Титульный лист

Задание на курсовую работу

**ОГЛАВЛЕНИЕ**

[**Введение** 4](#_Toc167206815)

[**1.** **Анализ предметной области** 5](#_Toc167206816)

[**2.** **Анализ существующих аналогов** 8](#_Toc167206817)

[**3.** **Техническое задание** 10](#_Toc167206818)

[**4.** **Архитектура** 10](#_Toc167206819)

[**5.** **Описание и обоснование выбора ПО** 12](#_Toc167206820)

[**6.** **Используемая СУБД** 16](#_Toc167206821)

[**7.** **Wireframe приложения** 17](#_Toc167206822)

[**8.** **Инструкция пользователя** 18](#_Toc167206823)

[**9.** **Тестирование на физическом устройстве** 19](#_Toc167206824)

[**Заключение** 20](#_Toc167206825)

[**Список используемой литературы** 21](#_Toc167206826)

# **Введение**

Компьютерные игры давно стали неотъемлемой частью современной культуры, привлекая миллионы людей по всему миру, включая Россию. Это уникальное явление, соединяющее людей разных возрастов, профессий и интересов. Тема для курсовой работы “Игровой календарь” была выбрана не случайно, несмотря на широкую аудиторию и высокий спрос, я не смог найти решения для отслеживания дат выхода новых игр и организации личных игровых списков.

Отсутствие такого инструмента создает определенные неудобства для геймеров, которые стремятся быть в курсе последних новинок и управлять своими игровыми достижениями. В свете этого, целью данной курсовой работы является разработка мобильного приложения, которое не только поможет игрокам следить за актуальными и предстоящими релизами, но и предоставит возможность вести учет своих игровых достижений.

Создание “Игрового календаря” предоставит людям инструмент, который не только удовлетворит их запросы, но и повысит их вовлеченность и удовольствие, получаемое от игр.

**Цель курсовой работы**

Разработать мобильное приложение календарь – данное приложение должно будет помочь отслеживать свои списки, а также быть в курсе последних и ближайших релизов. Приложение должно быть реализовано на языке высокого уровня Java.

**Задачи курсовой работы**

1. проанализировать предметную область мобильного приложения;
2. сделать обзор существующих аналогов разрабатываемого приложения;
3. cформировать техническое задание на разработку программы в соответствии с ГОСТ 19.201-78;
4. описать архитектуру программной системы, привести структурную и функциональную диаграммы, схему базы данных;
5. спроектировать интерфейс мобильного приложения;
6. реализовать код программы на языке высокого уровня Java, протестировать его и отладить;
7. реализовать контрольный пример работы программы, начиная с открытия, показать все этапы работы вашего приложения.
8. **Анализ предметной области**

Календарь специализируется не только в отслеживании анонсированных игр, вызывающих интерес в игровом сообществе, но и предоставляет пользователям возможность организовывать свои личные списки ожидаемых релизов. Это позволяет геймерам оставаться в курсе последних новостей и быть уверенными, что они не пропустят выход игр, которые они с нетерпением ждут.

Для эффективного функционирования календаря требуется база данных, которая будет способная хранить информацию. Данная база данных должна включать в себя:

1. Детальные сведения об игре: название, жанр, разработчик, издатель и описание;
2. Текущее состояние игры: в разработке или уже можно играть;
3. Дата выхода игры: точная дата или предполагаемый период релиза.
4. Информация о пользователе: всевозможные данные пользователя (его статус, никнейм и фотография профиля);
5. Состояние игры в списках у пользователя: отметки ожидания, прохождения, избранное и т.д.

База данных строиться с учетом следующих особенностей:

1. Состояние игры должно меняться в реальном времени;
2. Каждый пользователь может сохранять в свои списки по несколько игр;
3. Каждая игра включает в себя связанные с этой релизы.

Администраторы базы данных играют ключевую роль в поддержании актуальности и точности хранимой информации. Их задачи включают в себя:

1. Добавление новых игр в календарь, обеспечивая, что данные о новых анонсах быстро становятся доступны пользователям;
2. Обновление информации об играх: изменение дат, добавление новостей о разработке, редактирование описаний и т.д.;
3. Мониторинг и управление списками пользователей: просмотр состояний игр в списках, отслеживание популярности игр среди пользователей.
4. **Анализ существующих аналогов**

Для анализа существующих аналогов я выбрал два широко распространённых календаря: международный Google Calendar и разработанный российскими специалистами Яндекс.Календарь. Оба этих календаря обладают своими положительными и отрицательными сторонами. Рассмотрим каждый из них по подробней.

Яндекс Календарь — это бесплатный персональный информационный менеджер от компании Яндекс. Он позволяет планировать дела с различными приоритетами, устанавливать напоминания, создавать несколько календарей на разные темы, а также использовать его совместно с другими пользователями. Яндекс Календарь интегрирован с другими сервисами Яндекс.

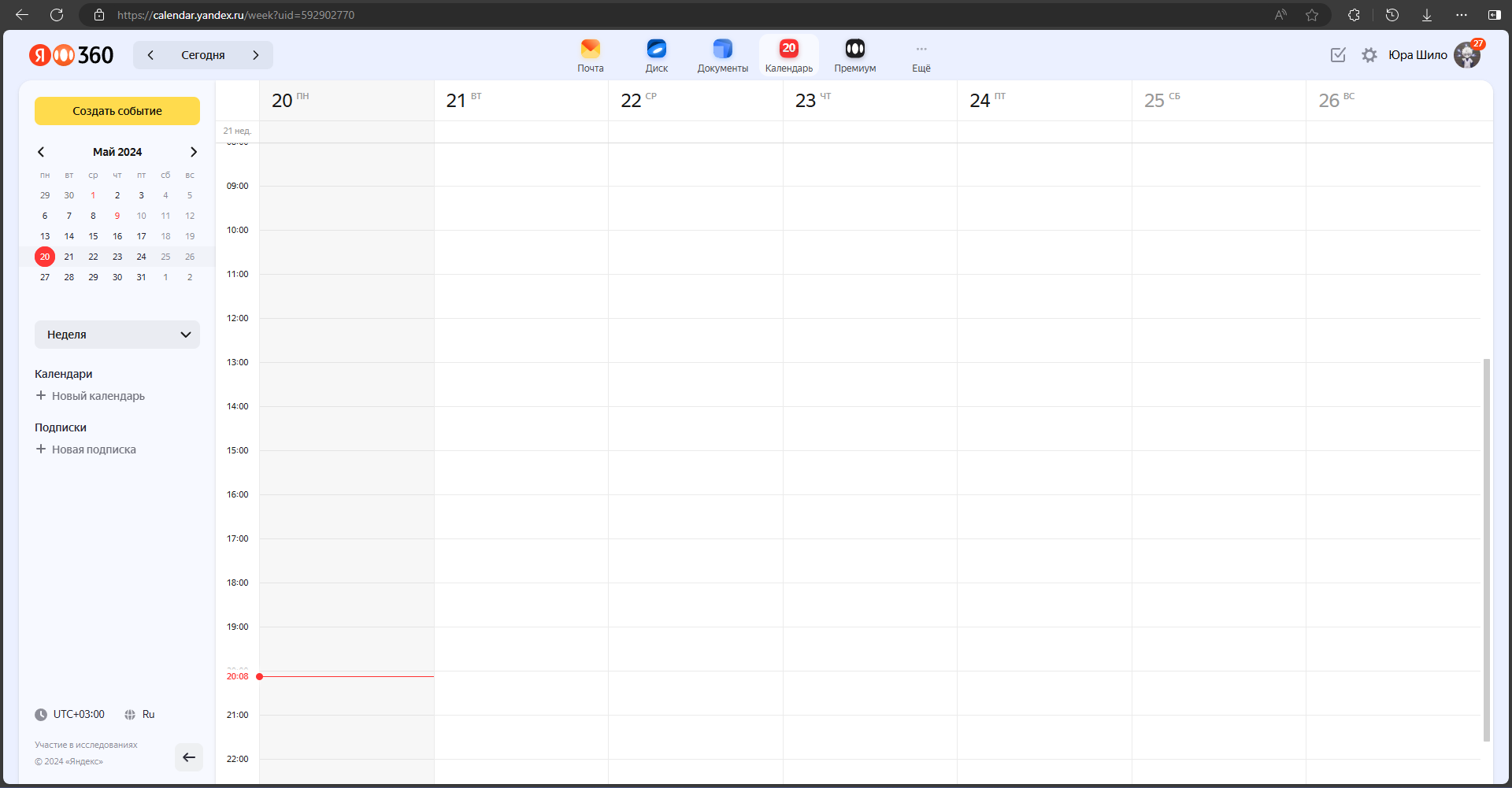
Плюсы:

* Интеграция с экосистемой Яндекс обеспечивает синхронизацию с картами, погодой и афишей, что делает планирование более удобным;
* Возможность создания списков задач и настройки напоминаний.

