Титульный лист

Задание на курсовую работу

**ОГЛАВЛЕНИЕ**

[**1.** **Введение** 4](#_Toc167019838)

[**2.** **Основная часть** 5](#_Toc167019839)

[2.1. Исследовательская часть 5](#_Toc167019840)

[2.2. Проектная часть 5](#_Toc167019841)

[2.3. Технологическая часть 5](#_Toc167019842)

[**3.** **Заключение** 6](#_Toc167019843)

[**4.** **Список используемой литературы** 7](#_Toc167019844)

# **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.

*Описываются объект и предмет исследования, используемые методы и информационная база исследования, а также кратко характеризуется структура КР по разделам.*

# **Основная часть**

## Исследовательская часть

*Анализ существующих аналогов с указание их достоинств и недостатков*

**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.

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 1 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:

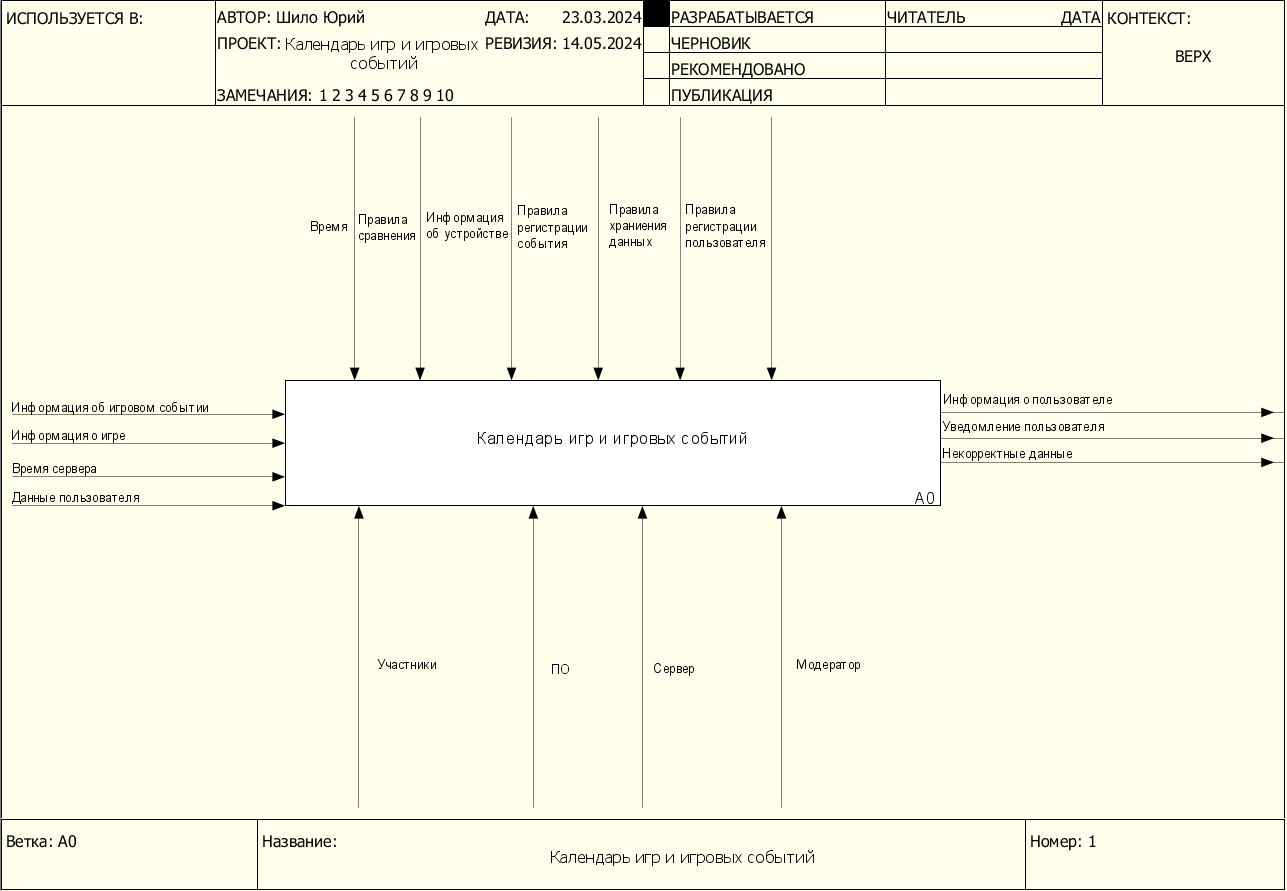
1. Участники: активные пользователи системы, которые ведут свои списки и взаимодействуют с календарем;
2. Сервер: обеспечивает обработку и хранение данных;
3. Модератор: отвечает за актуализацию данных.

