get; set; }
    public int Games { get; set; }
    public int Goals { get; set; }
    public int ID_tournament { get; set; }
}

```

Листинг Б.19 – Файл Team.cs

```

public class Team
{
    public int ID { get; set; }
    public string Title { get; set; }
    public int ID_sport { get; set; }
}

```

Листинг Б.20 – Файл Team_Roster.cs

```

public class Team_Roster
{
    public int ID_player { get; set; }
    public string Name { get; set; }
    public int Number { get; set; }
    private string position;
    public int ID_Position { get; set; }
    public string Position
    {
        get
        {
            return position;
        }
    }
}

```

```

    }
    set
    {
        if (value == "Вратарь")
            position = "Вр.";
        else if (value == "Защитник")
            position = "Зщ.";
        else if (value == "Полузащитник")
            position = "Пзщ.";
        else if (value == "Нападающий")
            position = "Нап.";
    }
}

public int ID_tournament { get; set; }
public int ID_team { get; set; }
public string Team { get; set; }
}

```

Листинг Б.21 – Файл Team_Roster.cs

```

public class Team_Roster
{
    public int ID_player { get; set; }
    public string Name { get; set; }
    public int Number { get; set; }
    private string position;
    public int ID_Position { get; set; }
    public string Position
    {
        get
        {
            return position;
        }
        set
    }
}

```

```

        {
            if (value == "Вратарь")
                position = "Вр.";
            else if (value == "Защитник")
                position = "Зщ.";
            else if (value == "Полузащитник")
                position = "Пзщ.";
            else if (value == "Нападающий")
                position = "Нап.";
        }
    }

    public int ID_tournament { get; set; }
    public int ID_team { get; set; }
    public string Team { get; set; }
}

```

Листинг Б.22 – Файл Tournament.cs

```

public class Tournament
{
    public int ID { get;set; }
    public string Title { get; set; }
    public string Description { get; set; }
    public int Sport_ID { get; set; }
    public int Time_half { get; set; }
}

```

Листинг Б.23 – Файл Type_Sports.cs

```

public class Type_sports
{
    public int ID { get; set; }
    public string Sport { get; set; }
}

```

Листинг Б.24 – Файл HomeController.cs

```

public class HomeController : Controller
{
    //PostData, PostData1 - запросы на получение видов спорта
    и турниров по нему.

    //PostData2 - PostData7 - запросы на получение таблицы,
    календаря, результаов, снайперов, асистентов, бомбардиров.

    //PostData8 - Получение списка игроков,
    PostData9,PostData10 - календарь Команды и результаты Команды (4-
    я Страница)

    //PostData11 - инфо об игроке . 5-я Страница. PostData12
    - последние игры игрока. 5-я Страница.

    // Функционал: Секретарь. Игры формата Плей-офф.
    Добавление новой игры плей-офф в Calendars

    [HttpPost]

    public void Insert_PlayOff_Calendars(string ID_team1,
    string ID_team2, string date, string ID_tournament, string stage)
    {
        string conn = "Data Source = DESKTOP-
        SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
        Security=True;";

        SqlConnection sqlConnection;
        SqlCommand command;

        sqlConnection = new SqlConnection(conn);
        sqlConnection.Open();

        command = new SqlCommand();

        command.CommandText = "INSERT INTO [Calendars] values
        ( " + ID_team1 + ", " + ID_team2 + ", '" + date + "', " +
        ID_tournament + ", " + "null, " + stage + " )";

        command.Connection = sqlConnection;

        command.ExecuteNonQuery();

        sqlConnection.Close();
    }

    // Функционал: Секретарь. Игры формата Плей-офф. Удаление
    2-х команд из вспомогательной таблицы Game_PlayOffs

    [HttpPost]

    public void Delete_From_GPOffs_Off(string ID)

```

```

    {
        string conn = "Data Source = DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

        SqlConnection sqlConnection;
        SqlCommand command;

        sqlConnection = new SqlConnection(conn);
        sqlConnection.Open();

        command = new SqlCommand();

        command.CommandText = "DELETE from [Games_PlayOff]
where ID = " + ID + ";

        command.Connection = sqlConnection;
        command.ExecuteNonQuery();

        sqlConnection.Close();
    }

    // Функционал: Секретарь. Игры формата Плей-офф.
Добавление записи в Games_PlayOff

    [HttpPost]

    public string Delete_From_GPOffs_Insert_Calendars(string
stage, int id_tournament)
    {
        int Stage = 0;

        //устанавливаем номер стадии, номер стадии++
        if (stage == "1/8 финала")
            Stage = 2;
        else if (stage == "1/4 финала")
            Stage = 3;
        else if (stage == "1/2 финала")
            Stage = 4;

        string conn = "Data Source = DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

        SqlConnection sqlConnection;

        SqlCommand command;

```

```

        SqlDataReader reader;

        sqlConnection = new SqlConnection(conn);
        sqlConnection.Open();

        command = new SqlCommand();

        command.CommandText = "SELECT * from [Games_PlayOff]
where Stage = " + Stage + " and Tournament_ID = " + id_tournament
+ "";

        command.Connection = sqlConnection;

        reader = command.ExecuteReader();

        My_Game_PlayOffs      my_Game_PlayOffs      =      new
My_Game_PlayOffs();

        my_Game_PlayOffs.GamePlayOffs      =      new
List<GamePlayOffs>();

        while (reader.Read())
        {

            GamePlayOffs gamePlayOffs = new GamePlayOffs();

            gamePlayOffs.ID = Convert.ToInt32(reader["ID"]);

            gamePlayOffs.Stage      =
Convert.ToInt32(reader["Stage"]);

            gamePlayOffs.Team_ID      =
Convert.ToInt32(reader["Team_ID"]);

            my_Game_PlayOffs.GamePlayOffs.Add(gamePlayOffs);

        };

        reader.Close();

        sqlConnection.Close();

        string      serialized      =
JsonConvert.SerializeObject(my_Game_PlayOffs);

        return serialized;

    }

    // Функционал: Секретарь. Игры формата Плей-офф.
Добавление записи в Games_PlayOff

    [HttpPost]

    public void Insert_Game_PlayOffs(int id_team, string
stage, int id_tournament)

    {

```

```

        int Stage=0;

        //устанавливаем номер стадии, номер стадии++
        if (stage == "1/8 финала")
            Stage = 2;
        else if (stage == "1/4 финала")
            Stage = 3;
        else if (stage == "1/2 финала")
            Stage = 4;

        string conn = "Data Source = DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

        SqlConnection sqlConnection;
        SqlCommand command;

        sqlConnection = new SqlConnection(conn);
        sqlConnection.Open();

        command = new SqlCommand();

        command.CommandText = "INSERT INTO [Games_PlayOff]
values ( " + id_team + ", " + Stage + ", " + id_tournament+" )" ;

        command.Connection = sqlConnection;
        command.ExecuteNonQuery();

        sqlConnection.Close();
    }

    // Функционал: Организатор. Изменение даты и времени
начала матча

    [HttpPost]

    public void Set_Date_In_Calendar(string date1, string
date2, string id_game)
    {

        string Date = date1.Substring(0, 10);

        Date += " ";

        Date += date2;

        string conn = "Data Source = DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

```