Минусы:

* В сравнении с международными аналогами функциональность может быть ограниченной;
* Не всегда приходят push уведомления на телефон.

На рисунке 1 предоставлен внешний вид веб версии Яндекс календаря.



*Рисунок 1. – Интерфейс веб версии Яндекс календаря*

Google Календарь, разработанный компанией Google, предоставляет возможность планировать встречи, события и дела. Google Календарь позволяет задавать время встречи, создавать повторяющиеся мероприятия, устанавливать напоминания и приглашать других участников с уведомлениями по электронной почте. Его можно использовать для совместного планирования встреч и событий, а также синхронизировать с устройствами.

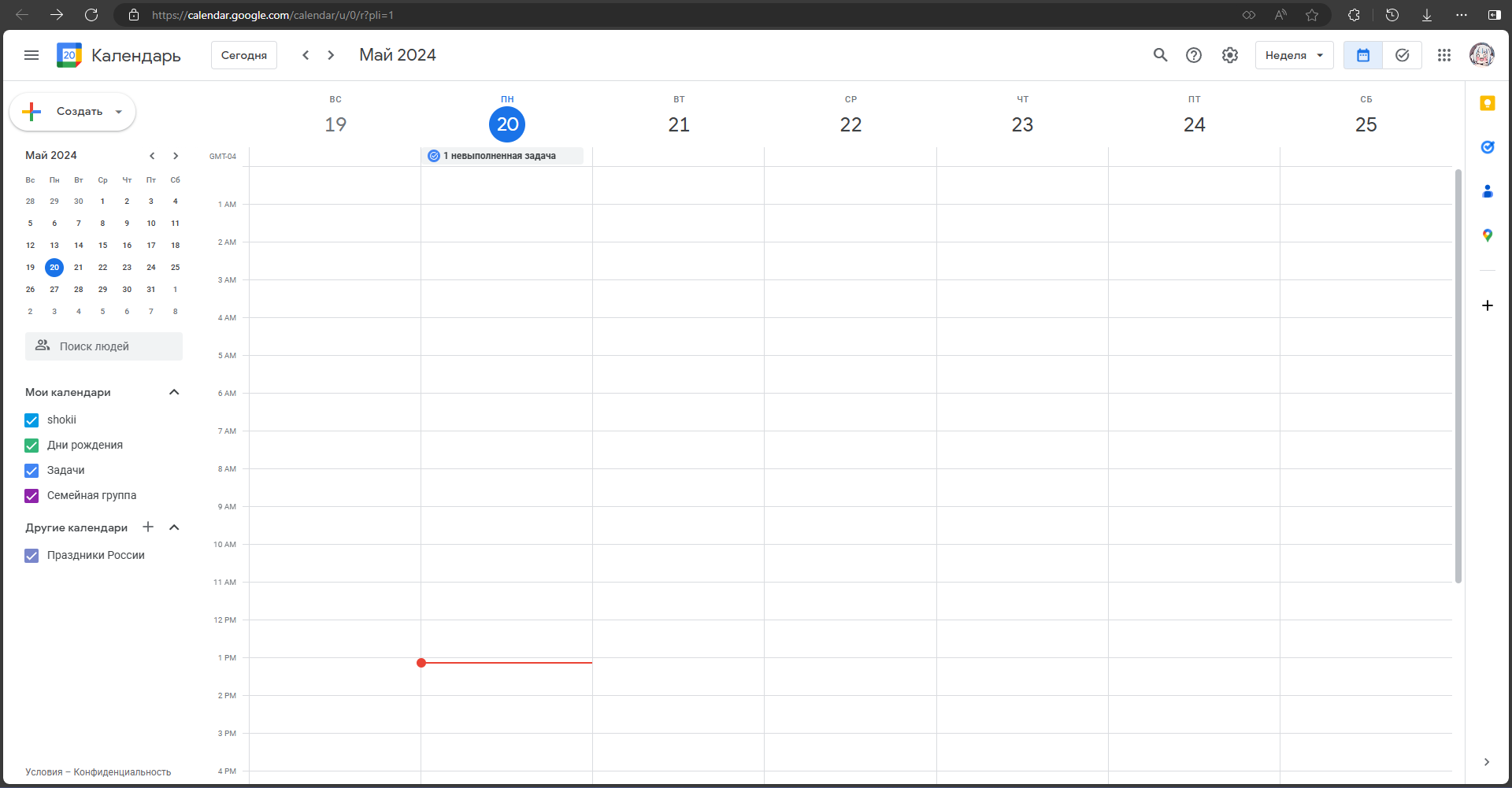
Плюсы:

* Возможность одновременного управления несколькими календарями, что идеально подходит для разделения личных и рабочих задач;
* Интеграция с Gmail и Google Tasks позволяет легко превращать письма в задачи и отслеживать их выполнение;
* Автоматическое добавление событий из электронной почты, таких как билеты на концерты или бронирование отелей, облегчает планирование.

Минусы:

* Некоторые продвинутые функции доступны только в рамках платных подписок Google Workspace, что может быть барьером для индивидуальных пользователей;
* Интеграция с корпоративными системами может быть затруднена без специализированных настроек.

На рисунке 2 предоставлен внешний вид веб версии Googleкалендаря.



*Рисунок 2. – Интерфейс веб версии Google календаря*

Проведя анализ этих двух продуктов, я выяснил, что два этих календаря, не содержат функционал, который требуется в моем приложении. Но несмотря на это я приступил к написанию технического задания для моего календаря. Моя цель создать приложение, которое бы объединило лучшие качества обеих систем, стараясь исключать их недостатки.

1. **Техническое задание**

*(пу-пу-пу)*

1. **Архитектура**

Для создания эффективной модели базы данных, которая будет служить основой для календаря игровых релизов, мы разработаем диаграмму IDEF0. Эта диаграмма поможет нам визуализировать и структурировать процессы взаимодействия данных и пользователей в системе.

На рисунке 3 представлен самый верхний уровень нашей схемы. Он иллюстрирует ключевые потоки информации и взаимодействия между различными компонентами системы:

Входы:

1. Информация об игре: включает в себя все данные об играх, такие как название, разработчик, издатель, описание и дата релиза;
2. Данные пользователя: личная информация, никнейм, фото профиля, списки ожидания и пройденных игр;
3. Текущее время сервера: используется для тайминга уведомлений и обновлений статусов игр.

Механизмы:

