



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных  
технологий

## **КУРСОВАЯ РАБОТА**

по дисциплине «Разработка мобильных приложений»

### **Тема курсовой работы**

«Мобильное приложение для отслеживания финансов»

**Студент группы ИКБО-13-22**

Тринеев Павел Сергеевич

(подпись студента)

**Руководитель курсовой работы**

доцент Рысин М.Л.

(подпись руководителя)

Работа представлена к защите

«25» мая 2024 г.

Допущен к защите

«25» мая 2024 г.

Москва – 2024



МИНОБРНАУКИ РОССИИ


Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

Утверждаю  
Заведующий кафедрой МОСИТ  
 Головин С.А.

«19» февраля 2024 г.

## ЗАДАНИЕ

### на выполнение курсовой работы по дисциплине

«Разработка мобильных приложений»

Студент Тринеев Павел Сергеевич

Группа ИКБО-13-22

**Тема работы:** «Мобильное приложение для отслеживания финансов»

**Исходные данные:**

**Разрабатываемый прототип** должен предоставлять возможности полностью интерактивной системы с понятным дружественным UI и обеспечивать необходимую функциональность, которая формируется в зависимости от заданной темы и предметной области изучаемых вопросов.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

Установка и настройка Android с применением виртуальных сред. Установка и настройка IDE

(Android Studio, Eclipse, IntelliJ IDEA, Kivi и т.п.). Установка и настройка эмуляторов Android.

Анализ предметной области разрабатываемой программной системы, сбор и анализ требований, составление ТЗ согласно ГОСТ 19.201-78.

Изучение жизненного цикла программ и создание мобильного программного комплекса с применением языка программирования JAVA, согласно теме курсовой работы. Реализация в создаваемом программном комплексе хранения данных в виде БД, файлов или пар ключ-значение.

Обеспечение их создания, чтения, записи во внутреннем и внешнем хранилище.

Обеспечение безопасности информации при работе планового программного комплекса (обеспечение работы ролевой модели безопасности).

Тестирование и отладка кода созданного программного продукта, контрольные примеры работы.

Возможно портирование написанной программной системы на внешних хостах в сети Интернет.

Отчет по курсовой работе в виде пояснительной записки.

**Срок представления к защите курсовой работы:** «25» мая 2024 г.

Задание на курсовую работу выдал



Рысин М.Л.

«19» февраля 2024 г.

Задание на курсовую работу получил



(Тринеев П.С.)

«19» февраля 2024 г.

## Содержание

ВВЕДЕНИЕ .....	4
АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	5
ОБЗОР СУЩЕСТВУЮЩИХ АНАЛОГОВ .....	8
ТЕХНИЧЕСКОЕ ЗАДАНИЕ .....	10
ВВЕДЕНИЕ .....	10
1. Основание для разработки .....	10
1.1. Документы, на основании которых ведется разработка:.....	10
1.2. Организация, утвердившая данный документ, и дата его утверждения: .....	10
2. Назначение разработки .....	10
3. Требования к программе или программному изделию .....	11
3.1. Требования к функциональным характеристикам .....	11
3.2. Требования к надежности.....	12
3.3. Требования к составу и параметрам технических средств .....	12
4. Требования к программной документации.....	12
5. Этапы разработки (таб. 1).....	13
АРХИТЕКТУРА ПРОГРАММНОЙ СИСТЕМЫ.....	14
ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ И WIREFRAME ПРИЛОЖЕНИЯ.....	17
КОНТРОЛЬНЫЙ ПРИМЕР .....	20
ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЮ .....	22
ЗАКЛЮЧЕНИЕ .....	23
СПИСОК ЛИТЕРАТУРЫ .....	24
ПРИЛОЖЕНИЕ .....	26

## **ВВЕДЕНИЕ**

В современном мире учет своих финансов становится важной задачей, особенно если нужно срочно разобраться с расходами и доходами. Мобильное приложение для подсчета финансов представляет собой решение, направленное на облегчение процесса управления деньгами для пользователей. Это приложение позволит пользователям отслеживать свои траты, планировать бюджет и видеть полную картину своих финансов в любое время.