```

        SqlConnection sqlConnection;
        SqlCommand command;
        sqlConnection = new SqlConnection(conn);
        sqlConnection.Open();
        command = new SqlCommand();
        command.CommandText = "update Calendars set Date = '"
+ Date + "' where ID = " + id_game+"";
        command.Connection = sqlConnection;
        command.ExecuteNonQuery();
        sqlConnection.Close();
    }
    //Получение даты игры из календаря
    [HttpPost]
    public string Return_DateGame_From_Calendar(string ID)
    {
        string conn = "Data Source =DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

        SqlConnection sqlConnection;
        SqlCommand command;
        SqlDataReader reader;
        sqlConnection = new SqlConnection(conn);
        sqlConnection.Open();
        command = new SqlCommand();
        command.CommandText = "select Date from [Calendars]
where ID=" + ID + "";
        command.Connection = sqlConnection;
        reader = command.ExecuteReader();
        string Date = "";
        while (reader.Read())
        {
            Date = reader["Date"].ToString();
        }
    };

```

```

        reader.Close();

        sqlConnection.Close();

        return Date;
    }

    //Получение id турниров, в которых возможно редактировать
    календарь

    [HttpPost]

    public string Return_ID_Tournaments()
    {
        string conn = "Data Source =DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

        SqlConnection sqlConnection;
        SqlCommand command;
        SqlDataReader reader;

        sqlConnection = new SqlConnection(conn);
        sqlConnection.Open();

        command = new SqlCommand();

        command.CommandText = "select ID, Title, time_half
from [Tournament] where Exist_Calendar = 1";

        command.Connection = sqlConnection;

        reader = null;

        reader = command.ExecuteReader();

        //List<int> ID_tournaments = new List<int>();

        MyTournaments myTournaments = new MyTournaments();
        myTournaments.Tournaments = new List<Tournament>();

        while (reader.Read())
        {
            Tournament tournament = new Tournament();

            tournament.ID = (Convert.ToInt32(reader["ID"]));

            tournament.Title = reader["Title"].ToString();

            tournament.Time_half = Convert.ToInt32(
reader["time_half"].ToString());

            myTournaments.Tournaments.Add(tournament);
        }
    }

```

```

    };

    reader.Close();

    sqlConnection.Close();

    string serialized =
JsonConvert.SerializeObject(myTournaments);

    return serialized;
}

//Получение id команды
[HttpPost]
public string Return_ID_Team()
{
    string conn = "Data Source =DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

    SqlConnection sqlConnection;
    SqlCommand command;
    SqlDataReader reader;

    sqlConnection = new SqlConnection(conn);
    sqlConnection.Open();

    command = new SqlCommand();

    command.CommandText = "select * from [Team] where ID
= (SELECT max (ID) from [Team])";

    command.Connection = sqlConnection;

    reader = null;

    reader = command.ExecuteReader();

    My_One_Team my_One_Team = new My_One_Team();
    my_One_Team.one_Teams = new List<One_Team>();
    while (reader.Read())
    {
        One_Team one_Team = new One_Team();
        one_Team.Title = reader["Title"].ToString();
        one_Team.ID = Convert.ToInt32(reader["ID"]);
        my_One_Team.one_Teams.Add(one_Team);
    }
}

```

```

        };

        reader.Close();

        command = null;

        sqlConnection.Close();

        string serialized =
JsonConvert.SerializeObject(my_One_Team);

        return serialized;
    }

    // Функционал: Организатор. Установление Exist_Calendar =
1, связано с Insert_Calendars

    [HttpPost]

    public void Set_Exist(int id_tournament)
    {
        string conn = "Data Source = DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

        SqlConnection sqlConnection;
        SqlCommand command;

        sqlConnection = new SqlConnection(conn);
        sqlConnection.Open();

        command = new SqlCommand();

        command.CommandText = "update Tournament set
Exist_Calendar = 1 where ID = " + id_tournament+"";

        command.Connection = sqlConnection;

        command.ExecuteNonQuery();

        sqlConnection.Close();
    }

    // Функционал: Организатор. Добавление нового матча в
календарь. ФОРМАТ ПЛЕЙ-ОФФ

    [HttpPost]

    public void Insert_Calendars_Play_Off(int id_team1, int
id_team2, string Date, int id_tournament,int stage)
    {

```

```

        string conn = "Data Source =DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

        SqlConnection sqlConnection;

        SqlCommand command;

        sqlConnection = new SqlConnection(conn);

        sqlConnection.Open();

        command = new SqlCommand();

        command.CommandText = "INSERT INTO [Calendars] values
( " + id_team1 + ", " + id_team2 + ", '" + Date + "' , " +
id_tournament + ", null, " + stage + " )";

        command.Connection = sqlConnection;

        command.ExecuteNonQuery();

        sqlConnection.Close();

    }

    // Функционал: Организатор. Добавление нового матча в
    календарь

    [HttpPost]

    public void Insert_Calendars(int id_team1, int id_team2,
    string Date, int id_tournament)

    {

        string conn = "Data Source =DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

        SqlConnection sqlConnection;

        SqlCommand command;

        sqlConnection = new SqlConnection(conn);

        sqlConnection.Open();

        command = new SqlCommand();

        command.CommandText = "INSERT INTO [Calendars] values
( " + id_team1 + ", " + id_team2 + ", '" + Date + "' , " +
id_tournament + ", null, null)";

        command.Connection = sqlConnection;

        command.ExecuteNonQuery();

        sqlConnection.Close();

```

```

    }

    //Функционал: Организатор. Выдача в "Генерация календаря" только
    тех турниров, у которых нет календарей

    [HttpPost]

    public string Select_Free_Tournament_On_Generic()
    {
        string conn = "Data Source =DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

        SqlConnection sqlConnection;
        SqlCommand command;
        SqlDataReader reader;

        sqlConnection = new SqlConnection(conn);
        sqlConnection.Open();
        command = new SqlCommand();
        command.CommandText = "Select * from [Tournament]
where Exist_Calendar is NULL";
        command.Connection = sqlConnection;
        reader = command.ExecuteReader();

        My_Free_Tournament my_Free_Tournament = new
My_Free_Tournament();

        my_Free_Tournament.free_Tournaments = new
List<Free_Tournament>();

        while (reader.Read())
        {
            Free_Tournament free_Tournament = new
Free_Tournament();

            free_Tournament.ID =
Convert.ToInt32(reader["ID"]);

            free_Tournament.Title =
reader["Title"].ToString();

            free_Tournament.Sport_ID =
Convert.ToInt32(reader["Sport_ID"]);

            free_Tournament.format_tournament =
reader["format_tournament"].ToString();

```

```

        free_Tournament.count_teams
Convert.ToInt32(reader["count_teams"]);

my_Free_Tournament.free_Tournaments.Add(free_Tournament);
    }
    reader.Close();
    //команды по конкретному id турнира
    List<int> ID_temas_in_tourn = new List<int>();
    List<int> ID_teams_in_freedom_generic_calendars = new
List<int>();
    for (int i =
0;i<my_Free_Tournament.free_Tournaments.Count;i++)
    {
        ID_temas_in_tourn = new List<int>();
        command.CommandText = "Select ID from [Team] where
ID_tournament = " + my_Free_Tournament.free_Tournaments[i].ID +
"";

        reader = command.ExecuteReader();
        while (reader.Read())
        {
            ID_temas_in_tourn.Add(Convert.ToInt32(reader["ID"]));
        }
        reader.Close();
        bool Full_Roster = true;
        if (ID_temas_in_tourn.Count ==
my_Free_Tournament.free_Tournaments[i].count_teams)
        {
            //проверка соответствия наличия у всех команд
из турнира минимум 18 человек в заявке
            for (int j=0;j<ID_temas_in_tourn.Count;j++)
            {
                int value_in_DB = 0;
                command.CommandText = "SELECT Count
(ID_Team) as Val from Player where ID_Team = " +
ID_temas_in_tourn[j] + "";

```