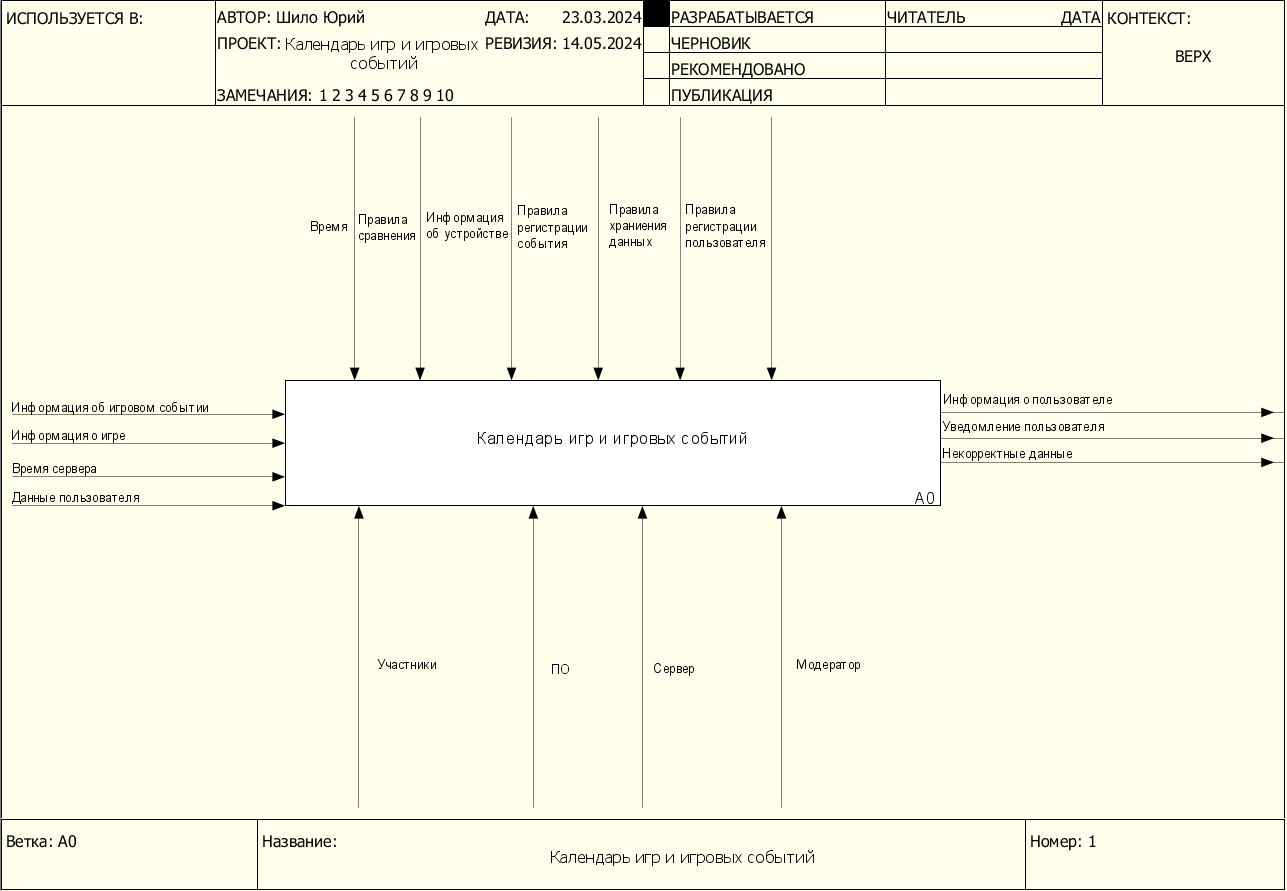
1. Участники: активные пользователи системы, которые ведут свои списки и взаимодействуют с календарем;
2. Сервер: обеспечивает обработку и хранение данных;
3. Модератор: отвечает за актуализацию данных.

Управляющие:

1. Время: определяет текущую дату и время;
2. Правила сравнения: используются для сравнения;
3. Правила регистрации игры и пользователя: определяют процесс добавления новых пользователей и игр в систему.

Выходы:

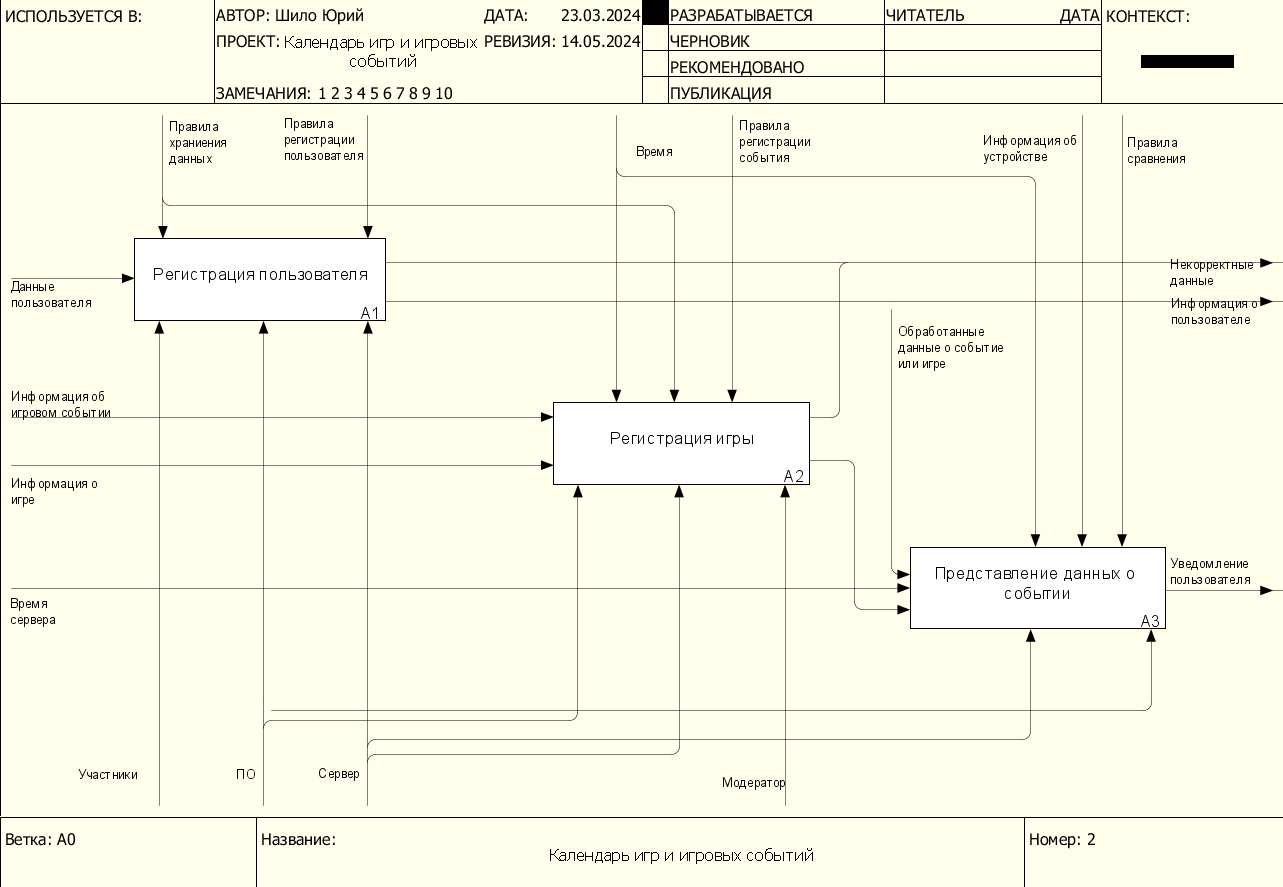
1. Информация о пользователе: обновленные данные о пользователях;
2. Уведомления о выходе игры: оповещения, отправляемые на устройства пользователей о скором релизе ожидаемых игр;
3. Сообщения об ошибках: система уведомит пользователя в случае обнаружения некорректно введенных данных или других проблем.



*Рисунок 3. – Верхний уровень построенной IDEF0 диаграммы.*

*(Поменять вход Данные пользователя на Информация о пользователе. Поменять название основного блока.)*