Цель курсовой работы: разработать мобильное приложение для подсчета своих финансов средствами языка Java, чтобы упростить процесс учета расходов, а также планирования личного бюджета.

Задачи курсовой работы:

- 1) проанализировать предметную область мобильного приложения;
- 2) сделать обзор существующих аналогов разрабатываемого приложения;
- 3) сформировать 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.

## АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Приложение предназначено для подсчета своих финансов. В приложении собраны данные о расходах. Пользователь может добавлять траты вручную. Также есть страница с руководством пользования, а также информацией о разработчике.

Это программа для отслеживания своих финансов, которая поможет вам увидеть полную картину ваших доходов и расходов. С её помощью вы сможете лучше управлять своими средствами и избегать ненужных долгов. Используя эту программу, вы сможете планировать свои финансовые цели и достигать их с уверенностью.

Информационная модель данных представлена ER-диаграммой. диаграмма представляет собой графическое отображение структуры данных, включающее сущности (объекты), их атрибуты и связи между ними (рис. 1).

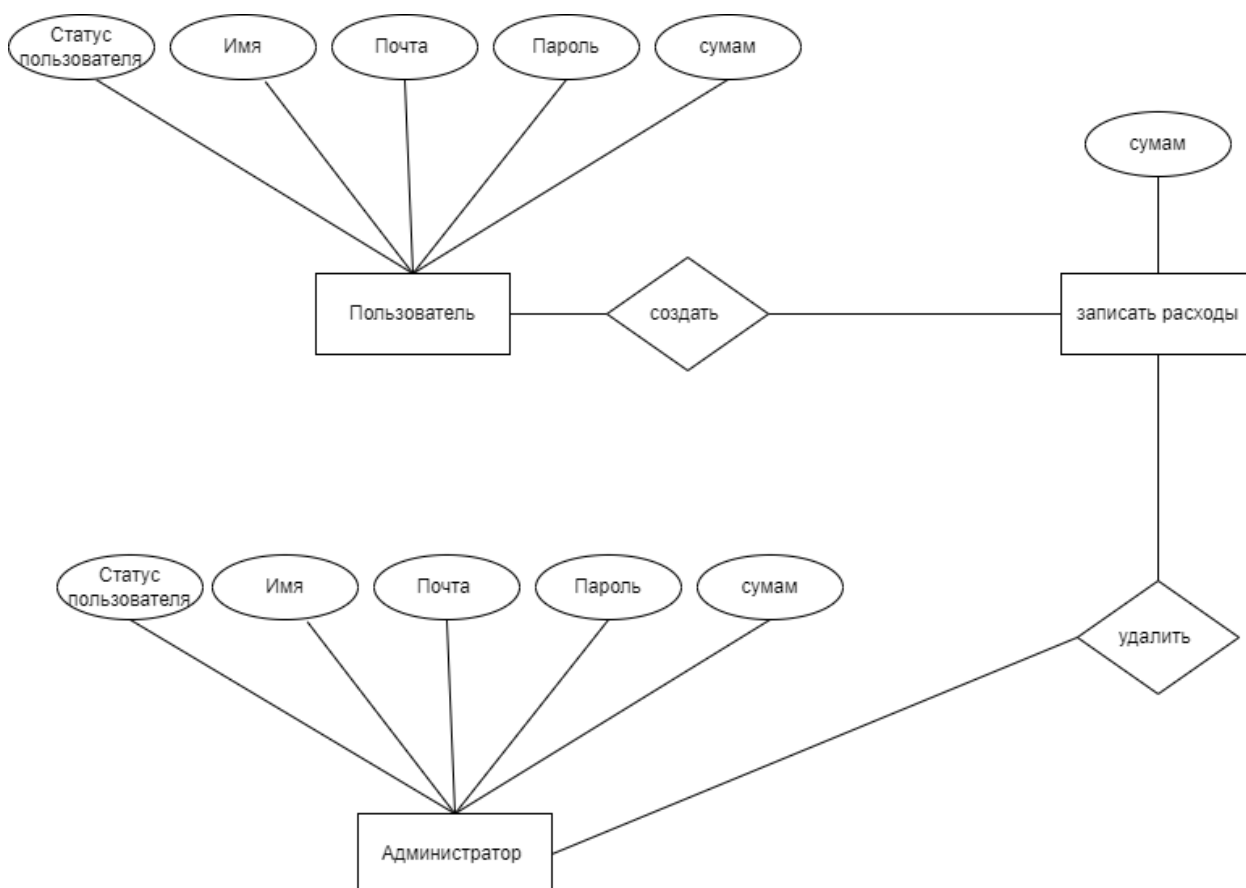


Рисунок 1. – ER -диаграмм.

Также в курсовой работе бизнес-процессы, в которых будет задействована система объявлена с помощью диаграммы IDEF0. Детальное представление этих процессов можно увидеть на представленной диаграмме IDEF0 (рис. 2-3).

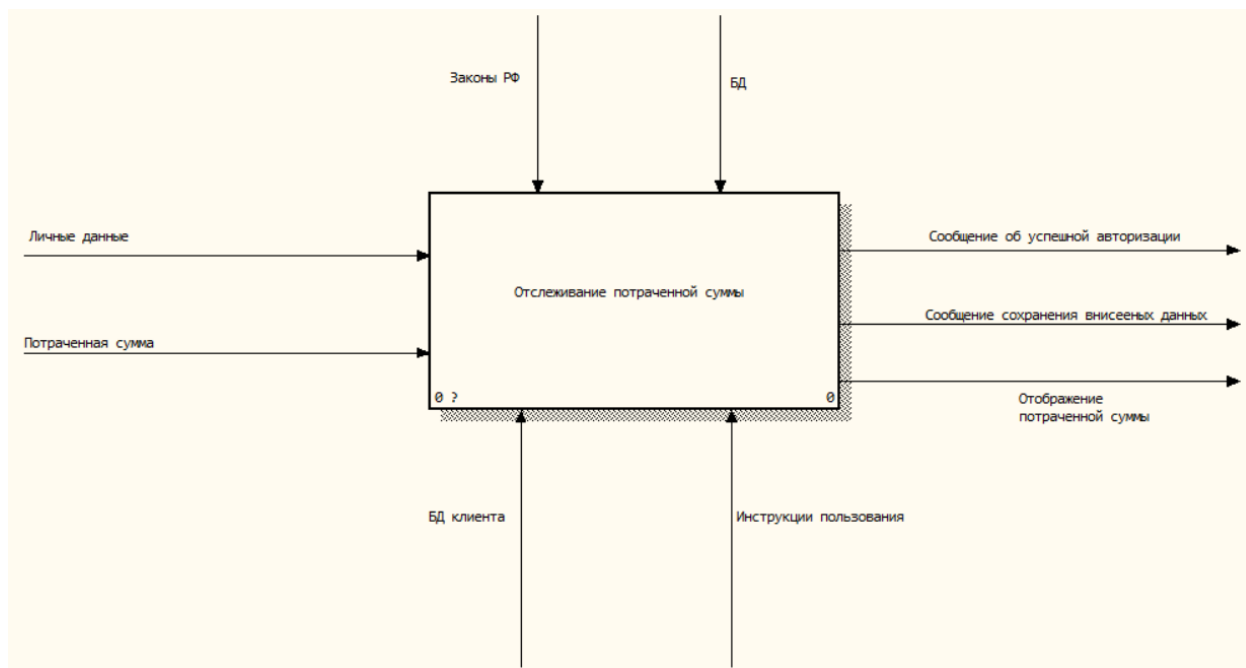


Рисунок 2. – Диаграмма IDEF0.