```

        reader = command.ExecuteReader();
        while (reader.Read())
        {
            value_in_DB =
(Convert.ToInt32(reader["Val"]));

        }
        reader.Close();
        if (value_in_DB >= 18)
            continue;
        else
        {
            Full_Roster = false;
            break;
        }
    }
    if (Full_Roster)

ID_teams_in_freedom_generic_calendars.Add(my_Free_Tournament.free
e_Tournaments[i].ID);

    }

    my_Free_Tournament = new My_Free_Tournament();
    my_Free_Tournament.free_Tournaments = new
List<Free_Tournament>();

    for (int i =
0;i<ID_teams_in_freedom_generic_calendars.Count;i++)
    {
        Free_Tournament free_Tournament = new
Free_Tournament();

        command.CommandText = "Select * from [Tournament]
where ID = " + ID_teams_in_freedom_generic_calendars[i] + "";
        reader = command.ExecuteReader();
        while (reader.Read())
        {

```

```

        free_Tournament.ID =
Convert.ToInt32(reader["ID"]);

        free_Tournament.Title =
reader["Title"].ToString();

        free_Tournament.Sport_ID =
Convert.ToInt32(reader["Sport_ID"]);

        free_Tournament.format_tournament =
reader["format_tournament"].ToString();

        free_Tournament.count_teams =
Convert.ToInt32(reader["count_teams"]);

    }

my_Free_Tournament.free_Tournaments.Add(free_Tournament);

    reader.Close();

}

    string serialized =
JsonConvert.SerializeObject(my_Free_Tournament);

    sqlConnection.Close();

    return serialized;

}

// Функционал: Организатор. Добавление нового игрока
[HttpPost]

public void Insert_Player(int id_team, int Player_number,
string date_birth, int id_position, string name)
{

    string conn = "Data Source =DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

    SqlConnection sqlConnection;

    SqlCommand command;

    sqlConnection = new SqlConnection(conn);

    sqlConnection.Open();

    command = new SqlCommand();

    command.CommandText = "INSERT INTO [Player] values (
" + id_team + ", " + Player_number + ", '" + date_birth + "', " +
id_position + ", '" + name.Trim() + "')";

```

```

        command.Connection = sqlConnection;
        command.ExecuteNonQuery();
        sqlConnection.Close();
    }

    //Функционал: Организатор. Выдача в "Добавление игрока"
    всех доступных номеров команды

    [HttpPost]
    public string Select_Numbers(string id_team)
    {
        string conn = "Data Source =DESKTOP-
SMSD9LC\\SQLEXPRESS;Initial Catalog=SportsEvent;Integrated
Security=True;";

        SqlConnection sqlConnection;
        SqlCommand command;
        SqlDataReader reader;
        sqlConnection = new SqlConnection(conn);
        sqlConnection.Open();
        command = new SqlCommand();

        command.CommandText = "Select Player_number from
[Player] where ID_Team = " + id_team + ";";

        command.Connection = sqlConnection;
        reader = command.ExecuteReader();

        My_Number my_Number = new My_Number();
        my_Number.Numbers = new List<Number>();
        while (reader.Read())
        {
            Number number = new Number();

            number.Numb =
Convert.ToInt32(reader["Player_number"]);

            my_Number.Numbers.Add(number);
        }

        //Все номера от 1 до 99
        int[] array_all_numbers = new int[99];
        for (int i = 0; i < 99; i++)

```

```

    {
        array_all_numbers[i] = i + 1;
    }
    for (int i = 0; i < 99; i++)
    {
        for (int j = 0; j < my_Number.Numbers.Count; j++)
        {
            if
(array_all_numbers[i] == my_Number.Numbers[j].Numb)
            {
                array_all_numbers[i] = 1000;
                break;
            }
        }
    }
    my_Number.Numbers.Clear();
    for (int i = 0; i < 99; i++)
    {
        if (array_all_numbers[i] != 1000)
        {
            Number number = new Number
            {
                Numb = array_all_numbers[i]
            };
            my_Number.Numbers.Add(number);
        }
    }

    string serialized =
JsonConvert.SerializeObject(my_Number);

    sqlConnection.Close();

    return serialized;
}

```

}

Приложение В.
ИСХОДНЫЙ КОД МОБИЛЬНОГО ПРИЛОЖЕНИЯ
(рекомендуемое)

Листинг В.1 – Файл MainPage.xaml.cs

```
public partial class MainPage : ContentPage
{
    Entry entry_login;
    Entry entry_password;
    public MainPage()
    {
        InitializeComponent();
        BackgroundColor = Color.Black;
        Title = "Мой Спорт";
        var stacklayout = new StackLayout
        {
            Orientation = StackOrientation.Vertical,
            Padding = new Thickness(10, 10, 10, 10)
        };
        entry_login = new Entry
        {
            Placeholder = "Логин",
            TextColor = Color.White,
            BackgroundColor = Color.Black,
            PlaceholderColor = Color.Gold
        };
        entry_password = new Entry
        {
            Placeholder = "Пароль",
            TextColor = Color.White,
            BackgroundColor = Color.Black,
            PlaceholderColor = Color.Gold,
            IsPassword = true
        };
        Button button_authorized = new Button
        {
```

```

        Text = "Войти",
        BackgroundColor = Color.Black,
        TextColor = Color.White,
        HorizontalOptions = LayoutOptions.CenterAndExpand
    };

    button_authorized.Clicked +=
Button_authorized_Clicked;

    Button button_user = new Button
    {
        Text = "Войти как пользователь",
        BackgroundColor = Color.Black,
        TextColor = Color.White,
        HorizontalOptions = LayoutOptions.CenterAndExpand
    };

    stacklayout.Children.Add(entry_login);
    stacklayout.Children.Add(entry_password);
    stacklayout.Children.Add(button_authorized);
    stacklayout.Children.Add(button_user);
    button_user.Clicked += Button_user_Clicked;
    Content = stacklayout;
}

private void Button_user_Clicked(object sender, EventArgs
e)
{
    this.Navigation.PushAsync(new
Page_Authorization("0"));

    Navigation.RemovePage(this);
}

private void Button_authorized_Clicked(object sender,
EventArgs e)
{
    string resp = Change_Authorization(entry_login.Text,
entry_password.Text);

    if (resp=="1")

```

```

        this.Navigation.PushAsync(new
Page_Organizator());
    }

    private string Change_Autorization(string login_, string
password_)
    {
        WebRequest request =
WebRequest.Create("http://192.168.0.104/WebApplication1/Home/Ret
urn_Autorization");

        request.Method = "POST"; // для отправки используется
метод Post

        // данные для отправки
        string data = "login_" + login_ + "&password_" +
password_ + "";
        // преобразуем данные в массив байтов
        byte[] byteArray =
System.Text.Encoding.UTF8.GetBytes(data);
        // устанавливаем тип содержимого - параметр
ContentType
        request.ContentType = "application/x-www-form-
urlencoded";