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.

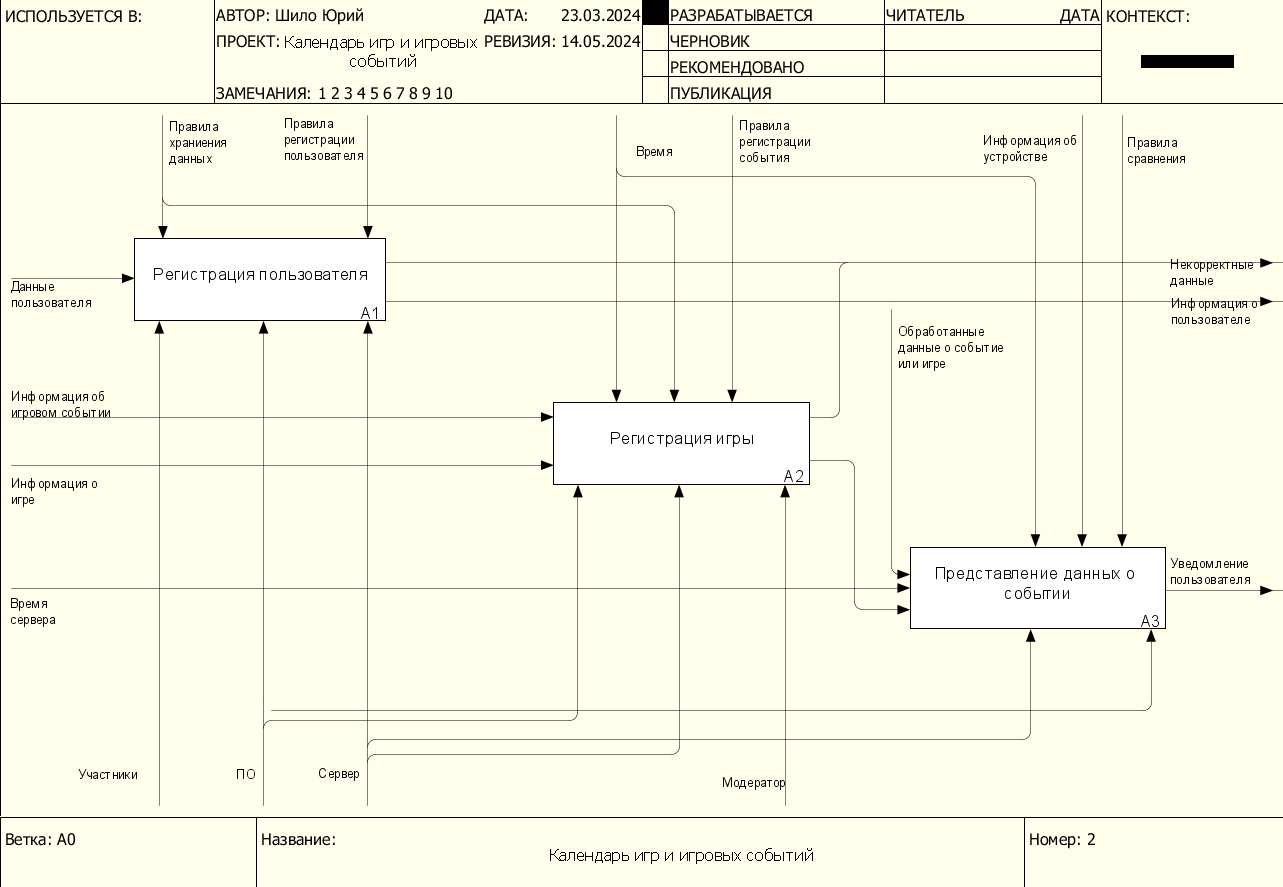
Выходы:

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 1. – Нулевой уровень построенной IDEF0 диаграммы.*