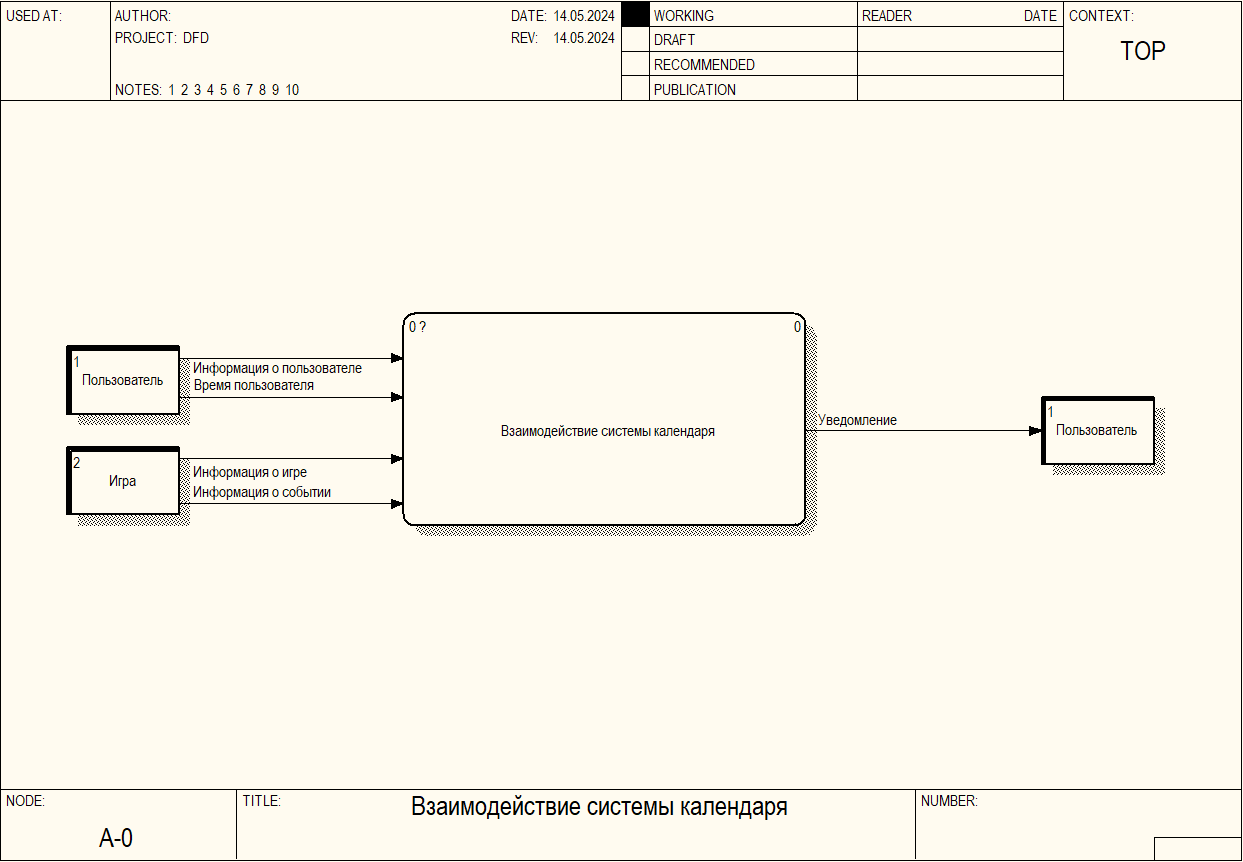
Для более глубокого понимания нашей системы рассмотрим её на более низком уровне. На этом уровне будет демонстрироваться процесс регистрации пользователя и игры, а также механизм отправки уведомлений. Взаимодействие каждого блока предоставлена на рисунке 4.



*Рисунок 4. – Декомпозиция IDEF0 диаграммы.*

Для того чтобы иметь более полное представление о реализуемой системе, мы применяем DFD (Data Flow Diagram) нотацию для нашего приложения. Это позволяет нам визуализировать и анализировать все потоки данных, проходящие через различные компоненты системы.

На рисунке 5 представлен самый первый слой реализуемой системы.

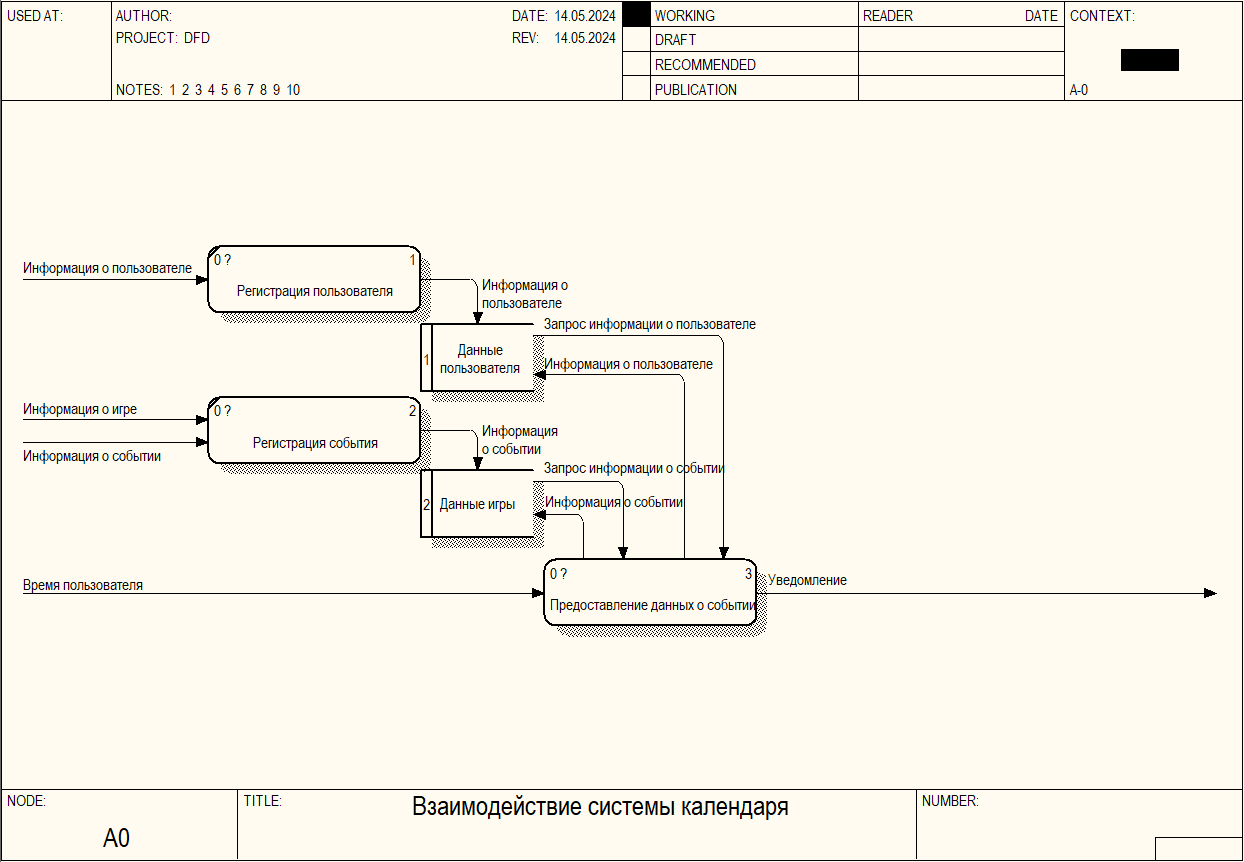


*Рисунок 5.* – *DFD нотация для нашей системы на верхнем уровне.*

*(Поменять название рисунка.*

*Поменять название основного блока.)*

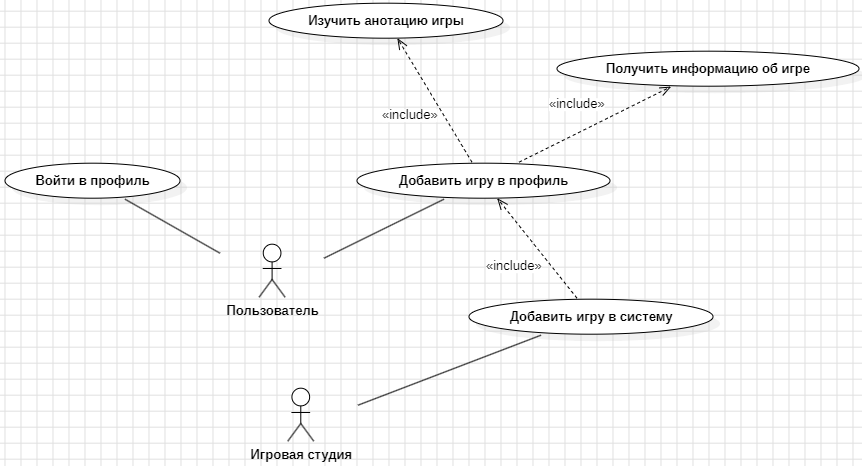
На рисунке 6, предоставлен более детальный уровень нашей системы, где каждый компонент и связь между ними рассматривается более тщательно. На этом уровне DFD мы можем увидеть, как взаимодействуют отдельные модули, какие конкретные данные они обменивают и какие преобразования с ними происходят.