        // Устанавливаем заголовок Content-Length запроса -
свойство ContentLength
        request.ContentLength = byteArray.Length;

        //записываем данные в поток запроса
        using (Stream dataStream =
request.GetRequestStream())
        {
            dataStream.Write(byteArray, 0, byteArray.Length);
        }

        WebResponse response = request.GetResponse();
        string json_string = "";
        using (Stream stream = response.GetResponseStream())
        {
            using (StreamReader reader = new
StreamReader(stream))

```

```

        {
            json_string += reader.ReadToEnd();
        }
    }
    if (json_string == "2")
    {
        this.Navigation.PushAsync(new
Page_Autorization(json_string));
        Navigation.RemovePage(this);
    }
    response.Close();
    return json_string;
}
}

```

Листинг В.2 – Файл Page_Autorization.xaml.cs

```

public partial class Page_Autorization : ContentPage
{
    //строка, содержащую роль клиента. 0 - пользователь, 1 -
    организатор, 2 - секретарь.
    string Id_role;
    ListView listView;
    MyTypeSports myTypeSports;
    public Page_Autorization(string ID_role)
    {
        Id_role = ID_role;
        InitializeComponent();
        Get_Type_Sports();
        BackgroundColor = Color.Black;
        var height = DeviceDisplay.MainDisplayInfo.Height;
        var width = DeviceDisplay.MainDisplayInfo.Width;

        Title = "Виды спорта";
    }
}

```

```

StackLayout stackLayout_top = new StackLayout
{
    Orientation = StackOrientation.Vertical,
    Padding = new Thickness(10, 10, 10, 10)
};

myTypeSports.Type_Sports[0].Image_Path =
"football.png";
myTypeSports.Type_Sports[1].Image_Path =
"hockey.png";
myTypeSports.Type_Sports[2].Image_Path = "rugby.png";
listView = new ListView
{
    SeparatorColor = Color.LightGoldenrodYellow,
    HasUnevenRows = true,
    BindingContext = myTypeSports.Type_Sports,
    ItemsSource = myTypeSports.Type_Sports,
    ItemTemplate = new DataTemplate(() =>
    {
        ImageCell imageCell = new ImageCell {
TextColor = Color.White };
        imageCell.SetBinding(ImageCell.TextProperty,
"Sport");

        imageCell.SetBinding(ImageCell.ImageSourceProperty,
"Image_Path");

        return imageCell;
    })
};

Button button_authorized = new Button
{
    Text = "Войти",
    BackgroundColor = Color.Black,
    TextColor = Color.White,
    HorizontalOptions = LayoutOptions.CenterAndExpand

```

```

};

listView.ItemSelected += ListView_ItemSelected;
stackLayout_top.Children.Add(listView);
Content = stackLayout_top;
}

private void ListView_ItemSelected(object sender,
SelectedItemChangedEventArgs e)
{
    Type_sports sport_selected = e.SelectedItem as
Type_sports;

    if (sport_selected != null)
    {
        listView.SelectedItem = null;

        Navigation.PushAsync(new
Page_FTournaments(sport_selected.ID, Id_role));
    }
}

private void Get_Type_Sports()
{
    WebRequest request =
WebRequest.Create("http://192.168.0.104/WebApplication1/Home/Ret
urn_Type_Sports");

    request.Method = "POST"; // для отправки используется
метод Post

                                // данные для отправки

    string data = "";

    // преобразуем данные в массив байтов

    byte[] byteArray =
System.Text.Encoding.UTF8.GetBytes(data);

    // устанавливаем тип содержимого - параметр
ContentType

    request.ContentType = "application/x-www-form-
urlencoded";

    // Устанавливаем заголовок Content-Length запроса -
свойство ContentLength

```

```

        request.ContentLength = byteArray.Length;

        //записываем данные в поток запроса
        using (Stream dataStream =
request.GetRequestStream())
        {
            dataStream.Write(byteArray, 0, byteArray.Length);
        }
        WebResponse response = request.GetResponse();
        string json_string = "";
        using (Stream stream = response.GetResponseStream())
        {
            using (StreamReader reader = new
StreamReader(stream))
            {
                json_string += reader.ReadToEnd();
            }
        }

        myTypeSports =
JsonConvert.DeserializeObject<MyTypeSports>(json_string);
        response.Close();
    }
}

```

Листинг В.3 – Файл Page_Organizator.xaml.cs

```

public partial class Page_Organizator : ContentPage
{
    public Page_Organizator()
    {
        InitializeComponent();
        BackgroundColor = Color.Black;
        var height = DeviceDisplay.MainDisplayInfo.Height;
        var width = DeviceDisplay.MainDisplayInfo.Width;

        Title = "Кабинет организатора";
    }
}

```

```

var stackLayout = new StackLayout
{
    Orientation = StackOrientation.Vertical,
    Padding = new Thickness(10, 150, 10, 10)
};

Button button_add_tournament = new Button
{
    Text = "Добавить турнир",
    BackgroundColor = Color.Black,
    TextColor = Color.White,
    HorizontalOptions =
LayoutOptions.CenterAndExpand,
    BorderWidth = 1,
    BorderColor = Color.Gold,
    WidthRequest = width/2
};

button_add_tournament.Clicked +=
Button_add_tournament_Clicked;

Button button_add_team = new Button
{
    Text = "Добавить команду",
    BackgroundColor = Color.Black,
    TextColor = Color.White,
    HorizontalOptions =
LayoutOptions.CenterAndExpand,
    BorderWidth = 1,
    BorderColor = Color.Gold,
    WidthRequest = width / 2
};

button_add_team.Clicked += Button_add_team_Clicked;

Button button_add_player = new Button

```

```

{
    Text = "Добавить игрока",
    BackgroundColor = Color.Black,
    TextColor = Color.White,
    HorizontalOptions =
LayoutOptions.CenterAndExpand,
    BorderWidth = 1,
    BorderColor = Color.Gold,
    WidthRequest = width / 2
};

button_add_player.Clicked +=
Button_add_player_Clicked;

Button button_generic_calendar = new Button
{
    Text = "Сгенерировать календарь",
    BackgroundColor = Color.Black,
    TextColor = Color.White,
    HorizontalOptions =
LayoutOptions.CenterAndExpand,
    BorderWidth = 1,
    BorderColor = Color.Gold,
    WidthRequest = width / 2
};

button_generic_calendar.Clicked +=
Button_generic_calendar_Clicked;

Button button_edit_calendar = new Button
{
    Text = "Редактировать календарь",
    BackgroundColor = Color.Black,
    TextColor = Color.White,
    HorizontalOptions =
LayoutOptions.CenterAndExpand,
    BorderWidth = 1,
    BorderColor = Color.Gold,

```