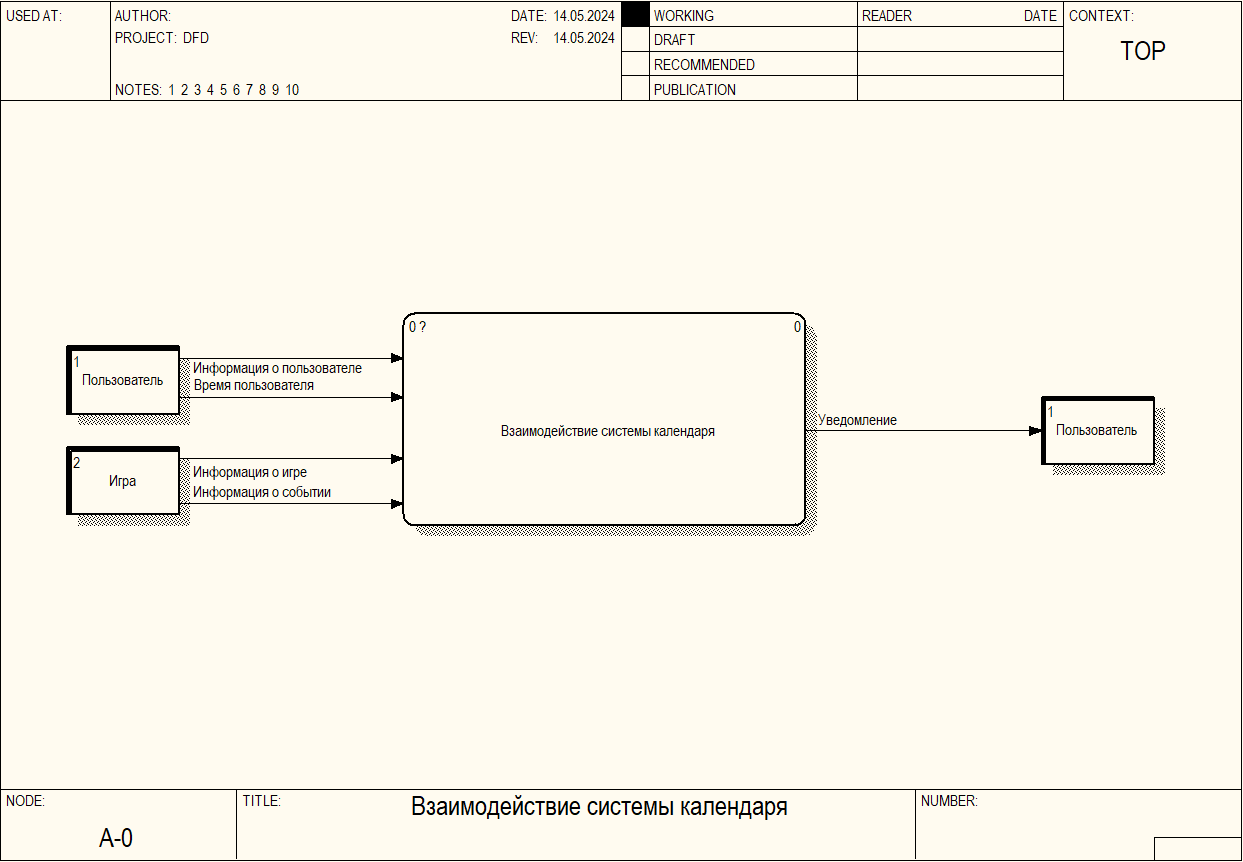
Для более глубокого понимания архитектуры нашей системы рассмотрим её на более низком уровне. На этом уровне будет демонстрироваться процесс регистрации пользователя и игры, а также механизм отправки уведомлений. Взаимодействие каждого блока предоставлена на рисунке 2.



