Частное учреждение образования

«Колледж бизнеса и права»

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

«Спортивные Новости»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проект по предмету «Базы данных и системы

управления базами данных»

КП Т.1910016.401

|  |  |
| --- | --- |
|  |  |
| Преподаватель | Сельханович М. А. |
| Учащийся | Короленко Д. А. |

2025

СОДЕРЖАНИЕ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

КП Т.1910016.401

Разраб.

Короленко Д.А

Провер.

Сельханович М. А.

Реценз.

Н. Контр.

Утверд.

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ «Спортивные Новости»

Лит.

Листов

КБП

50

у

[1 Описание задачи 5](#_Toc193271630)

[1.1 Анализ предметной области 5](#_Toc193271631)

[1.2 Постановка задачи 6](#_Toc193271632)

[2 Проектирование мобильного приложения 8](#_Toc193271633)

[2.1 Проектирование модели 8](#_Toc193271634)

[2.2 Требования к мобильному приложению 12](#_Toc193271635)

[2.3 Структура мобильного приложения 13](#_Toc193271636)

[2.4 Проектирование макета мобильного приложения 16](#_Toc193271637)

[2.4.1 Экран загрузки (Splash Screen) 16](#_Toc193271638)

[2.4.2 Экран входа (Log in screen) 17](#_Toc193271639)

[2.4.3 Экран регистрации (Sign up screen) 18](#_Toc193271640)

[2.4.4 Главный экран (Home Screen) 18](#_Toc193271641)

[2.4.5 Экран профиля (Profile Screen) 19](#_Toc193271642)

[2.4.6 Экран матчей (Matches Screen) 20](#_Toc193271643)

[2.4.7 Экран статистики (Statistics Screen) 20](#_Toc193271644)

[2.4.8 Экран новостей (News Screen) 21](#_Toc193271645)

[2.4.9 Экран настроек (Settings Screen) 21](#_Toc193271646)

[2.5 Защита и сохранность данных 22](#_Toc193271647)

[2.6 Организация и ведение информационной базы (модели) 22](#_Toc193271648)

[2.6.1 Таблица News 23](#_Toc193271649)

[2.6.2 Таблица Users 23](#_Toc193271650)

[2.6.3 Таблица news\_tags 23](#_Toc193271651)

[2.6.4 Связи между таблицами 24](#_Toc193271652)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

КП Т.1910016.401

Разраб.

Короленко Д.А

Провер.

Сельханович М. А.

Реценз.

Н. Контр.

Утверд.

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ «Спортивные Новости»

Лит.

Листов

КБП

50

у

[3 Реализация мобильного приложения 26](#_Toc193271653)

[3.1 Программно-технические средства, необходимые для разработки приложения 26](#_Toc193271654)

[3.2 Описание разделов приложения 26](#_Toc193271655)

[3.3 Функциональное тестирование 27](#_Toc193271656)

[4 Примененим 32](#_Toc193271657)

[4.1 Общие сведения 32](#_Toc193271658)

[4.2 Назначение мобильного приложения 33](#_Toc193271659)

[Заключение 34](#_Toc193271660)

[Список информационных источников 35](#_Toc193271661)

[ПРИЛОЖЕНИЕ А 36](#_Toc193271662)

[ПРИЛОЖЕНИЕ Б 52](#_Toc193271665)

**Введение**

В современном мире спорт играет важную роль не только как часть здорового образа жизни, но и как значимое социальное и культурное явление. С ростом популярности спортивных событий увеличивается и спрос на оперативное получение актуальной информации о результатах матчей, трансляциях и аналитических материалах. Однако существующие платформы зачастую перегружены информацией, что затрудняет поиск нужных данных для пользователей. В связи с этим разработка мобильного приложения, которое предоставляет удобный и персонализированный доступ к спортивным новостям, является актуальной задачей.

Целью курсового проекта является разработка мобильного приложения Спортивные новости для агрегации и персонализированного представления спортивных новостей, обеспечивающего пользователям удобный доступ к актуальной информации.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ существующих приложений и платформ для получения спортивных новостей;

- исследовать потребности целевой аудитории и сформулировать требования к функционалу приложения;

- разработать архитектуру и интерфейс мобильного приложения;

- реализовать основные функции приложения, включая персонализацию контента и уведомления;

- провести тестирование приложения и оценить его удобство использования.

Объектом исследования являются мобильные приложения для получения спортивных новостей. Предмет исследования — методы и технологии разработки персонализированных мобильных приложений.

В работе использовались следующие методы:

- анализ существующих решений;

- проектирование интерфейсов;

- разработка схеы базы данных;

- разработка программного обеспечения с использованием различных технологий;

- тестирование и оценка удобства использования.

Практическая значимость работы заключается в создании мобильного приложения, которое может быть использовано спортивными фанатами для удобного и быстрого доступа к актуальным новостям.

1. Описание задачи
   1. Анализ предметной области

Темой индивидуального задания является разработка мобильного приложения «Спортивные Новости». Приложение будет посвящено спортивной тематике и предоставит пользователям доступ к актуальным новостям, видео и расписаниям матчей. Оно также будет включать функционал для просмотра статистики, результатов игр и персонализированных уведомлений о событиях любимых команд, а также искусственный интеллект, который поможет разобраться с любым вопросом в спортивной сфере. Интуитивно понятный интерфейс обеспечит удобство использования, а современные технологии позволят приложению работать быстро и без сбоев.

Предметная область данной разработки связана с новостного приложения связана с управлением, анализом и предоставлением информации о спортивных событиях, командах, игроках и матчах. Приложение помогает пользователям следить за ходом соревнований, анализировать результаты и получать оперативные обновления.

Спорт является важной составляющей жизни современного общества. Большое количество людей занимается спортом или просто наблюдает за проведением спортивных мероприятий. Это создает потребность в быстром информировании пользователей о спортивных новостях. Также многие предпочитают мобильность в просмотре видео спортивного характера. В связи с этим проект фокусируется на предоставлении удобного инструмента для получения новостей и видео о спорте на мобильных устройствах.

Основным объектом в системе является матч. У каждого матча есть уникальный идентификатор, временные рамки, команды-участники и результаты. Команды обладают своими названиями, логотипами, составами игроков и статистикой. Игроки представляют собой членов команды, чья активность и действия непосредственно влияют на результаты матча.

Целевая аудитория приложения включает:

- спортивных болельщиков, активно следящих за играми, командами и соревнованиями;

- любителей аналитики и прогнозов, которые предпочитают изучать статистику и анализировать спортивные события;

- активных пользователей мобильных устройств, которым важно получать новости и видео в удобном формате, где бы они ни находились;

- профессиональных спортсменов и тренеров, которые могут использовать приложение для анализа матчей и просмотра актуальных событий.

Во время матча фиксируются события, такие как голы, замены или карточки. Эти события записываются с указанием времени, команды, игрока и типа события. После завершения матча данные используются для анализа и формирования статистики, например, рейтингов игроков, результатов команд или прогнозов на будущие соревнования.

Функционал приложения включает обработку событий матча и предоставление аналитики. Пользователи могут изучать информацию о командах, смотреть лучшие моменты матчей. Дополнительно приложение будет ориентировано на предоставление актуальных новостей и спортивных видеоматериалов, что удовлетворит потребность пользователей в мобильном доступе к информации.

В качестве целевых устройств выбрано направление мобильных устройств, так как на текущий момент это наиболее используемые и удобные устройства для просмотра информации. Таким образом, разработка приложения под Android позволит охватить широкую аудиторию пользователей и удовлетворить их запросы в спортивной тематике.

Постановка задачи

Данный курсовой проект направлен на разработку системы баз данных для серверной части мобильного приложения «Спортивные Новости».

В рамках проекта будет разработана серверная часть с системой баз данных, обеспечивающая хранение и обработку информации о пользователях, новостях, матчах, лигах и других данных. Также будет создан пользовательский интерфейс мобильного приложения, удобный для просмотра контента, поиска информации и взаимодействия с другими болельщиками.

Пользовательский функционал будет включать:

- регистрацию и авторизацию (вход через email/пароль);

- восстановление пароля (через email);

- редактирование профиля (изменение аватара, имени);

- настройки уведомлений (push-уведомления о матчах, новостях, трансляциях);

- просмотр новостей с возможностью фильтрации по командам, лигам, датам;

- поиск и фильтрация новостей, по ключевым словам,

- категоризацию новостей по спорту (футбол, баскетбол, хоккей и т. д.);

- просмотр комментариев, лайков и сохранение статей в «Избранное»;

- просмотр видео и лучших моментов (интеграцияc с Instend и YouTube Api, встроенный видеоплеер);

- таблицы чемпионатов (например, Серия A, Ла Лига, АПЛ);

- расписание матчей с указанием времени начала, стадиона;

- архив прошедших матчей с результатами и статистикой;

- онлайн-обновления матчей (счет, составы, важные моменты);

- историю встреч между командами;

- прогнозы на матчи (AI-аналитика);

- чат болельщиков;

- возможность делиться новостями и видео в соцсетях.

Для обеспечения работы приложения предусмотрен административный интерфейс, который позволит:

- добавлять и редактировать новости, но основной поток новостей будет собираться с сайта Беларусь 5;

- модерировать комментарии и пользователей (блокировка, предупреждения).

Техническая часть задач включает в себя реализацию таких функций как:

- мультиязычная поддержка;

- api для интеграции с внешними сервисами (gemini, YouTube и Instend);

- структура базы данных.

Система баз данных будет включать такие основные таблицы как:

- пользователи (логин, email, аватар, подписки);

- новости (заголовок, текст, изображение, категория, теги, дата публикации);

- видео (ссылка на YouTube, описание, категория);

- матчи (дата, команды, счет, стадион, составы);

- таблицы лиг (положение команд, статистика) будут приходить в виде json со стороннего API;

- комментарии и лайки (пользователь, новость, текст, дата).