```

        WidthRequest = width / 2

    };

    button_edit_calendar.Clicked +=
Button_edit_calendar_Clicked;

    stackLayout.Children.Add(button_add_tournament);
    stackLayout.Children.Add(button_add_team);
    stackLayout.Children.Add(button_add_player);
    stackLayout.Children.Add(button_generic_calendar);
    stackLayout.Children.Add(button_edit_calendar);
    Content = stackLayout;
}

private void Button_edit_calendar_Clicked(object sender,
EventArgs e)
{
    this.Navigation.PushAsync(new Page_Edit_Calendar());
}

private void Button_generic_calendar_Clicked(object
sender, EventArgs e)
{
    this.Navigation.PushAsync(new
Page_Organizator_Generic_Calendar());
}

private void Button_add_player_Clicked(object sender,
EventArgs e)
{
    this.Navigation.PushAsync(new
Page_Organizator_Player());
}

private void Button_add_team_Clicked(object sender,
EventArgs e)
{
    this.Navigation.PushAsync(new
Page_Organizator_Team());
}

```

```
        private void Button_add_tournament_Clicked(object sender,
EventArgs e)
        {
            this.Navigation.PushAsync(new
Page_Organizator_Tournament());
        }
    }
```

Приложение Г.
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ СЕРВЕРНОЙ ЧАСТИ
(рекомендуемое)

POST http://192.168.0.104/WebApplication1/Home/Return_Tournaments?id=1

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Headers 7 hidden

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 1146 ms Size: 1.76 KB Save Response

Pretty Raw Preview Visualize HTML

```

1 [{"Tournaments":[{"ID":1,"Title":"Городская Футбольная лига","Description":"Круговой турнир. Победа - 3 очка, ничья - 1
2 очко, поражение - 0 очков","Sport_ID":1,"Time_half":7},{ID:2,"Title":"Кубок Новикова","Description":"Круговой турнир.
3 Победа - 3 очка, ничья - 1 очко, поражение - 0 очков","Sport_ID":1,"Time_half":7},{ID:3,"Title":"Первенство 2002
4 г.р.","Description":"Круговой турнир. Победа - 3 очка, ничья - 1 очко, поражение - 0
5 очков","Sport_ID":1,"Time_half":7},{ID:14,"Title":"Кожаный мяч","Description":"Традиционный новогодний
6 турнир","Sport_ID":1,"Time_half":8},{ID:18,"Title":"Турнир Трех Дворов","Description":"Районный турнир по
7 футболу","Sport_ID":1,"Time_half":7},{ID:19,"Title":"Турнир Сибирякова (U17)","Description":"Юниорский круговой
8 турнир","Sport_ID":1,"Time_half":5},{ID:22,"Title":"Школьный турнир","Description":"Турнир для школьников мкр.
9 Арбеково","Sport_ID":1,"Time_half":8},{ID:23,"Title":"Дворовый турнир","Description":"Турнир между дворовыми
10 командами","Sport_ID":1,"Time_half":7},{ID:24,"Title":"Сетка на 8","Description":"Футбольный турнир на
11 вылет","Sport_ID":1,"Time_half":6}]}]
```

Рисунок Г.1 – Получение турниров

POST http://192.168.0.104/WebApplication1/Home/Select_Free_Tournament_On_Generic

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Headers 7 hidden

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 2.42 s Size: 607 B Save Response

Pretty Raw Preview Visualize HTML

```

1 [{"free_Tournaments":[{"ID":17,"Title":"Кубок
2 Первухина","Sport_ID":3,"format_tournament":"Таблица","count_teams":6},{ID:18,"Title":"Турнир Трех
3 Дворов","Sport_ID":1,"format_tournament":"Таблица","count_teams":5},{ID:20,"Title":"Rugby Cup
4 2021","Sport_ID":3,"format_tournament":"Таблица","count_teams":5}]}]
```

Рисунок Г.2 – Получение турниров, для которых не сгенерирован календарь

POST

http://192.168.0.104/WebApplication1/Home/InsertTournament?title="Кубок Быстрова"&description="Ежегодный кубок среди детей 10-11 лет"&id_sport=1&time_half=10&format="Таблица"&count_teams=5

Send

Save

Params

Auth

Cookies

Code

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
☒	title	"Кубок Быстрова"			
☒	description	"Ежегодный кубок среди детей 10-11 лет"			
☒	id_sport	1			
☒	time_half	10			
☒	format	"Таблица"			
☒	count_teams	5			
	Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 228 ms

Size: 205 B

Save Response

Рисунок Г.3 – Создание турнира

POST

http://192.168.0.104/WebApplication1/Home/Return_Team_Roster_In_Playing?id_team1=1&id_team2=2&id_tournament=1

Send

Save

Params

Authorization

Headers (13)

Body

Pre-request Script

Tests

Settings

Cookies

Code

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
☒	id_team1	1			
☒	id_team2	2			
☒	id_tournament	1			
	Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 52 ms

Size: 6.12 KB

Save Response

Pretty

Raw

Preview

Visualize

HTML