*Рисунок 2. – Архитектура нашей системы на нижнем уровне.*

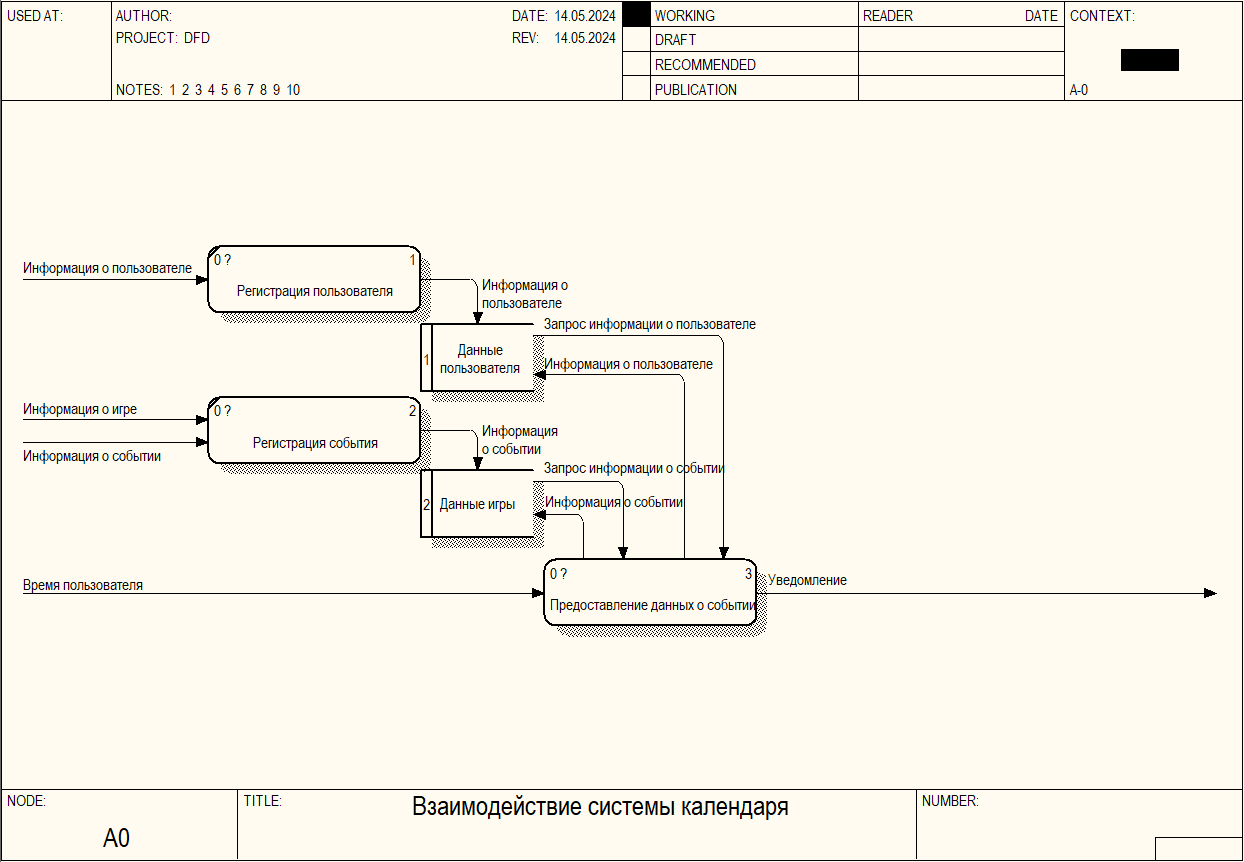
Для того чтобы иметь более полное представление о реализуемой системы, мы применяем DFD (Data Flow Diagram) нотацию для нашего приложения. Это позволяет нам визуализировать и анализировать все потоки данных, проходящие через различные компоненты системы.

На рисунке 3 представлен самый первый слой реализуемой системы.



*Рисунок 3.* – *DFD нотация для нашей системы на верхнем уровне.*

На рисунке 4, 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 4. – DFD нотация для нашей системы на нижнем уровне.*

## Проектная часть

*Структурная и функциональная схемы*

**Описание и обоснование выбора ПО**

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.