Разрабатываемая система баз данных обеспечит работу серверной части приложения «Спортивные Новости», позволяя пользователям получать актуальную информацию о спортивных событиях в удобном формате. Приложение станет полезным инструментом для болельщиков, объединяя новости, видео, статистику и социальное взаимодействие в одном месте.

1. Проектирование мобильного приложения
   1. Проектирование модели

Для проектирования модели используются UML-диаграммы, которые наглядно показывают систему в виде диаграмм и удобном для прочтения виде. Диаграмма прецедентов или диаграмма вариантов использования ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Use case diagram) в UML – диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент – функциональность системы, доступная пользователям, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних [требований](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%BC%D1%83_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8E) к системе.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

На рисунке 2.1 представлена диаграмма прецедентов, которая отображает способы взаимодействия пользователя с программой. Мобильное приложение «Спортивные новости» подразумевает различные способы взаимодействия, получение и отображение разного вида информации. Основными способами использования приложения являются:

- регистрация;

- авторизация;

- просмотр последних спортивных новостей;

- получение уведомлений о результатах матча и перед началом матча;

- просмотр видео;

- просмотр турнирной таблицы;

- разговор с AI;

- просмотр информации про матч;

- изменение настроек.

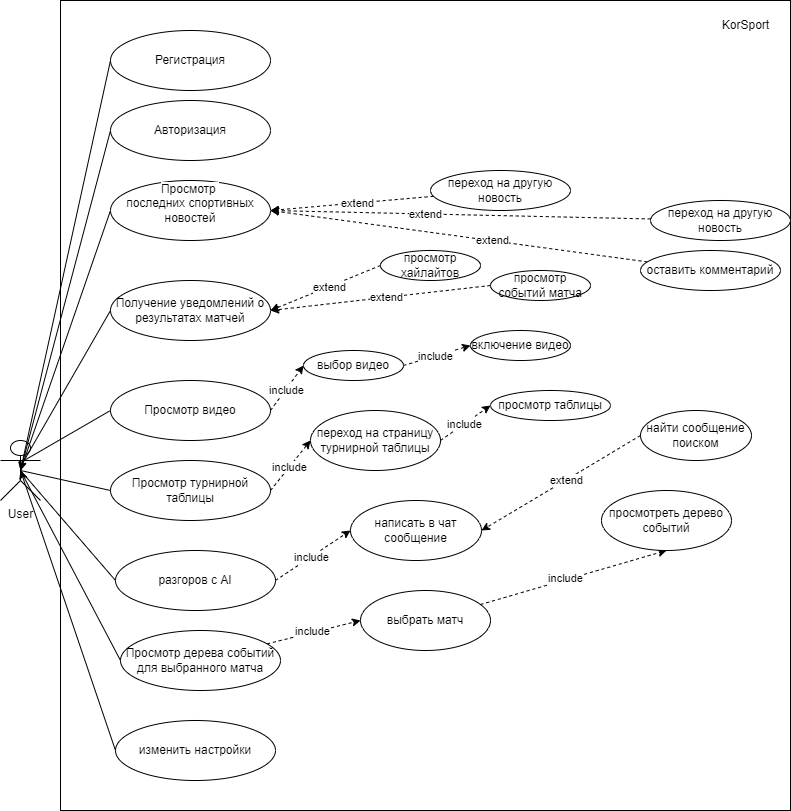


Рисунок 2.1 – Диаграмма вариантов использования

На диаграмме последовательности изображен процесс взаимодействия между четырьмя основными участниками передачи данных, а именно загрузчик данных, Сервер, Клиентская часть приложения и Пользователь.

загрузчик данных начинает свою работу, инициируя процесс получения данных с целевого веб-сайта. После завершения загрузки, загрузчик сохраняет результат в формате JSON-файла, который содержит структурированные данные с веб-ресурса.

Сервер запускает процесс обработки JSON-файла, полученного от загрузчика. В ходе обработки JSON-файла сервер извлекает данные и сохраняет их в базу данных.

Клиентская часть приложения инициирует запрос к серверу с целью получения последних новостей. Сервер обрабатывает запрос клиента, извлекая данные из базы данных. После обработки сервер отправляет последние новости в клиентскую часть приложения также в формате json.

Клиентская часть приложения получает данные от сервера и начинает их обработку для отображения. После обработки данных, клиентская часть формирует представление и отображает последние спортивные новости для пользователя. Пользователь видит обновлённые данные, которые были запрошены и обработаны системой.

На диаграмме также показаны действия пользователя: Пользователь взаимодействует с приложением, инициируя просмотр новостей. Приложение, в свою очередь, отправляет запрос к серверу для получения данных. Сервер передаёт данные приложению, которое их визуализирует для пользователя.

Процесс работы можно разделить на несколько этапов:

- этап сбора данных. Включает загрузка и сохранение в формате JSON;

- этап обработки данных на сервере. Включает извлечение, сохранение и предоставление данных по запросу;

этап обработки на клиентской стороне. Включает получение, форматирование и отображение информации.

Диаграмма визуализирует синхронное взаимодействие между участниками системы через сообщения, отмеченные стрелками. Каждое сообщение соответствует определённой функции: запрос, передача данных или отображение. Диаграмма подчёркивает, что данные проходят три ключевых уровня обработки: загрузчик, сервер, клиент.

Взаимодействие завершается успешным отображением данных пользователю, обеспечивая удобный доступ к актуальным новостям.

Диаграмма демонстрирует строгую последовательность действий, начиная с автоматического загрузки данных и заканчивая отображением информации для конечного пользователя.

Загрузчик данных играет ключевую роль в извлечении свежей информации с веб-сайтов, формируя исходные данные для дальнейшей обработки.

JSON-файл служит стандартным форматом передачи данных между загрузчиком и сервером, упрощая их обработку и интеграцию.

Сервер выполняет функцию центрального узла системы, обрабатывая данные, храня их в базе и передавая клиенту по запросу.

Взаимодействие между сервером и клиентской частью приложения организовано через запросы на получение последних новостей, что обеспечивает актуальность отображаемой информации.

Клиентская часть приложения отвечает за обработку данных, полученных от сервера, и их преобразование в удобный для пользователя интерфейс.

Пользователь взаимодействует только с клиентской частью приложения, не замечая сложных процессов обработки, происходящих на сервере и этапе загрузки.

Важным моментом является хранение данных в базе на сервере, что обеспечивает возможность повторного доступа без необходимости повторного загрузки.

Общая структура работы системы позволяет эффективно обновлять данные и предоставлять пользователю максимально актуальную информацию. Диаграмма последовательности представлена на рисунке 2.2.

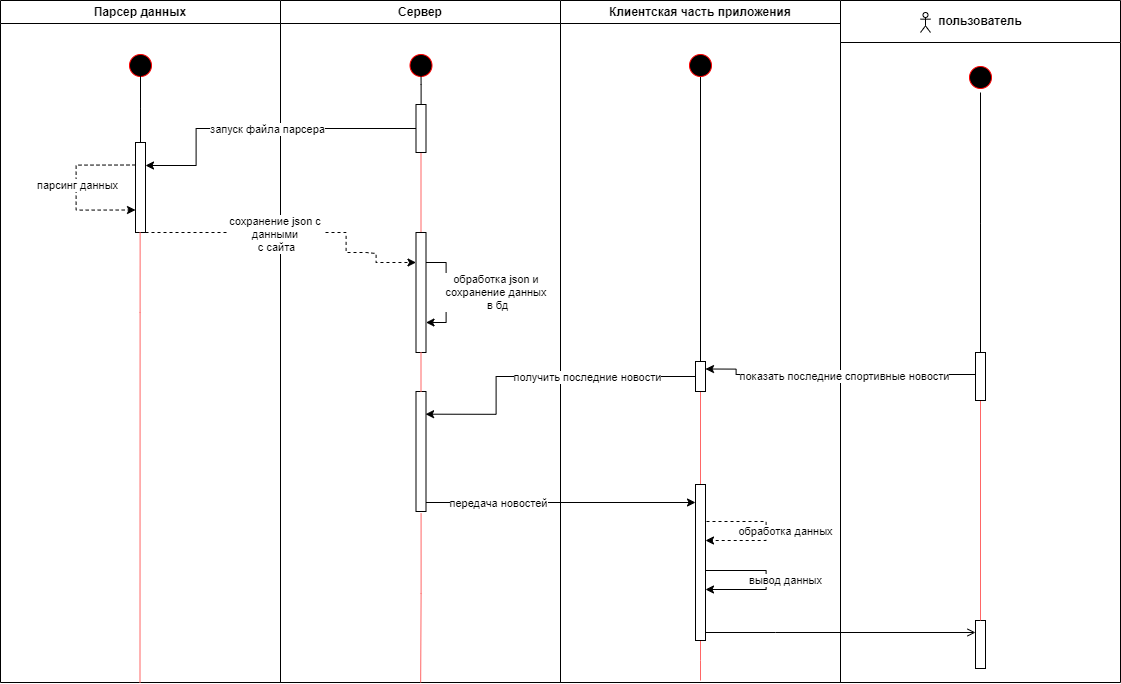


Рисунок 2.2 – Диаграмма последовательности

На рисунке 2.3 изображена user flow диаграмма, которая описывает процесс взаимодействия пользователя с приложением. Начинается с запуска приложения, после чего пользователь выбирает, зарегистрирован он или нет. Если пользователь зарегистрирован, он вводит email, пароль и нажимает кнопку "Войти", после чего переходит на страницу AI. Если пользователь не зарегистрирован, он вводит email, пароль и нажимает кнопку "Регистрация", также попадая на страницу AI. Далее пользователь вводит вопрос и получает ответ от AI.