```
1 [{"Rosters":[{"ID_player":1,"Name":"Crng
2 Плетикоса", "Number":1, "ID_Position":1, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":2, "Name":"Антон
3 Касапуев", "Number":44, "ID_Position":1, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":908, "Name":"Евгений
4 Козлов", "Number":2, "ID_Position":1, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":3, "Name":"Ероп
5 Краско", "Number":33, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":4, "Name":"Анатолий
6 Бикос", "Number":13, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":5, "Name":"Сепрей
7 Сидос", "Number":17, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":6, "Name":"Сепрей
8 Белый", "Number":12, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":7, "Name":"Евгений
9 Куропаточкин", "Number":29, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":8, "Name":"Максим
10 Белов", "Number":4, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":9, "Name":"Иоцарт", "Number":5, "ID_Position":3, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"),
11 {"ID_player":10, "Name":"Александр
12 Шейков", "Number":125, "ID_Position":3, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":11, "Name":"Басилий
13 Касапуев", "Number":10, "ID_Position":3, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":12, "Name":"Константин
14 Кичев", "Number":16, "ID_Position":3, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":13, "Name":"Нагоред
15 Акимомов", "Number":90, "ID_Position":3, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":14, "Name":"Антон
16 Сидор", "Number":99, "ID_Position":3, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":15, "Name":"Лес
17 Борисов", "Number":27, "ID_Position":3, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":16, "Name":"Роман
18 Павленко", "Number":19, "ID_Position":4, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":17, "Name":"Райн", "Number":7, "ID_Position":4, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"),
19 {"ID_player":18, "Name":"Александр
20 Прудников", "Number":85, "ID_Position":4, "Position":null, "ID_tournament":0, "ID_team":1, "Team":"Снаптрак"), {"ID_player":19, "Name":"Глеб
21 Носов", "Number":120, "ID_Position":1, "Position":null, "ID_tournament":0, "ID_team":2, "Team":"Гранит"), {"ID_player":20, "Name":"Климент
22 Архипов", "Number":44, "ID_Position":1, "Position":null, "ID_tournament":0, "ID_team":2, "Team":"Гранит"), {"ID_player":21, "Name":"Геннадий
23 Васильков", "Number":59, "ID_Position":1, "Position":null, "ID_tournament":0, "ID_team":2, "Team":"Гранит"), {"ID_player":22, "Name":"Андрей
24 Кручинин", "Number":33, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":2, "Team":"Гранит"), {"ID_player":23, "Name":"Николай
25 Рассказов", "Number":5, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":2, "Team":"Гранит"), {"ID_player":24, "Name":"Евгений
26 Класов", "Number":17, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":2, "Team":"Гранит"), {"ID_player":25, "Name":"Степан