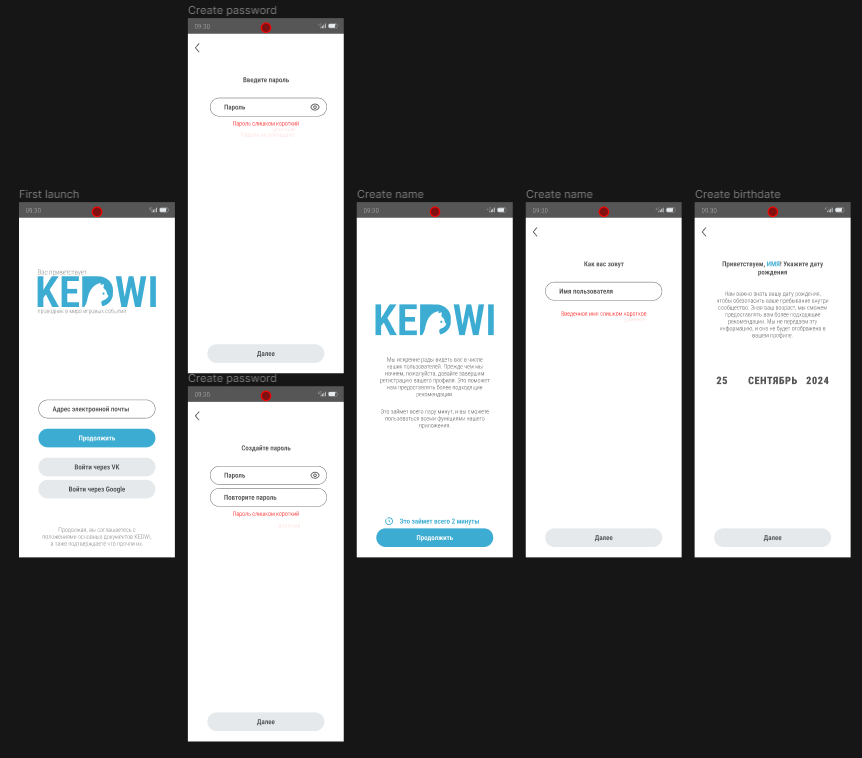
На рисунке 3, представлен итоговый вариант иконки.



*Рисунок 5. – 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 4 представлены некоторые из окон нашего приложения, дизайн которых был спроектирован в Figma.

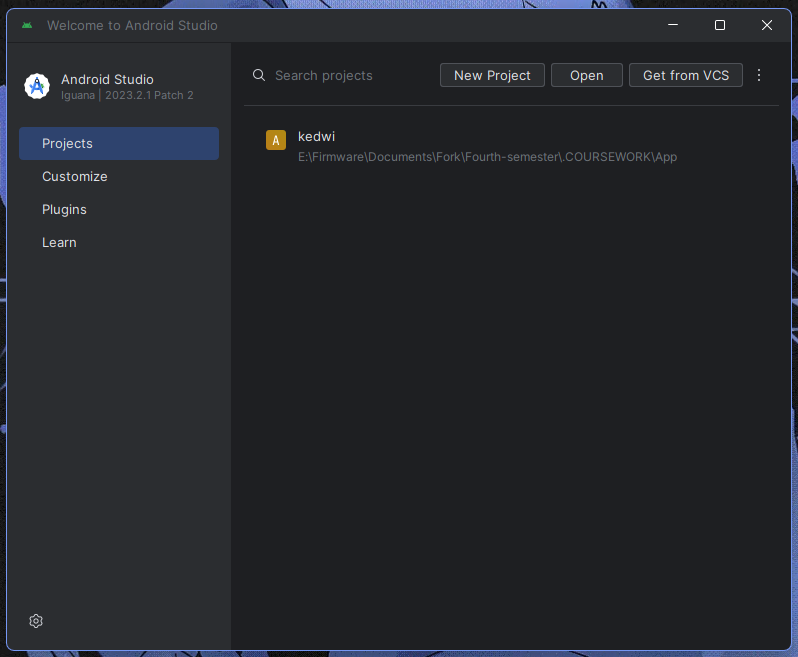


*Рисунок 6. – Дизайн окон регистрации, построенный в Figma.*

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 5 показан приветственный экран Android Studio.



*Рисунок 7. – Окно, запускаемое при входе в Android Studio.*

*архитектура программной системы (проектирование пользовательского интерфейса, проектирование базы данных)*

**Сетевая база данных**

## Технологическая часть

*подробное описание разрабатываемой системы (инструкция пользователя)*

*тестирование программного продукта (по желанию – метод черного ящика)*

# **Заключение**

# **Список используемой литературы**

<https://stackoverflow.com/>

<https://android-tools.ru/>

<https://developer.android.com/>

<https://bumptech.github.io/glide/>

<https://www.geeksforgeeks.org/>

<https://www.youtube.com/>

<https://firebase.google.com/docs?hl=en>