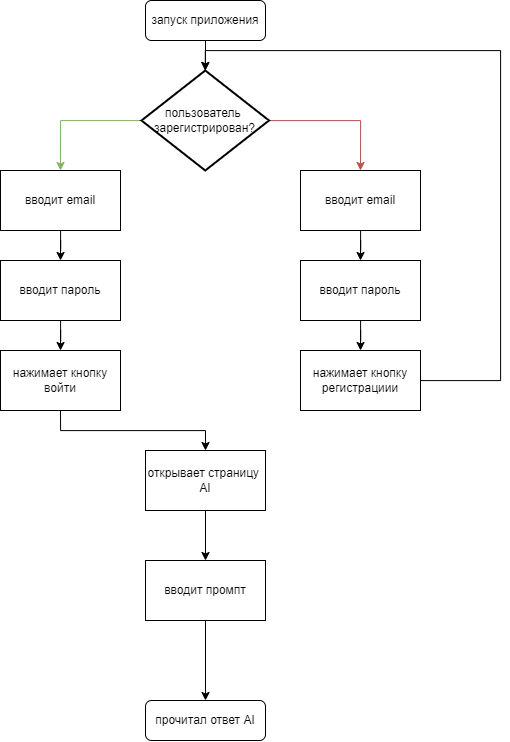


Рисунок 2.3 – Диаграмма User Flow

Требования к мобильному приложению

Приложение «Спортивные Новости» предназначено для мобильных устройств и должно работать на операционных системах, обеспечивающих стабильность, высокую производительность и совместимость с современными технологиями. Во-первых, приложение должно поддерживать Android 8.0 (Oreo) и выше, чтобы охватить широкую аудиторию пользователей и обеспечить совместимость с актуальными устройствами. Кроме того, обязательным является наличие Google Play Services, так как возможна интеграция с сервисами Google, такими как уведомления Firebase.

Для хранения данных чата с ботом требуется поддержка локальной базы данных, такой как SQLite. Важно, чтобы операционная система поддерживала адаптивные иконки, что позволит приложению корректно отображать свой значок на различных устройствах. Также необходимо наличие поддержки динамического масштабирования интерфейса, что обеспечит удобное использование приложения на экранах с разными разрешениями. Поддержка API уведомлений (Notification API) играет важную роль для оперативного информирования пользователей о событиях и новостях.

Поскольку приложение использует сервер для получения данных, операционная система должна обеспечивать надежное интернет-соединение как через Wi-Fi, так и через мобильные сети. Для плавного отображения анимации и интерфейса важна поддержка OpenGL ES 3.0 или выше.

В приложении будет использоваться такой шрифт как Roboto различного начертания и размера.

Контент является важной составляющей любого приложения, и важно соблюдать некоторые принципы и требования, чтобы у пользователя сложилось приятное впечатление.

Требования к новостям:

- новости должны быть актуальными и обновляться в реальном времени;

- заголовки должны быть информативными, краткими и привлекать внимание пользователя;

- в новостях должны присутствовать изображения и видео для наглядности;

- должна быть возможность поиска и фильтрации новостей по дате, категории и популярности;

- источники новостей должны быть проверенными и достоверными.

Требования к ответам AI:

- ответы должны быть релевантными, информативными и корректными;

- исключение нецензурной лексики, дезинформации и некорректных данных;

- ответы должны быть четко структурированы и легко восприниматься.

Требования к комментариям пользователей:

- шрифт должен быть читабельным;

- система модерирования и фильтрации оскорбительных или спам-сообщений;

- возможность поставить лайк или дизлайк;

- ограничение по длине комментариев для предотвращения спама.

Соблюдение всех вышеуказанных требований позволит «Спортивные Новости» работать стабильно, эффективно и комфортно для пользователей.

* 1. Структура мобильного приложения

Организация данных очень важный процесс при создании системы такого типа. Ведь именно благодаря правильной структуре данные будут отображаться быстро и правильно. Важным условием является своевременность и актуальность полученных данных, для этого в разрабатываемой системе будет использоваться API, для получения необходимой и актуальной спортивной информации.

В приложении организована структура data классов, которые представляют классы, используемые для десериализации данных, которые приходят с внешнего API, в формате json.

В структуру входит папки, по которым и разделяются модельки для удобного использования. В структуре содержатся 2 основные папки: api, и domain.

Папка api содержит файлы, которые используются для получения данных. В структуру данной папки входят:

– папка entities;

– папка mappers;

– папка viewModels;

– файл SoccerRepository.

В папке entities хранятся data классы, которые представляют модели, приходящие из внешнего источника, далее модели разделяются по папкам на основании принадлежности их к разным сущностям. Структура папки entities, выглядит так:

– matches;

– matchReport;

– ranking.

В папке mappers хранятся файлы типа. kt, в которых описаны методы для взаимодействия с данными, которые приходят.

В папке viewModels «вью модели” – это сущности для сохранения состояний, которые «живут» от их создания до закрытия приложения.

Файл SoccerRepository является интерфейсом, который обеспечивает правильное и лаконичное передвижение данных между метода и функциями, данным принимает информацию, записанную в формате json и с при помощи различных инструментов преобразует данные в нужный формат.

JSON (JavaScript Object Notation) — это текстовый формат обмена данными, основанный на синтаксисе объектов JavaScript. Он используется для хранения и передачи структурированных данных между приложениями. JSON прост в чтении и написании как для человека, так и для машины.

Правильная организация данных в структуре json файла очень важна для быстрого доступа к необходимым полям объекта, что позволит работать приложению быстро и корректно. Ниже приведен пример разбора файла, который хранит в себе информацию про определенный матч:

* competition\_id: Идентификатор турнира;
* season\_id: Идентификатор сезона;
* match\_id: Идентификатор матча;
* match\_unique\_code: Уникальный код;
* date: Дата матча;
* match\_day\_id: Идентификатор тура;
* match\_day\_name: Название тура;
* match\_day\_short\_name: Сокращение тура;
* date\_order: Дата проведения;
* team\_a\_id: ID команды A;
* team\_a\_name: Название команды;
* team\_a\_short\_name: Сокращённое название;
* team\_a\_acronym: Акроним команды;
* team\_b\_id: ID команды B;
* team\_b\_name: Название команды;
* team\_b\_short\_name: Сокращённое название;
* team\_b\_acronym: Акроним команды;
* goals\_team\_a: Голы команды A;
* goals\_team\_b: Голы команды B;
* stadium\_name: Название стадиона;
* stadium\_city: Город стадиона;
* match\_status: Статус матча;
* is\_abandoned: Матч отменён;
* is\_postponed: Матч перенесён;
* is\_suspended: Матч приостановлен;
* broadcaster: Транслятор матча;
* match\_start\_time: Время начала;
* match\_phase: Фаза матча;
* opta\_id: Opta идентификатор;
* is\_forfeit\_win: Техническая победа;
* minute: Текущая минута.

Таким образом эта структура, которая создается после сериализации объекта отображает все необходимые поля.

На рисунке 2.4 представлен пример содержимого json файла с расписанием матчей.

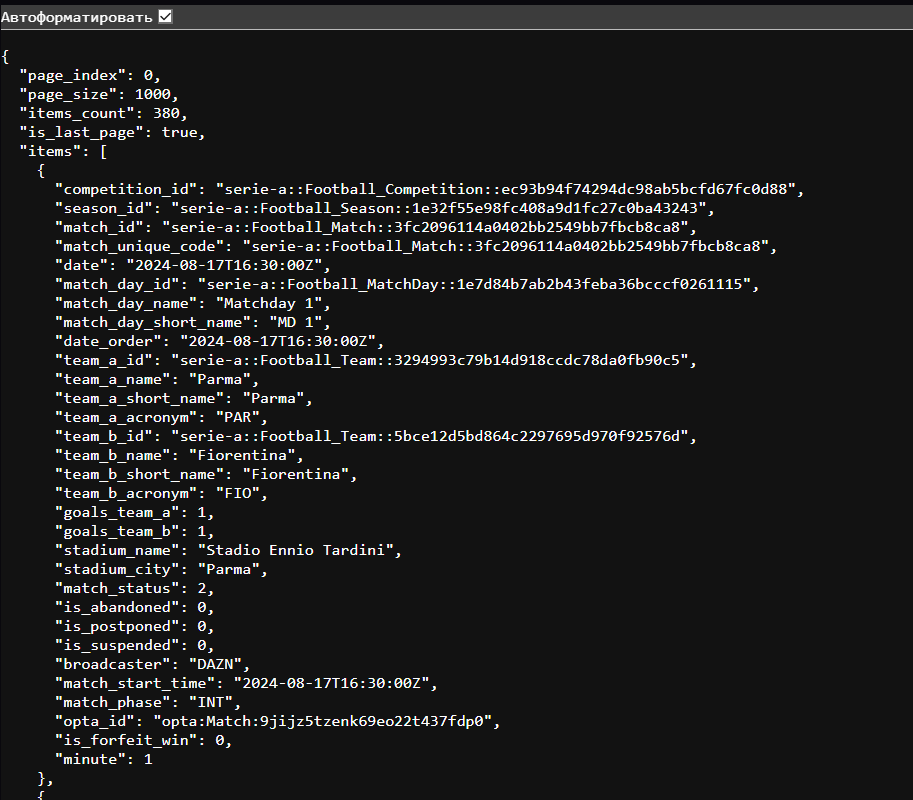


Рисунок 2.4 – Пример файла json

Проектирование макета мобильного приложения

* + 1. Экран загрузки (Splash Screen)

Экран загрузки – это первый экран, который видит пользователь при открытии приложения. Он содержит логотип приложения «Спортивные Новости» и логотипы партнеров KorFood и KorFin. Фоновый цвет экрана – белый, что делает его минималистичным и стильным. В центре экрана расположена иконка с изображением футболиста в движении – логотип приложения. Это помогает сразу создать ассоциацию с футбольной тематикой приложения. Также под логотипом находится мотивационная фраза: «Будь в курсе, будь лидером». Данный экран не требует никакого взаимодействия от пользователя. Он появляется лишь на несколько секунд, пока загружается основное приложение. Цветовая гамма экрана выдержана в спортивном стиле, что подчеркивает его тематику. Логотипы партнеров находятся в нижней части экрана, но при этом они достаточно заметны. Отсутствие сложных анимаций делает загрузку быстрой и плавной. Это важно, так как пользователи не любят долгие ожидания. Используемые шрифты четкие и легко читаемые. Нет лишних деталей, которые могли бы отвлекать от основной информации. Экран выполнен в современном стиле, с акцентом на простоту и удобство. Визуально он передает динамику спорта и футбола. После завершения загрузки приложение автоматически переходит на следующий экран. Пользователю не нужно нажимать никаких кнопок. Весь экран направлен на создание первого положительного впечатления о приложении. Лаконичный дизайн помогает не перегружать внимание лишними элементами. Размер шрифта подобран таким образом, чтобы текст был читаемым даже на небольших экранах. Визуальная композиция сбалансирована, элементы размещены пропорционально. Основная цель экрана – мгновенно вовлечь пользователя в спортивную атмосферу. Белый фон делает приложение стильным и современным. В целом, экран загрузки является важной частью пользовательского опыта. Он создает первое впечатление и задает тон всему приложению.