27 Зерочкин", "Number":12, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":2, "Team":"Гранит"), {"ID_player":26, "Name":"Антон
28 Куропаткин", "Number":29, "ID_Position":2, "Position":null, "ID_tournament":0, "ID_team":2, "Team":"Гранит"), {"ID_player":27, "Name":"Маркос
```

Рисунок Г.4 – Получение составов команд

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** http://192.168.0.104/WebApplication1/Home/Return_Team?id=20
- Send Button:** A blue button labeled "Send" with a dropdown arrow.
- Params Tab:** Contains a table for Query Params.
- Query Params Table:**

KEY	VALUE	DESCRIPTION
☒ Id	20	
Key	Value	Description
- Body Tab:** Shows the response body in JSON format.
- Status Bar:** Displays "Status: 200 OK", "Time: 24 ms", and "Size: 298 B".
- Response Body:**

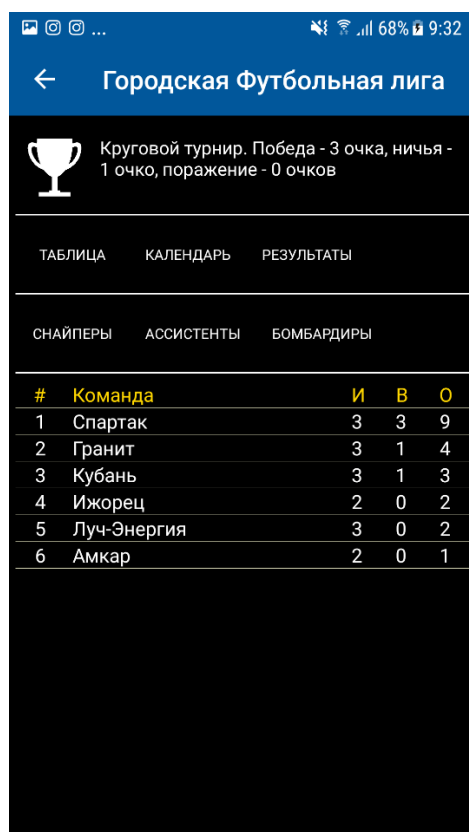
```
1 { "one_Teams": [{"Title": "ЗнелНав-2008", "ID": 0}] }
```

Рисунок Г.5 – Получение названия команды

Приложение Д.

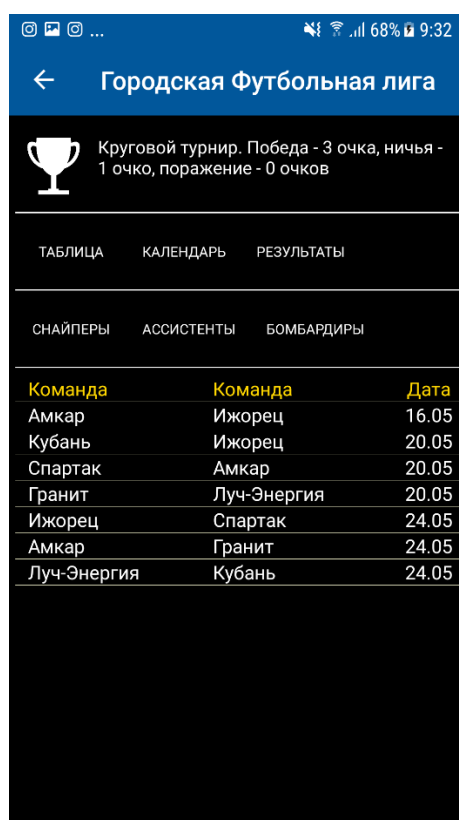
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

(рекомендуемое)



#	Команда	И	В	О
1	Спартак	3	3	9
2	Гранит	3	1	4
3	Кубань	3	1	3
4	Ижорец	2	0	2
5	Луч-Энергия	3	0	2
6	Амкар	2	0	1

Рисунок Д.1 – Просмотр таблицы турнира



Команда	Команда	Дата
Амкар	Ижорец	16.05
Кубань	Ижорец	20.05
Спартак	Амкар	20.05
Гранит	Луч-Энергия	20.05
Ижорец	Спартак	24.05
Амкар	Гранит	24.05
Луч-Энергия	Кубань	24.05

Рисунок Д.2 – Просмотр календаря турнира



Команда	Команда	Счет
Спартак	Кубань	2-1
Гранит	Ижорец	0-0
Луч-Энергия	Амкар	1-1
Спартак	Гранит	2-0
Ижорец	Луч-Энергия	2-2
Кубань	Амкар	2-1
Луч-Энергия	Спартак	1-2
Гранит	Кубань	3-1

Рисунок Д.3 – Просмотр результатов матчей турнира



#	Игрок	И	Г
1	Моцарт	3	3
2	Антон Хапов	3	2
3	Егор Краско	2	1
4	Ильнур Маметов	2	1
5	Александр Дёмочкин	3	1
6	Давид Пашинян	1	1
7	Евгений Петров	3	1
8	Валерий Плахута	1	1
9	Максим Солнцев	2	1
10	Алексей Иванов	1	1
11	Лев Борисов	1	1
12	Жо	2	1
13	Анатолий Быков	1	1
14	Владимир Попов	3	1
15	Андрей Степанов	1	1
16	Роман Павлюченко	3	1

Рисунок Д.4 – Просмотр снайперов турнира

...

68%

9:33

Городская Футбольная лига

Круговой турнир. Победа - 3 очка, ничья - 1 очко, поражение - 0 очков

ТАБЛИЦА

КАЛЕНДАРЬ

РЕЗУЛЬТАТЫ

СНАЙПЕРЫ

АССИСТЕНТЫ

БОМБАРДИРЫ

№	ИГРОК	Г	А
1	Егор Вахромеев	2	2
2	Роман Павлюченко	3	2
3	Василий Казарцев	2	1
4	Максим Солнцев	2	1
5	Евгений Куропаточкин	1	1
6	Давид Пашинян	1	1
7	Виктор Луговской	2	1
8	Ильнур Маметов	2	1
9	Александр Борисов	2	1
10	Игорь Чугайнов	2	1
11	Федот Стрельцов	1	1
12	Евгений Викулов	2	1
13	Рауль	2	1
14	Сергей Лопушанский	2	1
15	Эктор Бракамонте	2	1
16	Маркос Рохо	2	1
17	Карвальо	2	1

Рисунок Д.5 – Просмотр ассистентов турнира

...

68%

9:33

←

Городская Футбольная лига

Круговой турнир. Победа - 3 очка, ничья - 1 очко, поражение - 0 очков

ТАБЛИЦА

КАЛЕНДАРЬ

РЕЗУЛЬТАТЫ

СНАЙПЕРЫ

АССИСТЕНТЫ

БОМБАРДИРЫ

#	Игрок	И	Г	А	О
1	Моцарт	3	3	0	3