*Рисунок 6. – DFD нотация для нашей системы на нижнем уровне.*

*(Поменять название рисунка.)*

Также создадим Use Case диаграмму для нашего приложения. (пу-пу-пу)



*Рисунок 7. – Use Case диаграмма реализуемого приложения*

1. **Описание и обоснование выбора ПО**

В качестве платформы для разработки был выбран Android. (пу-пу-пу)

*(Логотип или что-то подобное)*

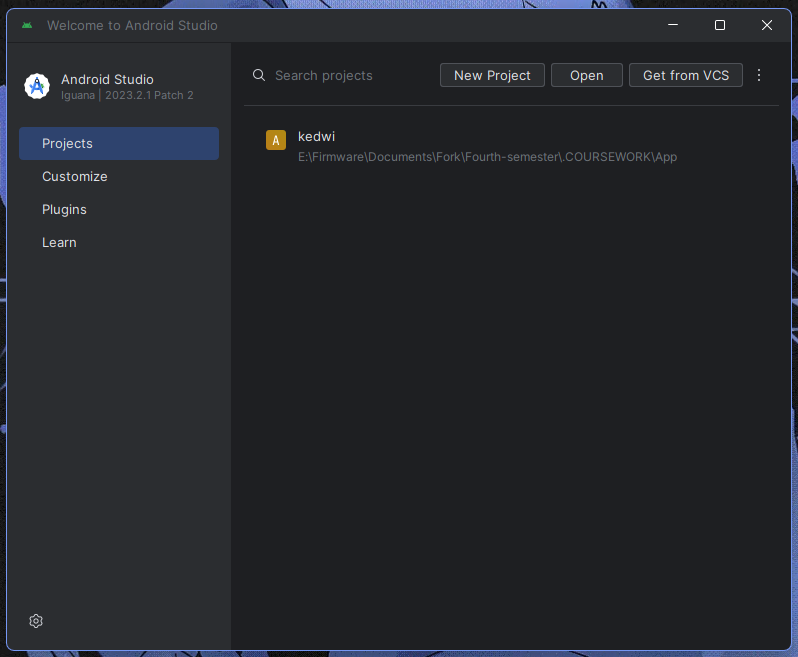
Для разработки приложения использовался язык программирования Java. (пу-пу-пу)

*(Логотип или что-то подобное)*

Для написания кода нашего приложения мной было использовано программа Android Studio. Android Studio — это официальная интегрированная среда разработки (IDE), предназначенная для разработки приложений Android. Это мощный инструмент, который предоставляет все необходимые функции для проектирования, написания, тестирования и отладки приложений. Он включает в себя редактор кода, который поддерживает языки программирования Kotlin, Java и C/C++, а также обширный набор инструментов для профилирования производительности и анализа использования ресурсов приложения.

В Android Studio также имеется встроенный эмулятор, который был использован для части тестов нашего приложения. Эмулятор поддерживает различные конфигурации устройств и версии Android, что позволило протестировать мое мобильное приложение на широком спектре устройств.

Функция установки приложения на телефон напрямую очень сильно упростила процесс тестирования и отладки, позволив быстро и легко загружать новые версии приложения на физическое устройство. На рисунке 8 показан приветственный экран Android Studio.



*Рисунок 8. – Окно, запускаемое при входе в Android Studio.*

Для создания уникальной иконки для нашего приложения, я выбрал инструмент графического дизайна – Adobe Illustrator. Это приложение от известной компании Adobe является стандартом в индустрии для создания векторной графики. С его помощью можно разрабатывать логотипы, иконки, эскизы, типографику и сложные иллюстрации.

Первое, что видит пользователь это иконка поэтому она должна быть запоминающейся. Именно поэтому я решил, что центральным элементом нашего логотипа станет кошка. Кошки пользуются большой любовью у людей по всему миру.

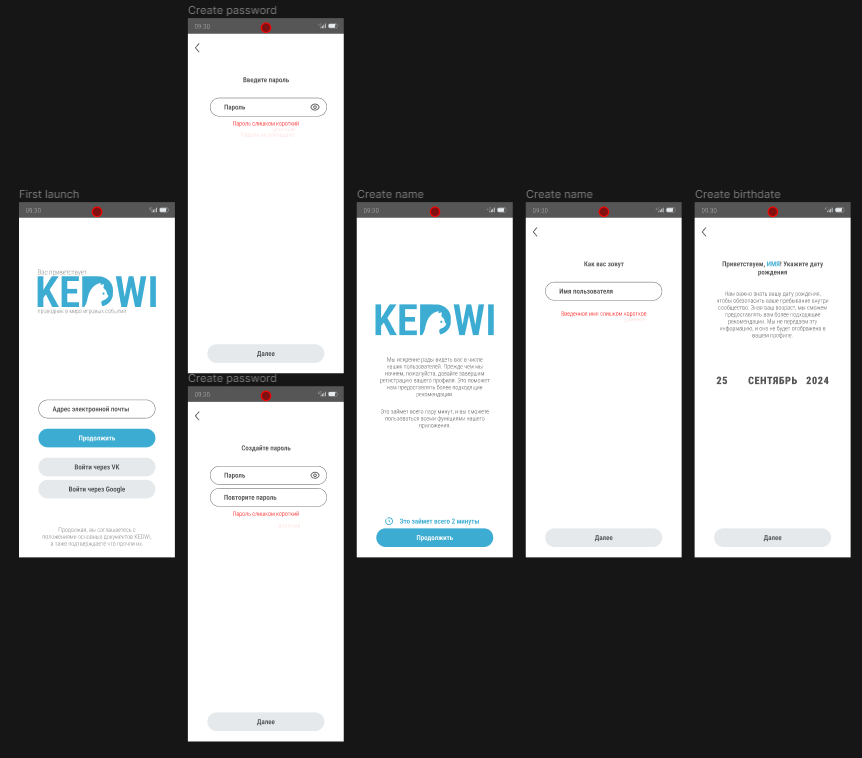
На рисунке 9, представлен итоговый вариант иконки.



*Рисунок 9. – Логотип мобильного приложения.*

Для создания красивого и функционального интерфейса нашего приложения, мы воспользовались программой Figma. Figma — это современный инструмент для совместного проектирования интерфейсов, который позволяет дизайнерам и разработчикам работать вместе в реальном времени. Это облачное приложение используется для создания, тестирования и развертывания дизайна интерфейса или продукта.

В моем проекте использовалась Figma для того, чтобы визуализировать идеи и создать дизайн, который будет выглядеть привлекательно. На рисунке 8 представлены некоторые из окон нашего приложения, дизайн которых был спроектирован в Figma.