Экран входа (Log in screen)

Экран входа предназначен для авторизации пользователей в приложении. Он имеет лаконичный дизайн с белым фоном и минимальным количеством элементов. В верхней части находится иконка футболиста, которая поддерживает общую тематику приложения. Основное содержимое экрана – это две текстовые формы для ввода email и пароля. Поля ввода расположены по центру и имеют удобный размер. Под ними находится красная кнопка «Войти», которая привлекает внимание. Ниже есть ссылка «Еще нет аккаунта? Зарегистрируйтесь», которая ведет на экран регистрации. Используется минимальное количество цветов – белый, черный и красный, что делает интерфейс чистым и удобным. Кнопка входа выделяется за счет контрастного красного цвета. Ввод данных осуществляется в привычном формате, а пароль скрывается для безопасности. Приложение также может поддерживать автозаполнение полей, если оно включено на устройстве. В случае неверного ввода система может показать сообщение об ошибке. Надписи на экране выполнены простыми и понятными словами. Элементы расположены таким образом, чтобы пользователь мог быстро авторизоваться. Дизайн адаптирован под различные размеры экранов смартфонов. Вся информация представлена четко и структурированно. Нет лишних графических элементов, которые могли бы отвлекать пользователя. Экран помогает создать интуитивный пользовательский опыт. Кнопка входа располагается так, чтобы ее было удобно нажимать даже одной рукой. Если пользователь забыл пароль, возможно, здесь будет кнопка «Забыли пароль?». Все элементы интерфейса находятся на своих логичных местах. Экран прост и интуитивно понятен любому пользователю. Его главная цель – быстрая и удобная авторизация.

Экран регистрации (Sign up screen)

Экран регистрации имеет схожий дизайн с экраном входа. Он предназначен для создания нового аккаунта в приложении. В верхней части расположена иконка с футболистом, что подчеркивает спортивную тематику. Пользователю предлагается ввести email и пароль. Поля ввода имеют удобный размер и заметные границы. Под ними расположена красная кнопка «Создать», которая завершает процесс регистрации. Ниже есть ссылка «Уже есть аккаунт? Авторизируйтесь», которая возвращает пользователя на экран входа. Визуальный стиль экрана чистый и минималистичный. Используются белый фон, черный текст и красные акценты. Размер шрифтов оптимален для удобного чтения. Ввод данных осуществляется стандартным способом, а пароль скрывается для безопасности. Если данные введены некорректно, приложение может сообщить об ошибке. Все элементы интерфейса расположены логично и удобно. Регистрация занимает минимум времени. Главное, чтобы пользователи могли быстро пройти этот этап и начать пользоваться приложением. Интерфейс адаптирован под разные размеры экранов. Ничего лишнего, только основные элементы, необходимые для создания аккаунта. Форма ввода содержит минимальное количество полей, что ускоряет процесс регистрации. Цветовая схема подчеркивает важность элементов и делает их интуитивно понятными. Экран понятен даже тем, кто впервые использует приложение. Его структура продумана до мелочей для удобства пользователей.

Главный экран (Home Screen)

Главный экран – это основной раздел приложения, где пользователь получает доступ к актуальной информации. В верхней части экрана расположен блок с названием лиги и счетом последнего матча. Это сразу привлекает внимание пользователя. Ниже находится раздел спортивных новостей. В нем представлены карточки с заголовками и изображениями, которые делают контент более наглядным. Еще ниже есть раздел с лучшими моментами (Highlight), где можно увидеть ключевые события матчей. Интерфейс выполнен в чистом и современном стиле. Используется сочетание белого фона, черного текста и красных акцентов. Навигация осуществляется через нижнюю панель с иконками. В ней есть кнопки для перехода на другие разделы приложения. Контент подается в удобном формате с возможностью прокрутки. Размер карточек новостей адаптирован для комфортного чтения. Картинки новостей делают интерфейс визуально привлекательным. Весь экран спроектирован так, чтобы пользователь мог быстро находить нужную информацию. Включение матчей и новостей в один экран делает его максимально полезным. Система обновляется в реальном времени, предоставляя свежие данные. Пользователь может кликнуть на карточку новости, чтобы перейти к ее полной версии. В разделе новостей отображается дата публикации и тематика статьи. Карточки выполнены в едином стиле, что делает интерфейс гармоничным. Визуальная иерархия помогает пользователю сразу понимать, что является главным. Навигация интуитивна, а интерфейс удобен даже для новых пользователей.

Экран профиля (Profile Screen)

Экран профиля позволяет пользователю просматривать и редактировать свою личную информацию. В верхней части экрана отображается аватар пользователя, который можно изменить, загрузив новое изображение. Рядом указано имя пользователя и его email. Ниже располагается список основных настроек аккаунта. Пользователь может изменить пароль, включить или отключить уведомления и выбрать предпочитаемый язык интерфейса. Также присутствует раздел статистики, где отображаются персональные достижения и активность в приложении. Например, сколько матчей пользователь просмотрел или какие команды добавил в избранное. Интерфейс экрана выполнен в минималистичном стиле с удобной навигацией. Белый фон делает страницу чистой, а красные акценты выделяют ключевые элементы. Кнопка «Редактировать профиль» позволяет изменять личные данные. Внизу экрана находится кнопка «Выйти из аккаунта», выделенная красным цветом. Все элементы расположены так, чтобы ими можно было удобно пользоваться одной рукой. Настройки организованы в виде списка с иконками, что делает их более понятными. Раздел достижений мотивирует пользователя быть активнее в приложении. Возможна интеграция с социальными сетями для синхронизации данных. Экран автоматически обновляется после внесения изменений. При необходимости система запрашивает подтверждение перед выходом из аккаунта. Пользователь может загрузить обложку профиля для персонализации страницы. Адаптивный дизайн позволяет комфортно работать с экраном на разных устройствах. Навигация остается интуитивной даже для новых пользователей. В целом, экран профиля помогает пользователю управлять своим аккаунтом и улучшать взаимодействие с приложением.

Экран матчей (Matches Screen)

Экран матчей отображает расписание предстоящих и прошедших футбольных игр. В верхней части экрана есть вкладки «Прошедшие» и «Предстоящие», позволяющие быстро переключаться между разделами. В каждой карточке матча указываются команды, дата и время игры, а также текущий статус. Для завершенных матчей дополнительно показывается счет. Дизайн карточек выполнен в строгом, но стильном формате с логотипами команд. Клик по карточке открывает подробную информацию о матче. Здесь пользователь может увидеть голы, карточки, замену игроков и другие события. Для предстоящих матчей возможен выбор напоминания. Если пользователь хочет следить за конкретной командой, он может добавить ее в избранное. Экран обновляется в реальном времени, чтобы отображать свежие результаты. Используется удобный фильтр для поиска матчей по лигам или командам. Навигация проста, все элементы располагаются логично. Цветовая гамма остается фирменной: белый фон, черный текст и красные акценты. Карточки матчей адаптированы для разных размеров экранов. Интерактивные элементы (например, кнопки «Добавить в избранное») интуитивно понятны. Внизу экрана можно увидеть сводку лучших игр текущей недели. Пользователь может включить push-уведомления для интересующих его матчей. Благодаря этому он не пропустит важные события. В целом, экран матчей делает просмотр футбольного контента удобным и быстрым.

Экран статистики (Statistics Screen)

Этот экран предоставляет детализированную информацию о командах, игроках и матчах. В верхней части находятся фильтры, позволяющие выбирать турниры, клубы или игроков. Главный блок содержит общую статистику по чемпионату: количество голов, побед, поражений и ничьих. Далее идет раздел с индивидуальной статистикой лучших игроков: голеадоров, ассистентов и вратарей. Каждая карточка игрока содержит его фотографию, имя и текущий клуб. Клик по карточке открывает детальный профиль спортсмена. Здесь можно увидеть его возраст, рост, национальность и статистику за сезон. Данные обновляются в режиме реального времени. д. Визуальные графики помогают анализировать показатели команд и игроков. Интерактивные элементы позволяют быстро переключаться между различными разделами. Шрифт и цвета подобраны так, чтобы все данные были легко читаемыми. В разделе командной статистики можно сравнить несколько клубов между собой. Для каждого клуба есть информация о текущем месте в турнирной таблице. Используемые диаграммы и таблицы делают восприятие информации проще. Адаптивный дизайн обеспечивает удобное отображение на разных устройствах. Экран ориентирован как на обычных болельщиков, так и на более продвинутых аналитиков. В целом, этот раздел приложения полезен для всех, кто хочет глубже разбираться в футболе.

Экран новостей (News Screen)

Этот экран предоставляет пользователям свежие футбольные новости. В верхней части расположены категории: «Популярное», «Матчи», «Игроки», «Клубы». Основной контент представлен в виде карточек с изображениями и заголовками. Клик по карточке открывает полную статью. Тексты новостей адаптированы для мобильного просмотра. На экране предусмотрена лента с возможностью бесконечной прокрутки. Цветовая схема поддерживает фирменный стиль приложения. Каждая статья содержит дату публикации и источник информации. Возможна интеграция с push-уведомлениями, чтобы пользователи не пропускали важные события. Для удобства реализован поиск по ключевым словам. Карточки новостей отображают краткое описание статьи. Пользователь может сохранить интересные статьи в избранное. В разделе «Популярное» отображаются самые читаемые материалы за последние дни. Приложение может поддерживать комментарии под статьями. При долгом нажатии на новость возможен быстрый предпросмотр. Экран новостей обновляется в режиме реального времени. Возможна интеграция с видеоматериалами, например, интервью игроков. В целом, этот экран делает чтение новостей удобным и интересным.