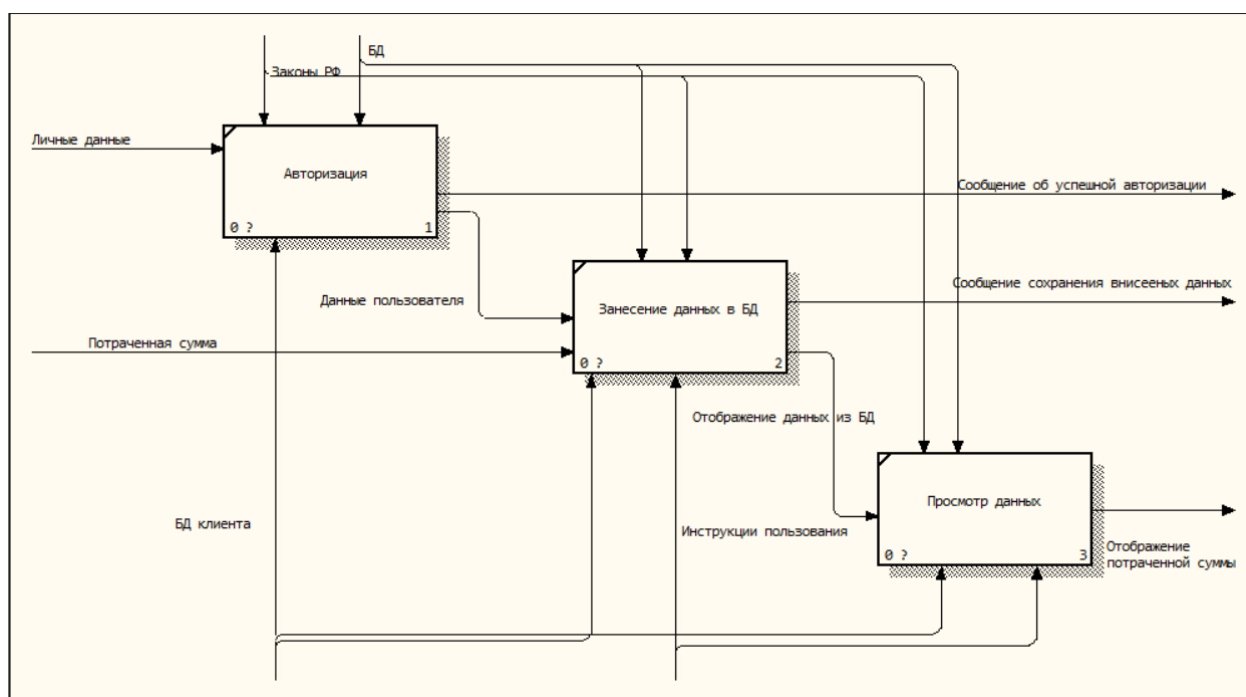


Рисунок 3. - Диаграмма IDEF0, декомпозиция процесса.

Вывод: Разработка мобильного приложения для отслеживания своих финансов является необходимой и актуальной задачей, так как такие приложения способствуют улучшению финансовой грамотности пользователей и упрощают процесс управления личным бюджетом. Эти приложения обеспечивают пользователям удобство и доступность при контроле расходов и доходов, что повышает уровень финансовой осведомленности и стабильности. Кроме того, такие приложения способствуют экономии времени и усилий пользователей, делая процесс управления финансами более эффективным и организованным.

## ОБЗОР СУЩЕСТВУЮЩИХ АНАЛОГОВ

На рынке существует несколько мобильных приложений, предназначенных для отслеживания своих трат. Рассмотрим несколько популярных аналогов: Money Manager, CoinKeeper, Wallet, Monefy.

### 1) Money Manager

Достоинства:

- Удобный интерфейс;
- Возможность синхронизации с банковскими счетами;
- Подробные отчеты и графики.

Недостатки:

- Пользователи жалуются на медленную работу приложения при большом объеме данных.