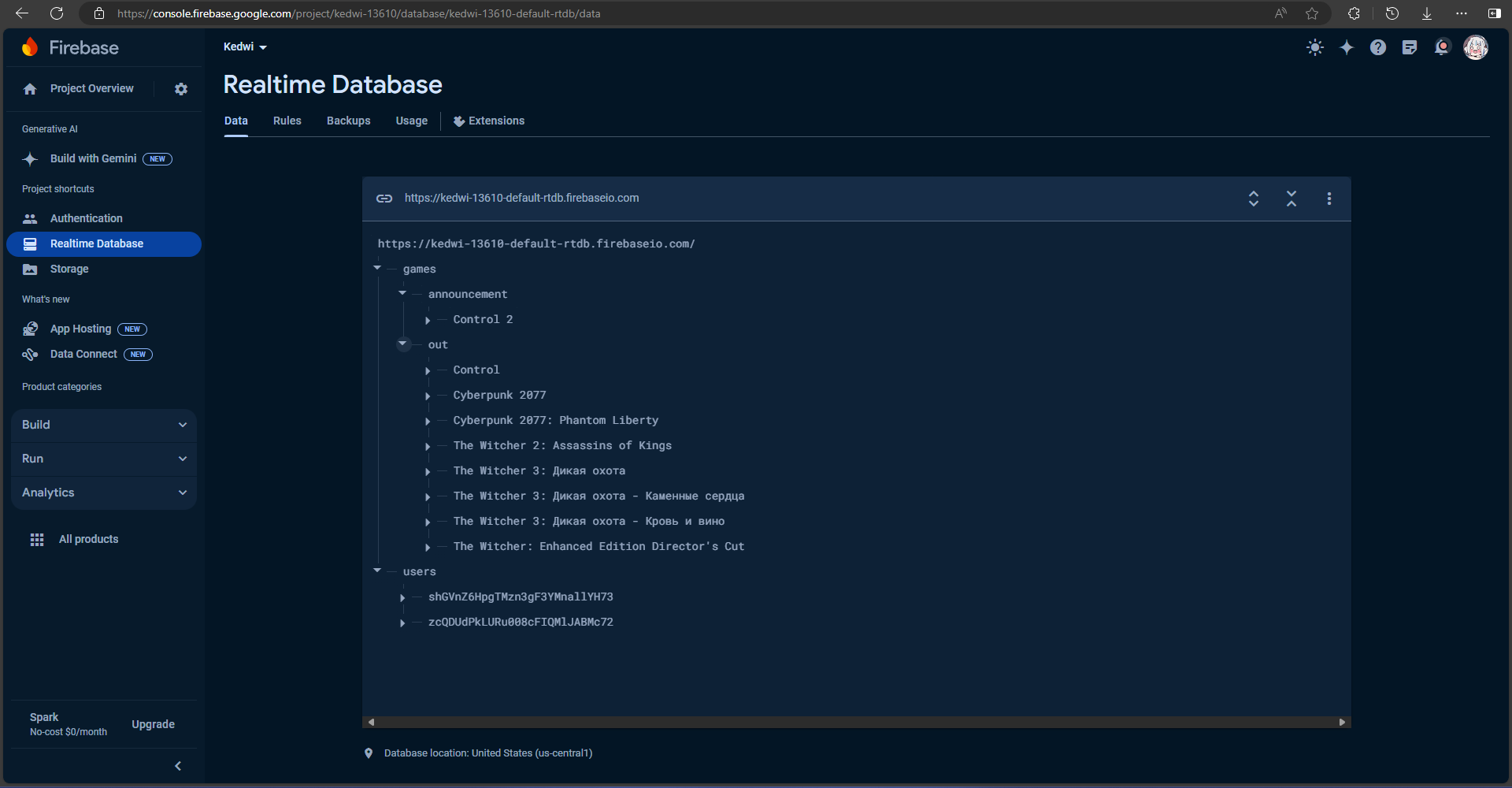


*Рисунок 10. – Дизайн окон регистрации, построенный в Figma.*

В качестве системы управления базами данных была использована Firebase. Firebase — это облачная база данных, которая позволяет хранить и синхронизировать данные в реальном времени между пользователями приложения. Это решение от компании Google, предоставляющее мощный набор инструментов для разработки кроссплатформенных приложений. Firebase не только упрощает процесс хранения данных, но и обеспечивает их безопасность, используя правила безопасности и валидацию данных на стороне сервера.

В нашем случае, Firebase используется для хранения всей необходимой информации о пользователях и играх. Для пользователей это такие данные, как никнейм, фотографию профиля, статус и списки, которые могут содержать избранные игры. Что касается информации об играх, то здесь хранятся название игры, дата выхода, разработчик, издатель, описание и рейтинг PEGI — система рейтинга, которая помогает понять, подходит ли игра для определенной возрастной категории.

На рисунке 11 показан интерфейс сайта с моей базой данных.

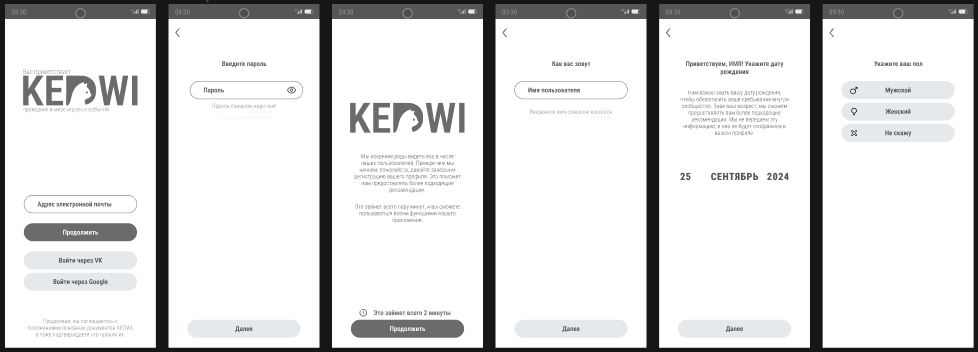


*Рисунок 11. – Интерфейс веб сайта с моей базой данных.*

1. **Wireframe** **приложения**

Перед тем как приступить к написанию разметки и кода нашего приложения я создал наброски в Figma. Wireframe — это схема с низким уровнем детализации, которая визуализирует структуру и содержание цифрового проекта. Wireframe показывает, как будут расположены все основные элементы продукта: навигация, карточки, текстовые блоки, иллюстрации и кнопки.

Первым шагом в процессе разработки моего мобильного приложения стало создание wireframe для процессов регистрации и авторизации пользователей. На рисунке 1 представлен первичный черновой макет, который демонстрирует расположение ключевых элементов интерфейса.



*Рисунок 12. – Wireframe окно регистрации*

После успешного создания wireframe для процессов регистрации и входа, я перешёл к разработке основных окон приложения. Основное окно было спроектировано, так чтобы пользователь мог просматривать списки недавно вышедших и скоро выходящих игр. Эта разметка послужила основой для дальнейшего дизайна пользовательских списков, где можно отслеживать интересующие игры.

Завершающим этапом стало создание wireframe для пользовательского профиля, где предусмотрены разделы для персональной информации, настроек и статистики. На Рисунке 2 представлены все перечисленные wireframe.



*Рисунок 12. – Wireframe для пользовательских окон*

1. **Инструкция пользователя**

После того как будет установлено и открыто приложение, пользователю будет предоставлено приветствует окно, в котором вам будет предложено поле для ввода электронной почты. После ввода электронной почты система проведет проверку, существует ли уже профиль с таким электронным адресом. Если профиль был найден, пользователь будет перенаправлены на страницу, где необходимо ввести пароль для входа в аккаунт.

Если же учетная запись с введенной электронной почтой не обнаружена, пользователю будет предложено зарегистрироваться, создав уникальный пароль. После этого, человека переносит на страницу, где я выражаю радость от того, что пользователь почти зарегистрировался в приложении, но для завершения профиля ему необходимо ввести дополнительные данные, такие как имя, дату рождения и пол.

После завершения регистрации, так же, как и после успешного входа в существующий аккаунт, пользователь будет перенаправлены на главный экран приложения. Где будет происходить последующие действия.

На главном экране нашего приложения, в нижней части расположена панель навигации. По нажатию на иконки, которой пользователь сможет легко переключаться между всеми существующими разделами.

На домашней странице расположены списки игр, как ожидаемых новинок, так и уже выпущенных игр. При нажатии на их логотипы пользователя перебрасывает на страницу, где описывается игра ее разработчик, издатель, дата выхода, описание и многое другое

На странице календаря пользователь сможет найти отслеживаемые игры, которые скоро выйдут и были добавлены в список отслеживаемых.