Экран настроек (Settings Screen)

Экран настроек позволяет пользователю персонализировать приложение под себя. Здесь можно изменить язык интерфейса, включить или отключить уведомления и выбрать тему оформления. Визуальный стиль остается минималистичным. В разделе «Аккаунт» можно сменить пароль или выйти из системы. Приложение поддерживает темную и светлую тему. В разделе «Уведомления» можно настроить частоту получения новостей и матчевых оповещений. Доступен выбор любимых команд для персонализированного контента. В разделе «О приложении» содержится информация о разработчиках и версиях. Все элементы интерфейса сгруппированы для удобства навигации. Фирменная цветовая схема остается неизменной. Экран адаптирован для работы на разных устройствах. Доступна функция сброса настроек к стандартным параметрам. Возможен экспорт данных профиля. Приложение поддерживает биометрическую авторизацию. В целом, экран настроек помогает сделать приложение максимально удобным для пользователя.

Защита и сохранность данных

В приложении «Спортивные Новости» особое внимание уделяется защите пользовательских данных и предотвращению несанкционированного доступа. Для этого реализуются несколько уровней безопасности, включая ограничение доступа, управление разрешениями и защиту от утечек информации.

Ограничение доступа к данным

Доступ к данным внутри приложения разграничивается в зависимости от типа информации и уровня прав пользователя. Так, например, личные данные доступны только авторизованным пользователям, а административные функции (добавление новостей, редактирование контента) доступны только модераторам или администраторам в отдельном приложении для администраторов.

Информация о матчах, новостях и видео открыта для всех пользователей, но без возможности модификации.

Запросы к серверу проходят через защищенные API, исключая возможность получения конфиденциальных данных сторонними приложениями.

Приложение требует минимальный набор разрешений, необходимых для его корректной работы:

- доступ к интернету – требуется для загрузки новостей, матчей и видео;

- доступ к уведомлениям – используется для отправки push-уведомлений о спортивных событиях.

При первом запуске приложение запрашивает только те разрешения, которые необходимы в данный момент. Пользователь может в любое время изменить настройки разрешений через системные параметры Android.

Защита информации от несанкционированного использования

Для предотвращения утечки данных и несанкционированного доступа применяются следующие меры:

- шифрование данных – передача данных между клиентом и сервером осуществляется через защищенный протокол HTTPS;

- хранение данных в защищенной среде – SQLite-база данных приложения хранится во внутреннем каталоге, недоступном для других приложений;

- личные данные пользователей хранятся в зашифрованном виде на сервере.

Организация и ведение информационной базы (модели)

Организация и ведение информационной базы приложения «Спортивные Новости» имеет сложную структуру, которая состоит из 3 систем управления базами данных (SQL server для работы сервера, FireBase хранения пользовательских данных и получения удобных сервисов от данной СУБД, а также SQl lite для организации хранения данных локально), несколько сторонних API, которые предоставляют данные

База данных «Спортивные новости» на сервере предназначена для хранения и управления спортивными новостями и пользовательской информацией. Она состоит из нескольких взаимосвязанных таблиц, каждая из которых выполняет свою роль в обеспечении работы системы.

Таблица News

Таблица News отвечает за хранение спортивных новостей. В ней содержатся ключевые поля:

- Sport — вид спорта, к которому относится новость (например, футбол, баскетбол);

- DateTime — дата и время публикации новости (используется как первичный ключ);

- Title — заголовок новости;

- ImageId — идентификатор изображения, связанного с новостью;

- ArticleText — основной текст статьи.

Эта таблица является основной для хранения контента, который будет отображаться пользователям – новость.

Таблица Users

Таблица Users предназначена для хранения информации о пользователях системы. Она включает в себя:

UserEmail — электронная почта пользователя (используется как уникальный идентификатор);

UserImage — изображение профиля пользователя.

Хотя в таблице немного полей, в дальнейшем она может быть расширена дополнительными данными, такими как имя, фамилия или предпочтения пользователя.

Таблица news\_tags

Эта таблица предназначена для хранения тегов, привязанных к новостям. В ней есть следующие поля:

* TagId — уникальный идентификатор тега (первичный ключ);
* NewsDateTime — дата и время новости, к которой относится тег (внешний ключ, ссылающийся на News (DateTime));
* Tag — сам тег (например, "Чемпионат мира", "Трансферы").

Связь между таблицами News и news\_tags организована с помощью внешнего ключа, который реализует каскадное удаление. Это означает, что при удалении новости из News, все связанные с ней теги также будут автоматически удалены. На рисунке 2.5 предоставлена ER диаграмма, которая показывает отношение между сущностями в базе данных.

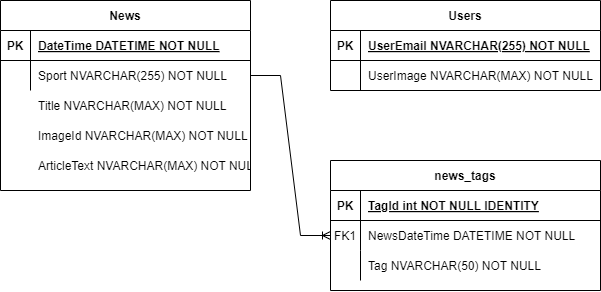


Рисунок 2.5 – ER диаграмма

Связи между таблицами

News и news\_tags связаны через поле NewsDateTime, обеспечивая возможность добавления нескольких тегов к одной новости.

Users пока не имеет прямой связи с остальными таблицами, но в дальнейшем можно добавить таблицу комментариев или систему подписок.

В News первичным ключом является DateTime.

В Users первичный ключ отсутствует, но уникальным идентификатором является UserEmail.

В news\_tags первичный ключ — TagId, а NewsDateTime является внешним ключом.

Используемые типы данных:

- NVARCHAR для текстовых данных (названий, тегов, email);

- DATETIME для хранения даты и времени;

- INT IDENTITY для автоинкрементных идентификаторов.

С применением технологии связывания таблиц между собой с помощью первичных и вторичных ключей в базе данных созданы между таблицами такие связи как:

- news\_tags связана с News через NewsDateTime (внешний ключ);

- Users пока не связана с другими таблицами, но может быть расширена;

- все связанные теги в news\_tags удаляются благодаря ON DELETE CASCADE

1. Реализация мобильного приложения
   1. Программно-технические средства, необходимые для разработки приложения

Для разработки приложения «Спортивные Новости», были выбраны такие технологии как: C#, Kotlin, SQL Server, FireBase, selenium и др.

Каждая технология используется для разных компонентов. Так, например, при помощи ASP.Net, C# реализуется серверная часть приложения, которая выполняет важную роль при использовании приложения. C# — это современный объектно-ориентированный язык программирования, разработанный Microsoft. Он используется для создания серверной части приложения благодаря своей производительности и широким возможностям. C# интегрируется с платформой ASP.NET, что позволяет разрабатывать надежные и безопасные веб-приложения. Также язык поддерживает асинхронное программирование, что улучшает отзывчивость серверной части.

JetPack Compose отлично подходит для создания декларативного написанного интерфейса и клиентской части приложения для андроида, а так же в будущем такую реализацию приложения легко перевести на другие платформы благодаря multiplatform, Kotlin — это современный язык программирования, который официально поддерживается Google для разработки Android-приложений. Kotlin полностью совместим с Java, что позволяет использовать существующие библиотеки и frameworks. В сочетании с Jetpack Compose он становится идеальным выбором для создания современных пользовательских интерфейсов.

Firebase — это платформа для разработки мобильных и веб-приложений, предоставляющая множество готовых решений. Она включает Firestore для хранения данных в реальном времени, что идеально подходит для чатов или динамических обновлений. Firebase также предлагает сервисы для пуш-уведомлений, авторизации пользователей и аналитики. Это значительно ускоряет разработку и упрощает поддержку приложения. FireBase предоставляет различные удобные сервисы такие как:

- нереляционные базы данных;

- пуш-уведомления и подписки на тему пуш-уведомлений;

- авторизация по почте.

И другие сервисы также могут быть использованы при дальнейшем расширении приложения.

Описание разделов приложения

При запуске приложения пользователю отображается приветственный экран с логотипом и названием, который демонстрируется несколько секунд. Если пользователь запускает приложение впервые, его перенаправляют на страницу авторизации и аутентификации. На этой странице пользователь может зарегистрироваться, создав новый аккаунт, или войти с использованием уже существующих данных. После успешной авторизации пользователь попадает на главный экран приложения.

Основное нижнее навигационное меню включает в себя такие разделы как, «Главная», «Видео», «Матчи», «Искусственный интеллект», «Избранное». После выбора раздела открывается соответствующий контент,

Пользовательский интерфейс приложения создан с использованием Jetpack Compose, что позволяет разрабатывать адаптивные и современные экраны. Главная страница содержит ленту новостей и кнопки для перехода к другим разделам. Анимации и стили оформления помогают создать приятный пользовательский опыт. Для работы с сетью и обработки данных используется библиотека Ktor, а данные хранятся в списках и JSON. Push-уведомления настраиваются через Firebase, чтобы уведомлять пользователя о важных событиях. Также реализована пагинация для загрузки большого количества данных.

В разделе «Главная страница» расположены последние новости и лучшие моменты матчей, а также ближайший/текущий матч. Новости и лучшие моменты расположены в горизонтальном строке, которая поддерживает пролистывание, а благодаря поддержке сервером пагинации пользователь не ожидает долго для получения информации. Данные поступают порционно.