2	Роман Павлюченко	3	1	2	3
3	Александр Борисов	2	1	1	2
4	Антон Хапов	3	2	0	2
5	Максим Солнцев	2	1	1	2
6	Егор Вахромеев	2	0	2	2
7	Ильнур Маметов	2	1	1	2
8	Давид Пашинян	1	1	1	2
9	Виктор Луговской	2	0	1	1
10	Евгений Петров	3	1	0	1
11	Валерий Плахута	1	1	0	1
12	Игорь Чугайнов	2	0	1	1
13	Федот Стрельцов	1	0	1	1
14	Евгений Викулов	2	0	1	1
15	Алексей Иванов	1	1	0	1
16	Лев Борисов	1	1	0	1

Рисунок Д.6 – Просмотр бомбардиров турнира



Скриншот мобильного приложения, отображающий состав команды «Гранит». Вверху — статус-бар с иконками уведомлений, 68% заряда батареи и временем 9:33. Под ним — заголовок «Гранит» с кнопкой «←» и иконкой футболки. Ниже — три вкладки: «РЕЗУЛЬТАТЫ», «КАЛЕНДАРЬ» и «СОСТАВ», где последняя активна. Основное содержимое — таблица с 20 игроками.

#	Игрок	Позиция
20	Глеб Носов	Вр.
44	Климент Архипов	Вр.
59	Геннадий Васильков	Вр.
33	Андрей Кручинин	Зщ.
3	Николай Рассказов	Зщ.
17	Евгений Квасов	Зщ.
22	Степан Земочкин	Зщ.
29	Антон Куропаткин	Зщ.
4	Маркос Рохо	Зщ.
5	Кирилл Чудинов	Пзщ.
25	Сергей Широков	Пзщ.
10	Виталий Глебов	Пзщ.
16	Валерий Карпин	Пзщ.
90	Александр Сёмин	Пзщ.
98	Глеб Павлов	Пзщ.
27	Семён Рукавишников	Пзщ.
88	Андрей Степанов	Пзщ.
58	Анатолий Быков	Пзщ.
87	Жо	Нап.

Рисунок Д.7 – Просмотр состава команды



Скриншот мобильного приложения, отображающий календарь команды «Гранит». Интерфейс идентичен предыдущему скриншоту, но активна вкладка «КАЛЕНДАРЬ». В таблице показаны последние два матча.

Команда	Команда	Дата
Гранит	Луч-Энергия	20.05
Амкар	Гранит	24.05

Рисунок Д.8 – Просмотр календаря команды

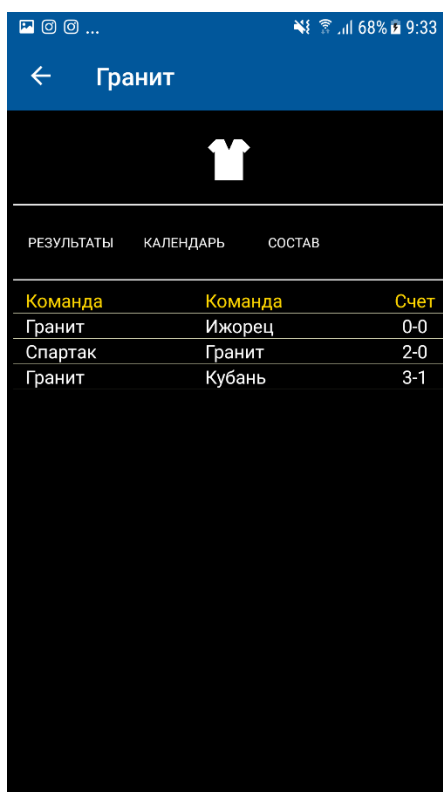


Рисунок Д.9 – Просмотр результатов матчей команды



Рисунок Д.10 – Просмотр протокола матча



Рисунок Д.11 – Просмотр составов команд на матч

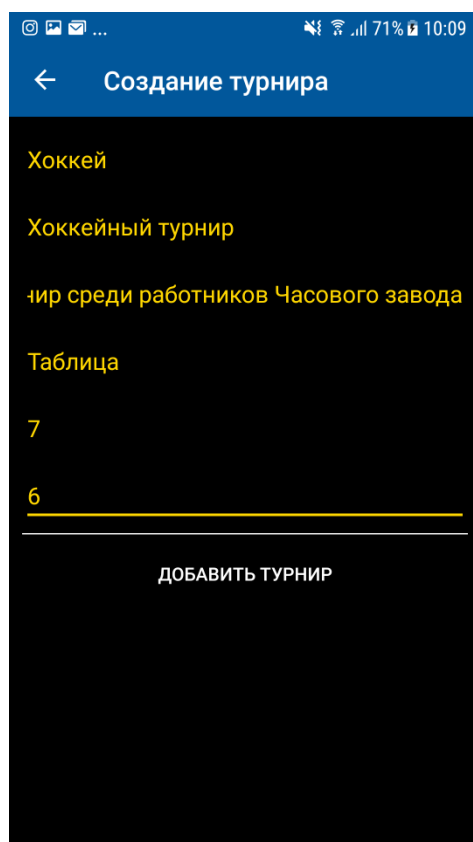


Рисунок Д.12 – Добавление турнира

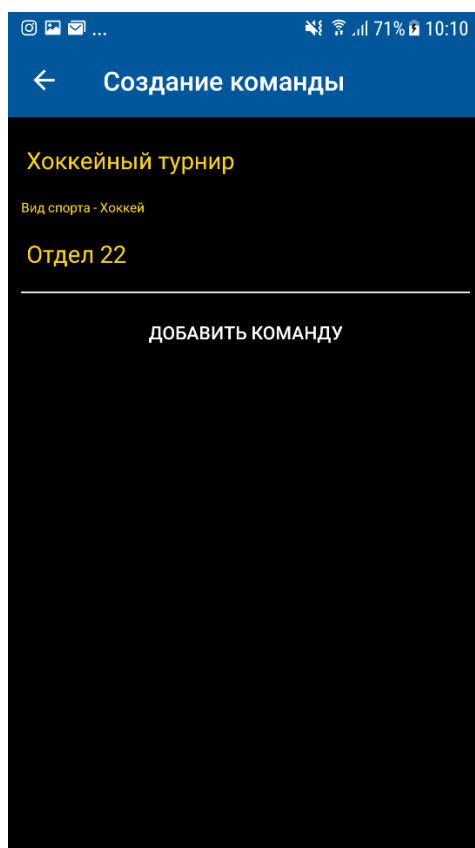


Рисунок Д.13 – Добавление команды в турнир (первая часть)

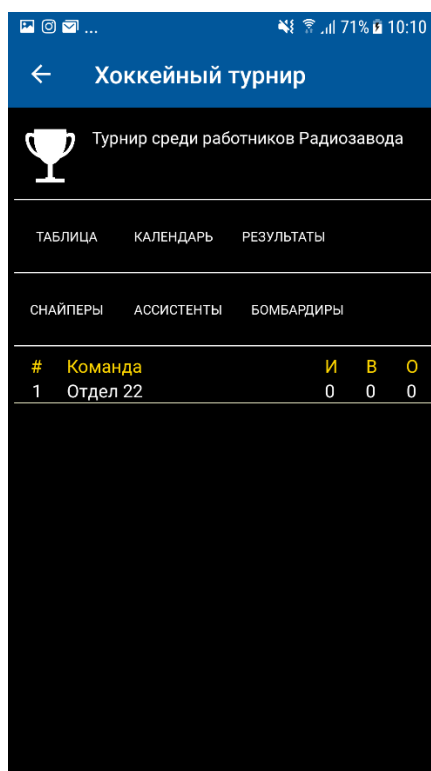


Рисунок Д.14 – Добавление команды в турнир (вторая часть)

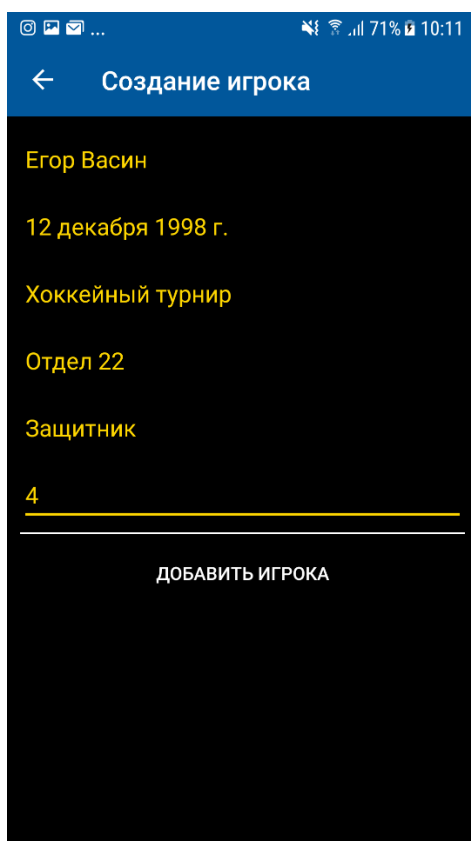


Рисунок Д.15 – Добавление игрока в команду (первая часть)



Рисунок Д.16 – Добавление игрока в команду (вторая часть)



Рисунок Д.17 – Редактирование календаря турнира (первая часть)



Рисунок Д.18 – Редактирование календаря турнира (вторая часть)



Рисунок Д.19 – Редактирование календаря турнира (третья часть)

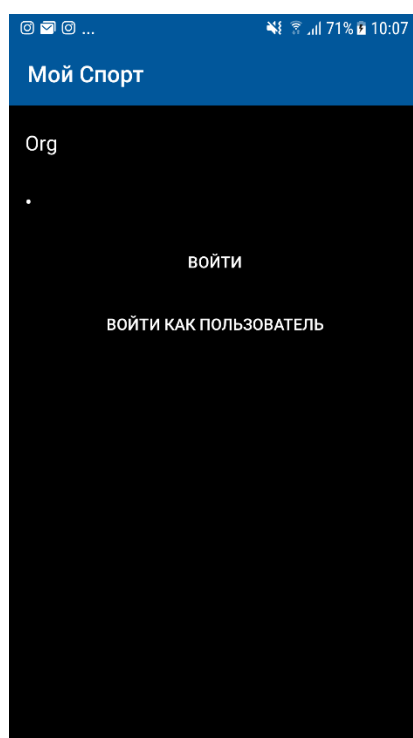


Рисунок Д.20 – Авторизация в приложении (первая часть)

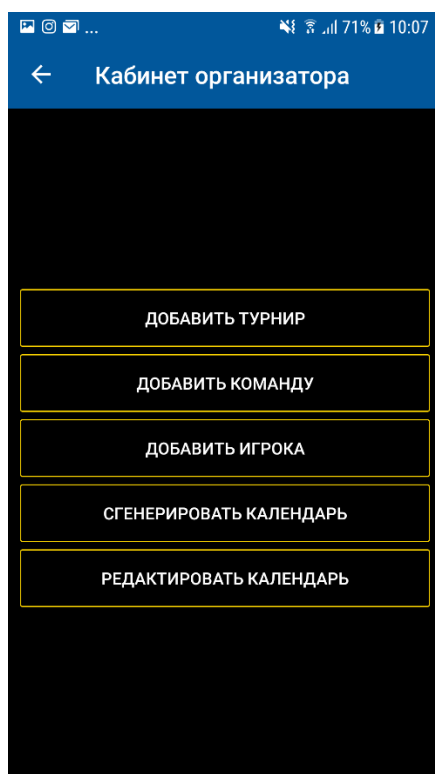


Рисунок Д.21 – Авторизация в приложении (вторая часть)



Рисунок Д.22 – Заполнение составов команд на матч

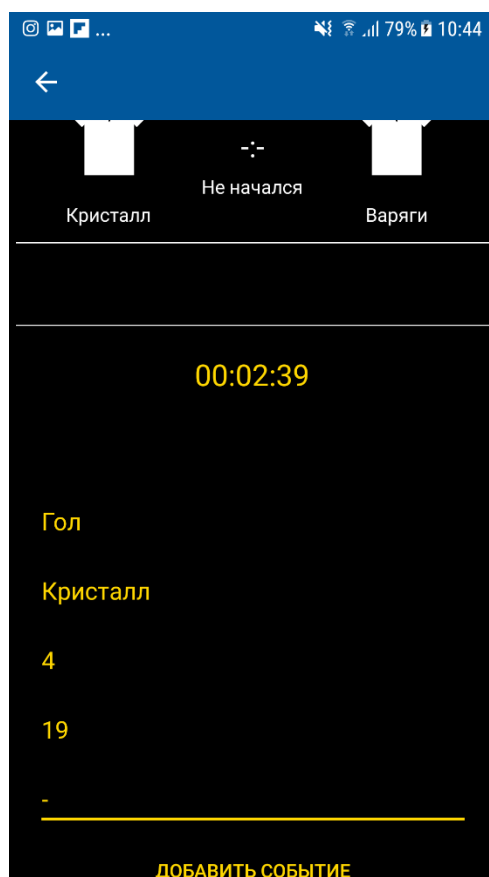


Рисунок Д.23 – Заполнение протокола матча (первая часть)

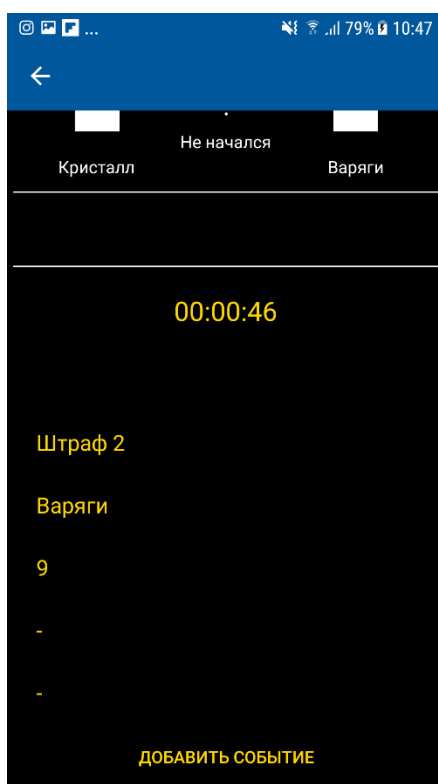


Рисунок Д.24 – Заполнение протокола матча (вторая часть)



Рисунок Д.25 – Заполнение протокола матча (третья часть)



Рисунок Д.26 – Заполнение протокола матча (четвертая часть)