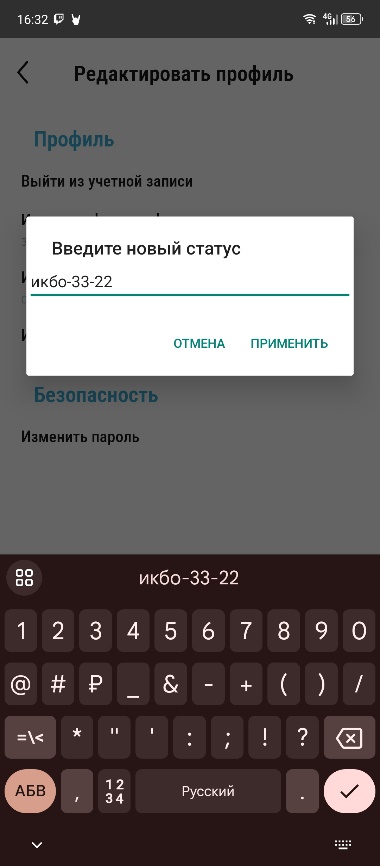
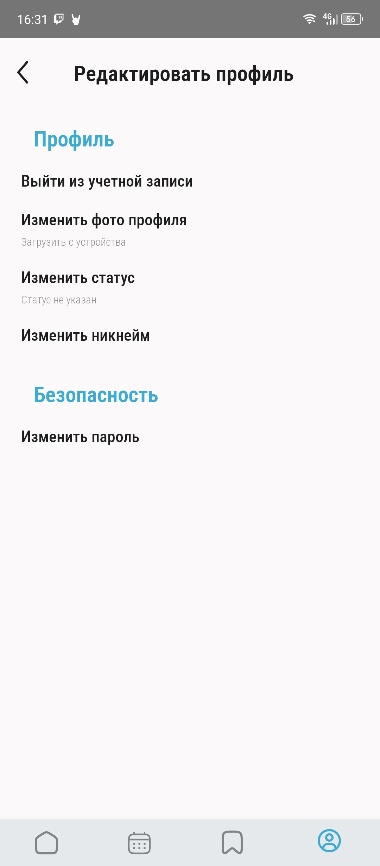
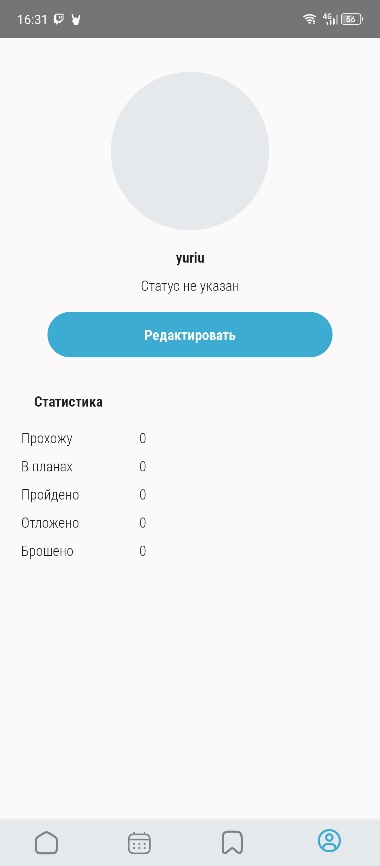
В разделе пользовательских списков пользователь сможет взаимодействовать с играми, которые были отмечены им как интересные, в процессе игры или уже пройденные.

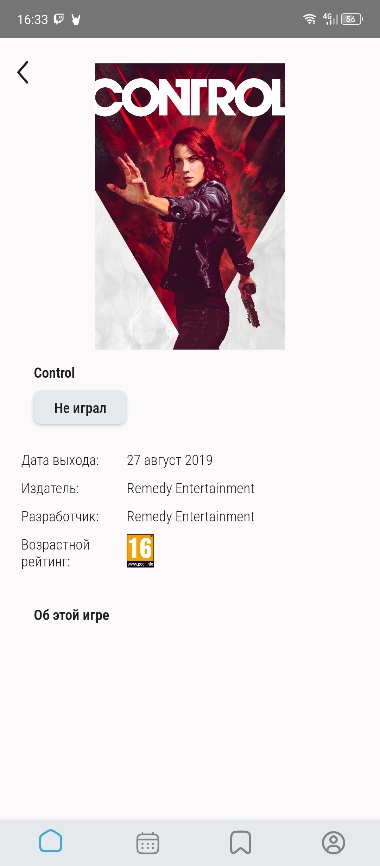
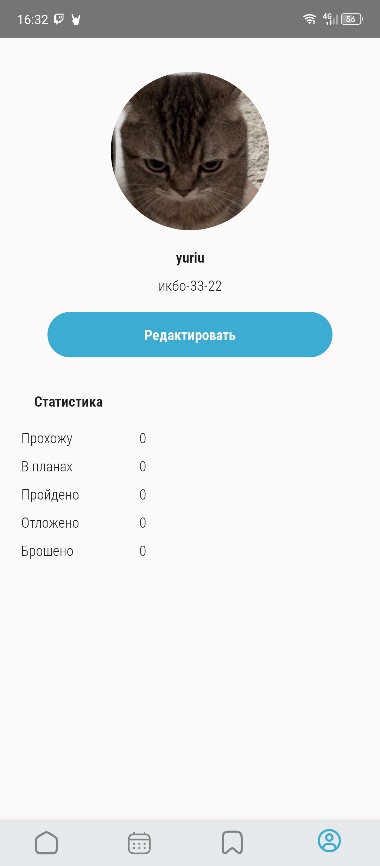
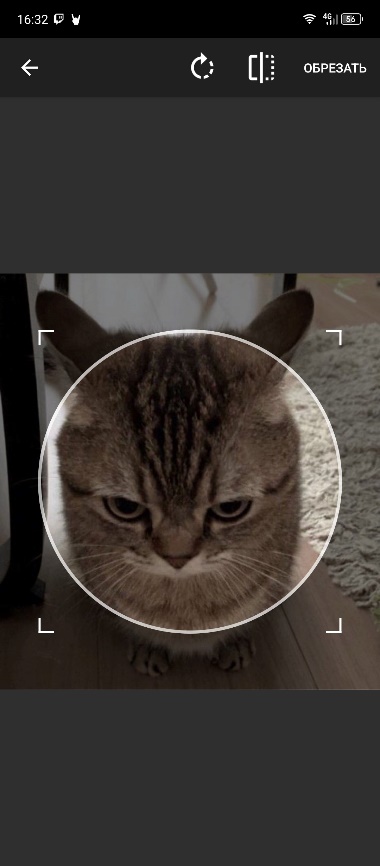
В профиле отображается вся важная информация о пользователе: фото, никнейм, статус в сообществе, а также количество игр в каждом из пользовательских списков. Здесь же расположена кнопка для редактирования профиля, чтобы была возможность обновить данные в любое время.

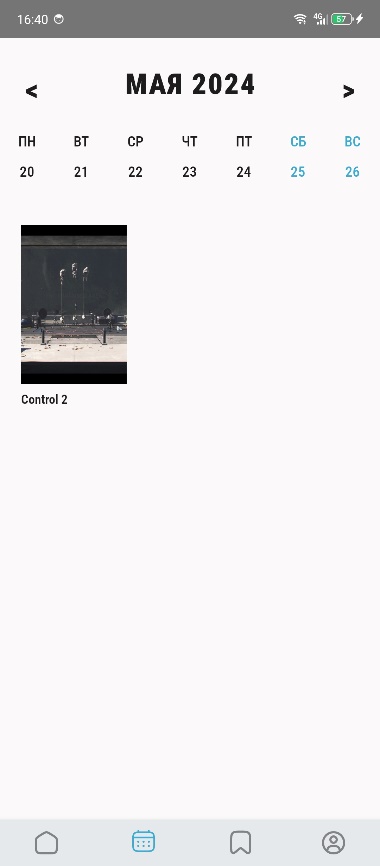
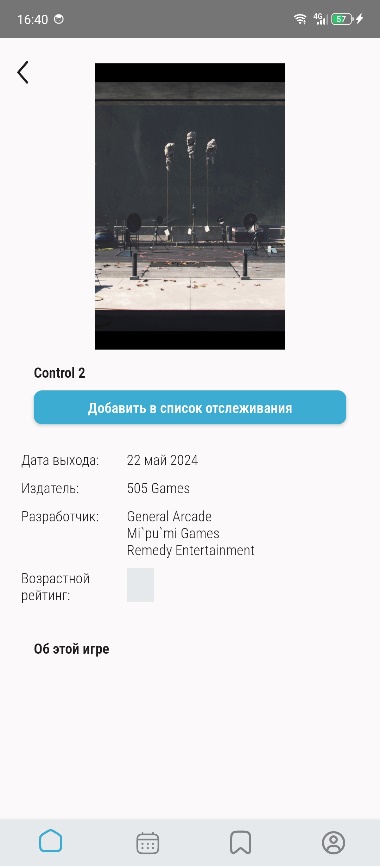
1. **Тестирование на физическом устройстве**