В разделе «Расписание матчей», расположен список матчей, а также кнопка для перехода на страницу ранговой таблицы и список игровых дней («match day»)

Функциональное тестирование

Тестирование работы мобильного приложения проводилась с использование устройства со следующими характеристиками:

- операционная система Android 13;

- использование сетей LTE и Wi-Fi;

- оперативная память объёмом 6 ГБ;

- процессор Helio G95;

- экран 6.4 дюймов.

А также с использованием эмулятора, для проверки корректности

работы и совместимости с устройствами, которые имеют другие свойства. Результаты тестирования и тест-кейсы приведены в таблице 4.1

Поскольку приложение про спортивные новости разрабатывается под платформу Android, его тестирование проводится на различных устройствах с этой операционной системой. Это позволяет убедиться в корректной работе всех функций приложения на разных версиях Android и устройствах с различными техническими характеристиками. В ходе тестирования особое внимание уделяется стабильности работы приложения, отображению информации в реальном времени, а также корректности отображения новостей и статистики. Тестирование включает проверку навигации между экранами, функциональности отображения матчей, работы с профилями команд и игроков. Кроме того, проводится проверка отображения иконок и правильности работы кнопок нижней навигационной панели. Поскольку приложение ориентировано на оперативное предоставление спортивной информации, важной частью тестирования является проверка обновления данных о матчах и событиях. Все тест-кейсы выполняются последовательно, чтобы убедиться в отсутствии критических ошибок. На основе полученных данных производится приложения для обеспечения лучшего пользовательского опыта.

Таблица 4.1 – Таблица описания тест-кейсов (часть 1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Предусловие: | | Запустить приложение | | |
| № | Описание | Шаги воспроизведения | Ожидаемый результат | Результат |
| 1 | Узнать на какой минуте и кем был забит гол | Открыть страницу расписания матчей  Выбрать прошедший матч  Нажать на выбранный матч  Просмотреть дерево событий  Найти события «гол»  Посмотреть время на какой минуте забит гол | Список матчей и информация про матч успешно загружен и  отображен | Пройден. Результат представлен на рисунке  Б.1 приложения Б |
| 2 | Анализ таблицы лидеров и просмотр команд, которые вылетают | Открыть страницу расписания матчей  Перейти на страницу таблицы лидеров  Проанализировать | Данные успешно загружены,  А 3 команды, которые набрали наименьшее количество очков помечены красным флагом | Пройден. Результат представлен на рисунке  Б.2 приложения Б |

Продолжение таблицы 4.1 (часть 2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Предусловие: | | Запустить приложение | | |
| № | Описание | Шаги воспроизведения | Ожидаемый результат | Результат |
|  |  | таблицы  Пролистать вниз  Просмотреть последние команды, которые  проигрывают в сезоне |  |  |
| 3 | Аунтефикация пользователя | Выйти из аккаунта, если до этого пользователь был зарегистрирован.  Ввести правильный email пользователя  Ввести правильный пароль пользователя  Нажать кнопки аунтефикации | Пользователь с таким email найден, а пароль правильный и пользователю даны права на использование приложения.  Осуществился переход на главную страницу | Пройден.  Пользователь перешел на главную страницу |
| 4 | Регистрация пользователя | Выйти из аккаунта, если до этого пользователь залогинился  На странице аунтефикации нажать на надпись «Don’t have account»  Ввести новый email пользователя,  Придумать и ввести пароль | Пользователь успешно зарегистрирован  И его данные правильно занесены в базы данных | Пройден. Результат представлен на рисунке  Б.3  приложения Б |

Продолжение таблицы 4.1 (часть 3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Предусловие: | | Запустить приложение | | |
| № | Описание | Шаги воспроизведения | Ожидаемый результат | Результат |
|  |  | Нажать кнопку регистрации |  |  |
| 5 | При запуске открывается экран с логотипом, слоганом и логотипами партнеров | Запуск приложения | При успешном запуске открывается приветственный экран, логотип приложения и партнеров корректно отображаются, а слоган соответствует действительному | Пройден. Результат представлен на рисунке Б.4 приложения |

1. Применение
   1. Общие сведения

Приложение «Спортивные Новости» можно установить на устройство вручную, используя APK-файл. Для этого сначала необходимо скачать APK-файл приложения с официального сайта разработчика или другого доверенного источника. Перед установкой следует убедиться, что в настройках устройства разрешена установка приложений из неизвестных источников. Для этого нужно открыть Настройки → Безопасность → Установка из неизвестных источников и активировать соответствующую опцию. Затем следует найти загруженный APK-файл в папке Загрузки или с помощью файлового менеджера. Открыв файл, пользователь увидит запрос на установку приложения, после чего необходимо нажать кнопку Установить.

Процесс установки займет несколько секунд, после чего появится кнопка Открыть. Запустив приложение, пользователь может пройти первичную настройку, войти в аккаунт или создать новый профиль. Важно учитывать, что при установке из APK-файла обновления приложения придется скачивать и устанавливать вручную.

При первом запуске может потребоваться предоставить разрешения, например, доступ к интернету, уведомлениям и хранилищу. Если устройство блокирует установку, рекомендуется проверить наличие свободного места, а также отключить антивирусные программы, которые могут препятствовать инсталляции.

Чтобы удалить приложение, нужно открыть Настройки → Приложения → «Спортивные Новости» и выбрать Удалить. Альтернативный способ – нажать и удерживать значок приложения на главном экране, а затем выбрать Удалить. При необходимости можно заново скачать APK-файл и повторить процесс установки.

Целью разработки приложения «Спортивные Новости» является изучение различных технологий мобильной разработки, а также создание удобного и функционального инструмента для просмотра спортивных новостей. В процессе работы над проектом исследуются и применяются современные методы работы с базами данных, взаимодействие с API для получения актуальной информации, а также подходы к разработке удобного пользовательского интерфейса.

Одной из ключевых задач является изучение принципов построения клиент-серверной архитектуры, что позволяет организовать эффективный обмен данными между приложением и удаленным сервером.

Помимо технических аспектов, проект направлен на исследование UX/UI-дизайна, что помогает создать удобный, интуитивно понятный интерфейс, соответствующий современным стандартам мобильных приложений. Отдельное внимание уделяется адаптации приложения для различных устройств, обеспечению его стабильности и высокой производительности.

Кроме того, в рамках разработки рассматриваются вопросы безопасности данных пользователей, включая методы аутентификации и защиты личной информации. Проект способствует изучению технологий push-уведомлений для своевременного информирования пользователей о новых новостях и спортивных событиях.

Финальной целью является создание готового продукта, который будет полезен спортивным фанатам, предоставляя им возможность удобно и быстро получать актуальные новости, смотреть лучшие моменты, следить за расписанием матчей и турнирными таблицами. При этом процесс разработки позволяет расширить знания в области мобильной разработки, оптимизации данных и создания качественного пользовательского опыта.

Назначение мобильного приложения

Назначение мобильного приложения «Спортивные Новости» – предоставление актуальной информации о спортивных событиях, включая новости, лучшие моменты, расписание матчей и таблицы лиг. Приложение ориентировано на любителей спорта, которым важно быть в курсе последних событий.

Потенциальная аудитория – поклонники различных видов спорта, которые хотят получать свежие новости и смотреть видеообзоры. Также приложение будет полезно аналитикам и бетторам, следящим за статистикой матчей.

Средства защиты и разграничения доступа включают аутентификацию пользователей при помощи логина и пароля, а в будущем – двухфакторную авторизацию. Данные пользователей хранятся в зашифрованном виде, а доступ к ним ограничен ролями (например, администратор, обычный пользователь). В дальнейшем панируется разработка отдельного для администраторов, при помощи которого они смогут выполнять свои действия

Области применения – спортивные медиа, аналитика матчей, персонализированные уведомления о предстоящих событиях. Класс решаемых задач включает агрегацию спортивных данных, предоставление удобного интерфейса для поиска информации и интеграцию с Instend.

Заключение

ВО время выполнения курсовой работы было изучено множество современных технологий, применяемых для разработки мобильных приложений. Особое внимание было уделено анализу различных фреймворков и инструментов, что позволило выбрать наиболее подходящие решения для реализации поставленных задач. В достаточном объеме исследована предметная область, включающая особенности работы с новостными данными и интеграции их в мобильное приложение. Были подробно изучены возможности выбранных технологий, в том числе способы обработки данных, взаимодействие с веб-ресурсами и методы отображения информации в удобном для пользователя формате. В процессе работы приобретены практические навыки в сопровождении и разработке программного обеспечения, что нашло отражение в соответствующих разделах отчета.

Индивидуальное задание было выполнено успешно, с соблюдением всех требований и сроков. В ходе выполнения задания были изучены методы загрузки данных с веб-сайтов с использованием современных API, что обеспечило оперативное получение актуальной информации. Реализована возможность обработки данных и их последующее отображение в удобном и понятном виде в приложении. Дополнительно была внедрена функция работы с таблицами и документами в формате Microsoft Word, что позволило пользователям экспортировать данные и анализировать их вне приложения.

Разработанное мобильное приложение предоставляет пользователям возможность следить за последними спортивными новостями в реальном времени, а также просматривать различную статистику по выбранным видам спорта. Серверная часть приложения была успешно протестирована в различных условиях, включая высокую нагрузку, и продемонстрировала стабильную работу при нормальных параметрах эксплуатации. В результате работы были получены практические навыки, применимые в реальных условиях разработки, что свидетельствует о высоком уровне освоения поставленных задач и успешном выполнении курсовой работы.