### 2) CoinKeeper

Достоинства:

- Простота использования;
- Возможность планирования бюджета;
- Уведомления о превышении лимитов.

Недостатки:

- Некоторые функции доступны только в платной версии.
- функции доступны только в платной версии.

### 3) Wallet

Достоинства:

- Широкий функционал;
- Возможность работы с разными валютами;
- Синхронизация с банками и платежными системами.

Недостатки:

- Некоторые пользователи отмечают сложность начальной настройки.

### 4) Wallet

Достоинства:



- Широкий функционал;
- Возможность работы с разными валютами;
- Синхронизация с банками и платежными системами.

Недостатки:

- Некоторые пользователи отмечают сложность начальной настройки.

Вывод: Существующие мобильные приложения для отслеживания своих трат обладают некоторыми преимуществами, такими как удобный интерфейс, возможность синхронизации с банковскими счетами, планирование бюджета и визуализация данных. Однако, все они сталкиваются с определенными недостатками, такими как медленная работа при большом объеме данных, ограничения в бесплатных версиях и сложность начальной настройки. Учитывая обнаруженные недостатки, разработка нового мобильного приложения для отслеживания своих трат представляется обоснованной.

# **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

## **ВВЕДЕНИЕ**

Настоящее техническое задание распространяется на разработку системы отслеживания своих трат. Широкий круг задач по управлению бюджетом и контролю расходов, что способствует достижению финансовой стабильности и планированию бюджета. В настоящее время для решения таких задач широко используются разнообразные приложения и алгоритмы, требующие различных вычислительных ресурсов и предлагающие неодинаковую функциональность. При этом данные методы и приложения не всегда систематизированы, оценки их эффективности и удобства использования не всегда полны и разбросаны по многим источникам. В рамках единой системы не существует программной реализации, которая бы охватывала все основные аспекты управления личными финансами.

### **1. Основание для разработки**

#### **1.1. Документы, на основании которых ведется разработка:**

Задание на выполнение курсовой работы по дисциплине  
“Разработка мобильных приложений”

#### **1.2. Организация, утвердившая данный документ, и дата его утверждения:**

Федеральное государственное бюджетное образовательное  
учреждение высшего образования «МИРЭА – Российский  
технологический университет» РТУ МИРЭА - «19» февраля 2024 г.

### **2. Назначение разработки**

Система призвана решить небольшой круг комбинаторно-оптимизационных задач, в частности задач разработчика:

отслеживание своих финансовых транзакций;  
планировании бюджета и достижении финансовых целей.

Пользователями могут выступать все, кто заинтересован в управлении личными финансами, включая студентов, специалистов различных профессий, а также людей, стремящихся к улучшению своей финансовой грамотности. Но

данное приложение было разработано под решения задач узкого круга пользователей.

### **3. Требования к программе или программному изделию**

#### **3.1. Требования к функциональным характеристикам**

Система должна представлять совокупность методических и программных средств решения следующих задач:

- Программа должна быть мобильным приложением с двумя или более Activity, с использованием фрагментов, всплывающих сообщений и уведомлений;
- Интерфейс программы – на русском языке;
- В составе view-элементов реализовать меню, кнопки, поля ввода, графическое изображение, адаптивную иконку (иные типы элементов – по необходимости);
- В программе реализовать список, каждый пункт должен содержать текст и изображение;
- В меню программы должен быть пункт «О программе», по которому должно открываться описание программы, а также пункт «Инструкция пользователю»;
- В меню программы должен быть пункт «Об авторе», по которому открываются данные о разработчике (фото или аватарка, фамилия и инициалы студента, группа, год разработки приложения, почта для связи);
- Программа должна обеспечить ролевую модель безопасности (вход для зарегистрированных пользователей с паролем, различные уровни доступа для разных категорий пользователей);
- Программа должна хранить данные в файле, а также в локальной или сетевой базе данных;
- В программе должны присутствовать элементы анимации;
- в процессе разработки должна быть использована система контроля

версий Github.