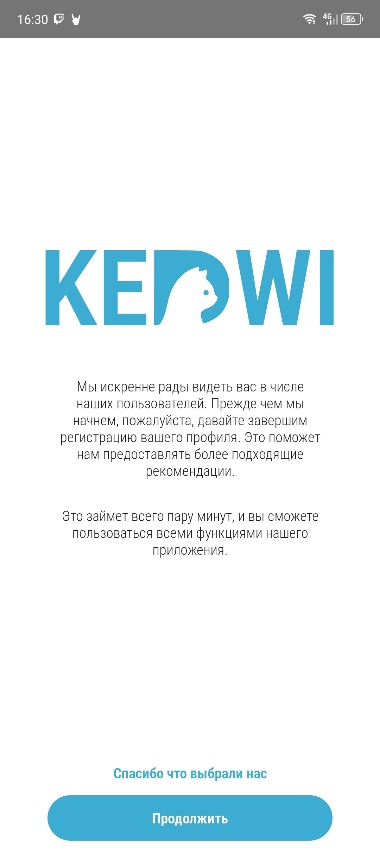
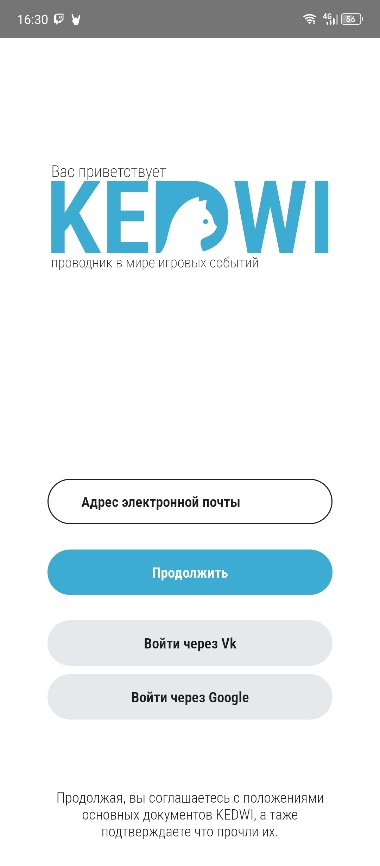
Чтобы обеспечить максимальную надежность и эффективность, важно провести тщательное тестирование программного обеспечения. Поэтому я решил выполнить запуск приложения на реальном устройстве, что позволит не только проверить корректность работы программы в условиях, максимально приближенных к реальным, но и оценить её производительность и удобство использования конечными пользователями.

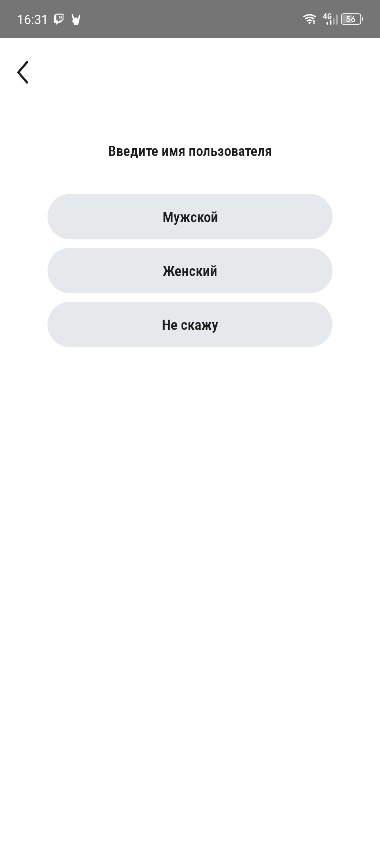
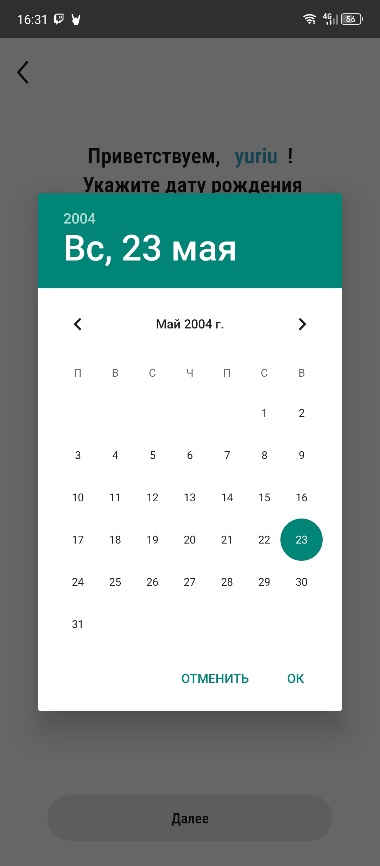
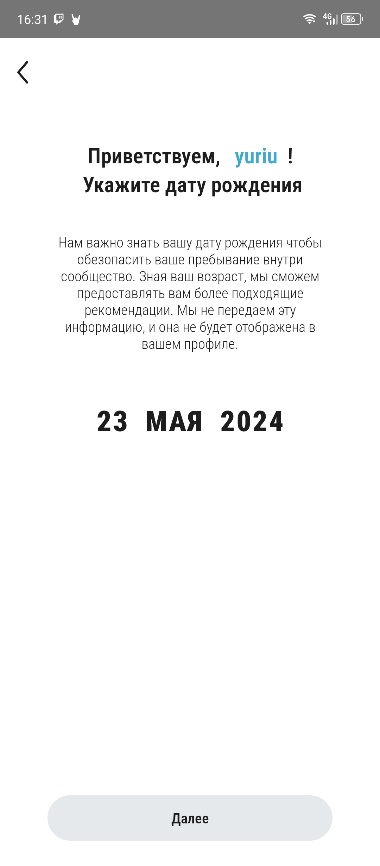
Для этих целей был выбран смартфон Infinix Note 30, работающий под управлением операционной системы Android версии 14.

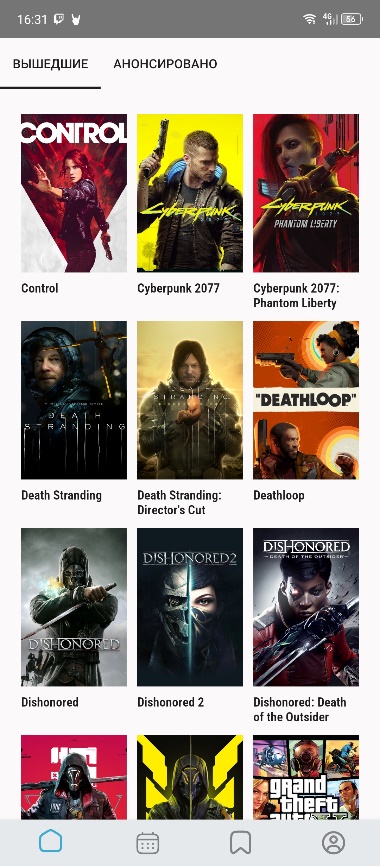












*(Расписать, что за страница, что на ней происходит)*

# **Заключение**

# **Список используемой литературы**

1. StackOverflow / [Электронный ресурс] //: [сайт]. — URL: https://stackoverflow.com/ (дата обращения: 20.05.2024).
2. Android Tools / [Электронный ресурс] //: [сайт]. — URL: https://android-tools.ru/ (дата обращения: 20.05.2024).
3. Developer Android / [Электронный ресурс] //: [сайт]. — URL: https://developer.android.com/ (дата обращения: 20.05.2024).
4. Glide / [Электронный ресурс] //: [сайт]. — URL: https://bumptech.github.io/glide/ (дата обращения: 20.05.2024).
5. Geeksforgeeks / [Электронный ресурс] //: [сайт]. — URL: https://www.geeksforgeeks.org/ (дата обращения: 20.05.2024).
6. YouTube / [Электронный ресурс] //: [сайт]. — URL: https://www.youtube.com/ (дата обращения: 20.05.2024).
7. Firebase / [Электронный ресурс] //: [сайт]. — URL: https://firebase.google.com/docs?hl=en (дата обращения: 20.05.2024).
8. Github / [Электронный ресурс] //: URL: https://github.com/ (дата обращения 20.05.2024)
9. ГОСТ 7.32—2017 Система стандартов по информации, библиотечному и издательскому делу ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ Структура и правила оформления — М.: Стандартинформ, 2017. — 32 с.
10. Клифтон Я. Проектирование пользовательского интерфейса в Android. 2-е изд. – М.: ДМК-Пресс, 2017. – 452 с.
11. Колисниченко Д.Н. Программирование для Android. Самоучитель. - 3-е изд. – СПб.: БХВ-Петербург, 2021. – 288 с.
12. Рудаков А.В. Технология разработки программных продуктов. Практикум: учеб. пособие для студ. - 4 -е изд. – М.: Издательский центр «Академия», 2014. – 192 с.
13. Эванс Б., Флэнаган Д. Java. Справочник разработчика. - 7-е изд. – СПб.: ООО «Диалектика», 2019. – 592 с.