Список использованных источников

1. Базы данных. Проектирование и реализация / Р. Элмасри, Ш. Наватхе. – 6-е изд. – Москва: Вильямс, 2020. – 1232 с.
2. Программирование на Kotlin для Android / П. Дж. Барнс. – Москва: ДМК Пресс, 2020. – 560 с.
3. Руководство по Android Jetpack Compose / Дж. Гамильтон. – Санкт-Петербург: Питер, 2022. – 368 с.
4. Базы данных. Учебник для вузов / В. Дейт. – 8-е изд. – Москва: Вильямс, 2021. – 1120 с.
5. Основы разработки мобильных приложений для Android / А. Бреслав. – Москва: Бином, 2021. – 672 с.
6. Kotlin. Полное руководство для разработчиков / М. Уайт. – Москва: БХВ-Петербург, 2021. – 720 с.
7. Современные базы данных. Теория и практика / В. Филлипс. – Москва: Диалектика, 2019. – 928 с.
8. Программирование на Kotlin. От основ до профессионала / Э. Сэдли, В. Черепанов. – Москва: Вильямс, 2021. – 576 с.
9. Руководство по Android Jetpack Compose / [Электронный ресурс] – Режим доступа: <https://developer.android.com/jetpack/compose>. Дата доступа: 10.01.2025.
10. Kotlin Documentation / [Электронный ресурс] – Режим доступа: https://kotlinlang.org/docs/home.html. Дата доступа: 15.01.2025 .

# Приложение А

# (обязательное)

# Текст программных модулей

package com.example.sportapp

import AppActivityViewModel

import BottomNavBar

import TopAppBar

import androidx.compose.foundation.layout.padding

import androidx.compose.material3.Scaffold

import androidx.compose.runtime.Composable

import androidx.compose.runtime.collectAsState

import androidx.compose.runtime.getValue

import androidx.compose.ui.Modifier

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.unit.dp

import androidx.navigation.NavHostController

import androidx.navigation.compose.NavHost

import androidx.navigation.compose.composable

import com.example.firebaseexample.pages.SignupPage

import com.example.sportapp.models.user.AuthViewModel

import com.example.sportapp.models.viewModels.MatchesActivitySoccerViewModel

import com.example.sportapp.models.viewModels.MatchesState

import com.example.sportapp.models.viewModels.NewsActivityViewModel

import com.example.sportapp.models.viewModels.NewsSate

import com.example.sportapp.models.viewModels.VideosState

import com.example.sportapp.models.viewModels.YoutubeActivityViewModel

import com.example.sportapp.pages.FirstPage

import com.example.sportapp.pages.HomePage

import com.example.sportapp.pages.LikePage

import com.example.sportapp.pages.LoginPage

import com.example.sportapp.pages.MatchesPage

import com.example.sportapp.pages.NewsPage

import com.example.sportapp.pages.VideoPage

import com.example.sportapp.shared.Loading

sealed class Screen(val route: String) {

data object Home : Screen("home")

data object Matches : Screen("matches")

data object Video : Screen("video")

data object Like : Screen("like")

data object News : Screen("news/{newsId}")

data object FirstPage : Screen("firstPage")

data object LoginPage : Screen("login")

data object SignupPage : Screen("signup")

data object Loading : Screen("loading")

}

@Composable

fun MyAppNavigation(

authViewModel: AuthViewModel,

newsState: NewsSate,

state: MatchesState,

videoState: VideosState,

newsViewModel: NewsActivityViewModel,

appActivity: AppActivityViewModel,

navController: NavHostController,

matchesViewModel: MatchesActivitySoccerViewModel,

videoViewModel: YoutubeActivityViewModel,

) {

val showBars by appActivity.showBars.collectAsState()

Scaffold(

containerColor = Color(0xFFF6F6F6),

bottomBar = {

if (showBars) {

BottomNavBar(navController = navController)

}

},

topBar = {

if (showBars) {

TopAppBar(appActivity, authViewModel, navController)

}

}

) { innerPadding ->

val paddings = if (showBars) 8.dp else 0.dp

NavHost(

navController = navController,

startDestination = Screen.FirstPage.route,

modifier = Modifier

.padding(innerPadding)

.padding(horizontal = paddings),

builder = {

composable(Screen.LoginPage.route) {

appActivity.changeShowBars(false)

LoginPage(navController, authViewModel)

}

composable(Screen.SignupPage.route) {

appActivity.changeShowBars(false)

SignupPage(navController, authViewModel)

}

composable(Screen.Loading.route) {

Loading()

}

composable(Screen.Home.route) {

appActivity.changeShowBars(true)

HomePage(

newsState = newsState,

state = state,

videoState = videoState,

newsViewModel = newsViewModel,

matchesViewModel = matchesViewModel,

videoViewModel = videoViewModel,

navController = navController,

authViewModel = authViewModel,

)

}

composable(Screen.FirstPage.route) {

FirstPage(navController)

}

composable(Screen.Matches.route) {

MatchesPage(matchesViewModel, state, appActivity)

}

composable(Screen.Video.route) { VideoPage(appActivity) }

composable(Screen.Like.route) { LikePage(appActivity) }

composable(Screen.News.route) { backStackEntry ->

val newsDateTime = backStackEntry.arguments?.getString("newsId")

NewsPage(

appActivity,

newsDateTime!!,

newsState,

navController,

newsViewModel

)

}

}

)

}

}package com.example.sportapp

import AppActivityViewModel

import android.annotation.SuppressLint

import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.activity.viewModels

import androidx.compose.runtime.collectAsState

import androidx.compose.runtime.getValue

import androidx.lifecycle.viewmodel.compose.viewModel

import androidx.navigation.compose.rememberNavController

import com.example.sportapp.models.user.AuthViewModel

import com.example.sportapp.models.viewModels.MatchesActivitySoccerViewModel

import com.example.sportapp.models.viewModels.NewsActivityViewModel

import com.example.sportapp.models.viewModels.YoutubeActivityViewModel

class mainActivity : ComponentActivity() {

@SuppressLint("CoroutineCreationDuringComposition", "StateFlowValueCalledInComposition")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

val appActivity: AppActivityViewModel = viewModel()

val newsViewModel: NewsActivityViewModel = viewModel()

val newsState by newsViewModel.getState().collectAsState()

val matchesViewModel: MatchesActivitySoccerViewModel = viewModel()

val state by matchesViewModel.getState().collectAsState()

val videoViewModel: YoutubeActivityViewModel = viewModel()

val videoState by videoViewModel.getState().collectAsState()

val navController = rememberNavController()

val authViewModel: AuthViewModel by viewModels()

MyAppNavigation(

authViewModel = authViewModel,

newsState = newsState,

state = state,

videoState = videoState,

newsViewModel = newsViewModel,

appActivity = appActivity,

navController = navController,

matchesViewModel = matchesViewModel,

videoViewModel = videoViewModel

)

}

}

}

import androidx.compose.animation.core.animateDpAsState

import androidx.compose.animation.core.tween

import androidx.compose.foundation.background

import androidx.compose.foundation.layout.Box

import androidx.compose.foundation.layout.Column

import androidx.compose.foundation.layout.fillMaxWidth

import androidx.compose.foundation.layout.height

import androidx.compose.foundation.lazy.LazyColumn

import androidx.compose.foundation.lazy.rememberLazyListState

import androidx.compose.foundation.shape.RoundedCornerShape

import androidx.compose.runtime.Composable

import androidx.compose.runtime.LaunchedEffect

import androidx.compose.runtime.derivedStateOf

import androidx.compose.runtime.getValue

import androidx.compose.runtime.mutableStateOf

import androidx.compose.runtime.remember

import androidx.compose.runtime.setValue

import androidx.compose.ui.Modifier

import androidx.compose.ui.draw.clip

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.unit.dp

import com.example.sportapp.models.soccer.api.domain.EventEntity

import com.example.sportapp.models.soccer.api.domain.MatchEntity

import com.example.sportapp.ui.theme.Blue100

import com.example.sportapp.widgets.home.CurrentMatch

import com.example.sportapp.widgets.matchInfo.MatchEventItem

import com.example.sportapp.widgets.matchInfo.PhaseNameLine

@Composable

fun MatchInfoContent(eventsList: List<EventEntity>, match: MatchEntity) {

val lazyListState = rememberLazyListState()

val isAtTop = remember {

derivedStateOf { lazyListState.firstVisibleItemIndex == 0 && lazyListState.firstVisibleItemScrollOffset == 0 }

}

var currentHeight by remember { mutableStateOf(200.dp) }

var currentSize by remember { mutableStateOf(100.dp) }

LaunchedEffect(isAtTop.value) {

currentHeight = if (isAtTop.value) 150.dp else 70.dp

currentSize = if (isAtTop.value) 100.dp else 10.dp

}

val boxHeight by animateDpAsState(

targetValue = currentHeight,

animationSpec = if (currentHeight == 70.dp) {

tween(durationMillis = 4000)

} else {

tween(durationMillis = 4000)

}

)

LazyColumn(state = lazyListState) {

item {

Box(

modifier = Modifier

.height(boxHeight)

.clip(shape = RoundedCornerShape(16.dp))

.background(color = Color.Transparent)

.fillMaxWidth()

) {

Column {

Box(

modifier = Modifier

.height(boxHeight)

.clip(shape = RoundedCornerShape(16.dp))

.background(color = Blue100)

.fillMaxWidth()

) {

CurrentMatch(match)

}

}

}

Box(

Modifier

.fillMaxWidth()

.height(15.dp)

)

}

items(eventsList.size) { index ->

if (index == eventsList.size - 1) PhaseNameLine("1st HALF")

else {

MatchEventItem(eventsList[index], match)

}

}

}

}

package com.example.sportapp.widgets.matchInfo.matchCard.specialEventsCards

import androidx.compose.runtime.Composable

import com.example.sportapp.models.soccer.api.domain.EventEntity

import com.example.sportapp.widgets.matchInfo.matchCard.specialEventsCards.playerSubstitulation.LeftSubstitulation

import com.example.sportapp.widgets.matchInfo.matchCard.specialEventsCards.playerSubstitulation.RightSubstitulation

@Composable

fun PlayerSubstitution(event: EventEntity, side: Boolean) {

if (side) {

LeftSubstitulation(event)

} else {

RightSubstitulation(event)

}

}package com.example.sportapp.widgets.matchInfo.matchCard.specialEventsCards

import androidx.compose.runtime.Composable

import androidx.compose.ui.unit.dp

import com.example.sportapp.R

import com.example.sportapp.models.soccer.api.domain.EventEntity

@Composable

fun GoalCard (event: EventEntity, side: Boolean) {

if (side) {

LeftLeaf(

event = event,

iconId = R.drawable.ball,

iconSize = 15.dp

)

}

else {

RightLeaf(

event = event,

iconId = R.drawable.ball,

iconSize = 15.dp

)

}

}package com.example.sportapp.widgets.matches.rankigs.rankingsParts

import androidx.compose.foundation.layout.Arrangement

import androidx.compose.foundation.layout.Row

import androidx.compose.foundation.layout.fillMaxWidth

import androidx.compose.material3.Text

import androidx.compose.runtime.Composable

import androidx.compose.ui.Modifier

import com.example.sportapp.models.soccer.api.domain.RankingEntity

@Composable

fun ScorePart (modifier: Modifier,item: RankingEntity) {

Row(

modifier = modifier

.fillMaxWidth(),

horizontalArrangement = Arrangement.SpaceEvenly

) {

Text(text = item.points.toString())

Text(text = item.gamesPlayed .toString())

Text(text = item.won.toString())

Text(text = item.draws.toString())

Text(text = item.lost.toString())

Text(text = (item.goalsMade - item.goalsConceeded).toString())

}

}

package com.example.sportapp.widgets.home.currentMatchFeatures

import androidx.compose.foundation.Image

import androidx.compose.foundation.layout.Arrangement

import androidx.compose.foundation.layout.Column

import androidx.compose.foundation.layout.Row

import androidx.compose.foundation.layout.fillMaxHeight

import androidx.compose.foundation.layout.fillMaxSize

import androidx.compose.foundation.layout.size

import androidx.compose.foundation.layout.width

import androidx.compose.material3.Text

import androidx.compose.runtime.Composable

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.text.TextStyle

import androidx.compose.ui.text.style.TextAlign

import androidx.compose.ui.text.style.TextOverflow

import androidx.compose.ui.unit.dp

import coil.compose.rememberAsyncImagePainter

import com.example.sportapp.models.soccer.api.domain.MatchEntity

import com.example.sportapp.ui.theme.style12

import com.example.sportapp.ui.theme.style4

import com.example.sportapp.ui.theme.style5

import com.example.sportapp.ui.theme.style6

import java.time.ZonedDateTime

import java.time.format.DateTimeFormatter

@Composable

fun MiddleLine(

nearestMatch: MatchEntity

) {

val painterLogoA = rememberAsyncImagePainter(nearestMatch.logoUrlA)

val painterLogoB = rememberAsyncImagePainter(nearestMatch.logoUrlB)

val currentTime = ZonedDateTime.now()

val currentMatchCardText: String

val mainCardTextStyle: TextStyle

if (nearestMatch.localDateTimeMatchStart.isBefore(currentTime) || nearestMatch.matchStatus == 1){

currentMatchCardText = "${nearestMatch.goalsTeamA} : ${nearestMatch.goalsTeamB}"

mainCardTextStyle = style5

}

else {

val dateFormatter = DateTimeFormatter.ofPattern("dd MMMM")

val timeFormatter = DateTimeFormatter.ofPattern("HH:mm")

val formattedDate = nearestMatch.localDateTimeMatchStart.format(dateFormatter)

val formattedTime = nearestMatch.localDateTimeMatchStart.format(timeFormatter)

currentMatchCardText = "$formattedTime\n$formattedDate"

mainCardTextStyle = style6

}

Row(

modifier = Modifier.fillMaxSize(),

horizontalArrangement = Arrangement.SpaceEvenly,

verticalAlignment = Alignment.CenterVertically

) {

Column(

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.SpaceEvenly,

modifier = Modifier.fillMaxHeight()

) {

Image(

painter = painterLogoA,

contentDescription = "",

modifier = Modifier

.size(60.dp)

)

Text(

text = nearestMatch.teamAAcronym,

maxLines = 2,

overflow = TextOverflow.Ellipsis,

modifier = Modifier.width(100.dp),

textAlign = TextAlign.Center,

style = style4

)

}

if (nearestMatch.matchStatus == 1){

Column (

verticalArrangement = Arrangement.Center,

horizontalAlignment = Alignment.CenterHorizontally,

) {

Text(

text = currentMatchCardText,

style = mainCardTextStyle,

)

Text(

text = nearestMatch.minute.toString() + "'",

style = style12,

)

}

}

else {

Text(

text = currentMatchCardText,

modifier = Modifier.align(Alignment.CenterVertically),

style = mainCardTextStyle

)

}

Column(

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.SpaceEvenly,

modifier = Modifier.fillMaxHeight()

) {

Image(

painter = painterLogoB,

contentDescription = "",

modifier = Modifier

.size(60.dp),

)

Text(

text = nearestMatch.teamBAcronym,

maxLines = 2,

overflow = TextOverflow.Ellipsis,

textAlign = TextAlign.Center,

modifier = Modifier.width(100.dp),

style = style4

)

}

}

}

# Приложение Б

**(справочное)**

# Результаты работы программы



Рисунок Б.1 – Результат просмотра на какой минуте был забит гол



Рисунок Б.2 – Результат просмотра турнирной таблицы и 3 команды, которые вылетают

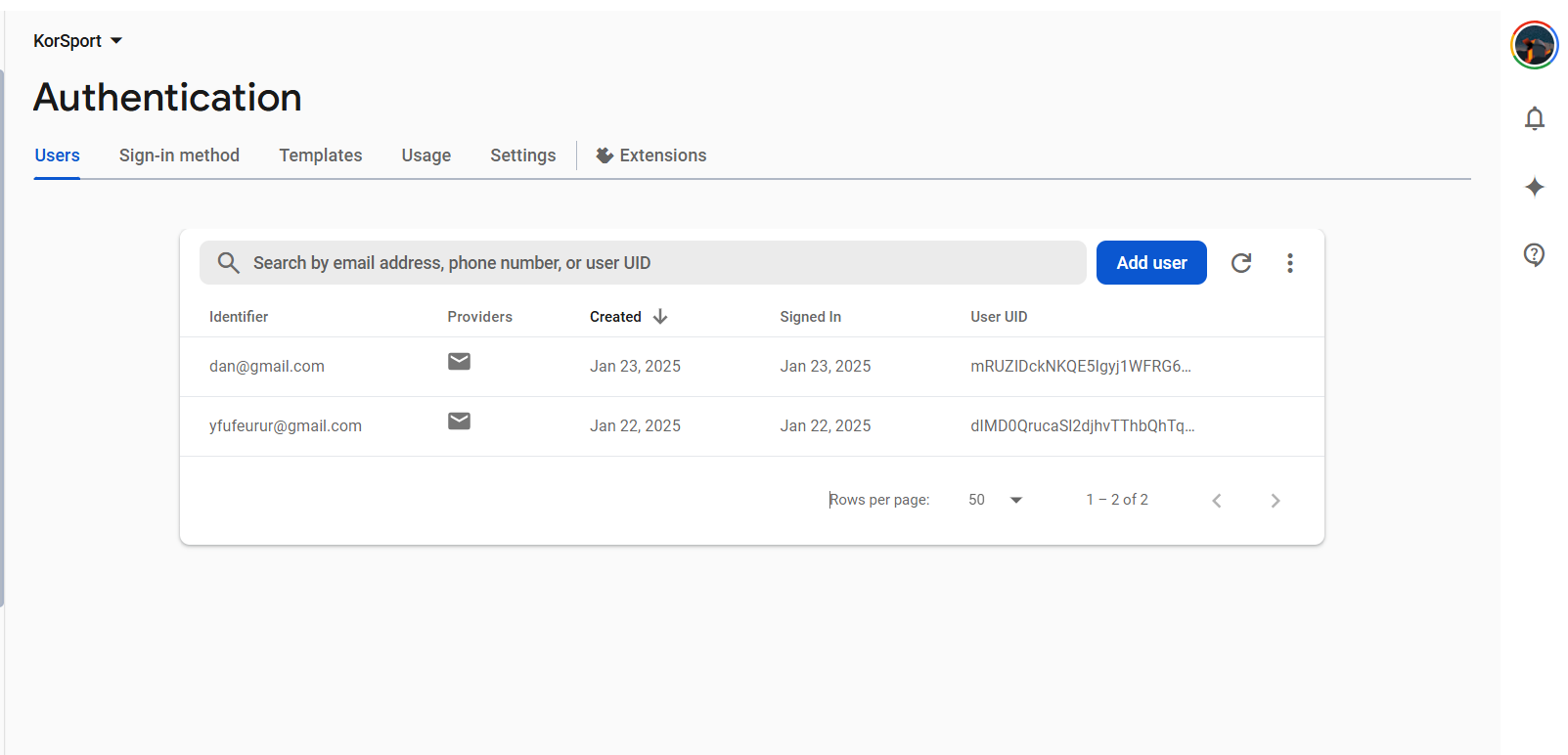


Рисунок Б.3 – Результат регистрации и пользователя

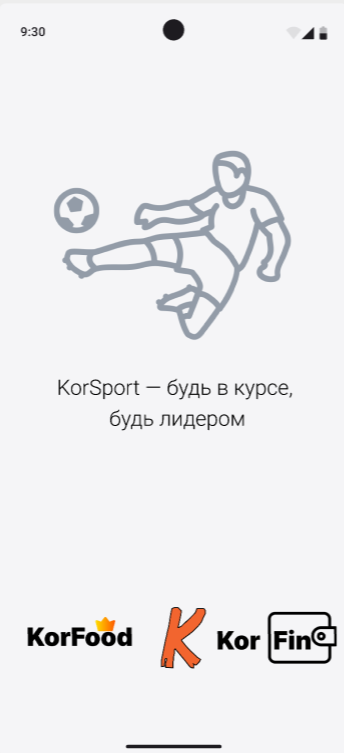


Рисунок Б.4 – Приветственный экран