### **3.2. Требования к надежности**

Программный продукт должен соответствовать современному уровню требований к надежности программного обеспечения:

- предусматривать контроль вводимой информации и блокировку некорректных действий пользователя при работе с системой;
- обеспечивать целостность информации, хранящейся в базе данных.

### **3.3. Требования к составу и параметрам технических средств**

Системные требования для работы программного продукта должны быть следующими: приложение должно быть доступно для использования на мобильных устройствах с операционной системой Android. Минимальные требования к устройствам для корректной работы приложения:

Процессор: Для Android - Snapdragon 660 или эквивалентный;

Оперативная память: Минимум 2 ГБ;

Свободное место на диске: Не менее 100 МБ;

Разрешение экрана: Минимум 720x1280 пикселей;

Версия операционной системы: Android 7.0 (Nougat) и выше.

Устройства, удовлетворяющие этим требованиям, обеспечат плавное и стабильное функционирование приложения.

## **4. Требования к программной документации**

Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа. Пояснительная записка должна быть загружена в СДО.

Техническое задание и пояснительная записка, титульные листы других документов должны быть напечатаны, подписаны академическим руководителем образовательной программы 09.03.04 «Программная инженерия», руководителем разработки и исполнителем перед сдачей курсовой работы.

Документация и программа также сдается в электронном виде в формате .pdf или .docx. в архиве формата .zip или .rar;

Состав материалов курсового проекта:

- Техническая документация,
- Программный проект,
- Исполняемый файл,
- Отзыв руководителя

## 5. Этапы разработки (таб. 1)

После утверждения технического задания организация-разработчик непосредственно приступает к созданию программного обеспечения.

### *Задание*

Разработать операционную систему на базе Android, на тему: “Приложения для подсчёта своих трат”, а также сформировать пояснительную записку, оформленную в соответствии с ГОСТ 19.106—78.

*Таблица 1. – Этапы разработки.*

Номер этапа	Название этапа	Срок	Отчетность
1	Выбор темы курсовой работы	09.02.2024 – 27.02.2024	Определение темы с учетом предпочтений и навыков
2	Разработка примерного представления проекта	27.02.2024 – 07.05.2024	Описание внутренних форматов, интерфейса и форматов данных базы. Реализация системы на уровне интерфейса
3	Первая контрольная точка	07.05.2024	Показ выполненной разработки
4	Вторая контрольная точка	21.05.2024	Показ объяснительной записки
5	Сдача курсовой работы	11.06.2024	Защита объяснительной записки и проекта

## АРХИТЕКТУРА ПРОГРАММНОЙ СИСТЕМЫ

Общее описание архитектуры программной системы может включать следующие основные аспекты:

1. Клиент-серверная архитектура: В случае мобильного приложения, клиентская часть взаимодействует с сервером для получения данных, обновлений и других ресурсов. Это позволяет обеспечить масштабируемость, удобство обновлений и централизованное управление данными.
2. Пользовательский интерфейс: Архитектура включает в себя компоненты, отвечающие за отображение пользовательского интерфейса, обработку пользовательских действий и взаимодействие с бэкендом.
3. Бэкенд: это часть системы, отвечающая за обработку бизнес-логики, работу с базой данных, аутентификацию и авторизацию пользователей, обработку запросов от клиентской части приложения.
4. База данных: Система использует NoSQL базу данных Firebase для хранения и управления данными, в зависимости от требований и особенностей приложения.
5. Безопасность: Архитектура включает в себя механизмы защиты данных, аутентификации и авторизации пользователей, обработку ошибок и предотвращение уязвимостей.

Процесс проектирования программного обеспечения включает в себя определение структурных компонентов программной системы и связей между ними. Результат уточнения структуры может быть представлен в виде структурной схемы (рис. 4).



Рисунок 4. - Структурная схема.

Функциональные схемы более информативны, чем структурные. На рис. 5 приведена функциональная схема программной системы.

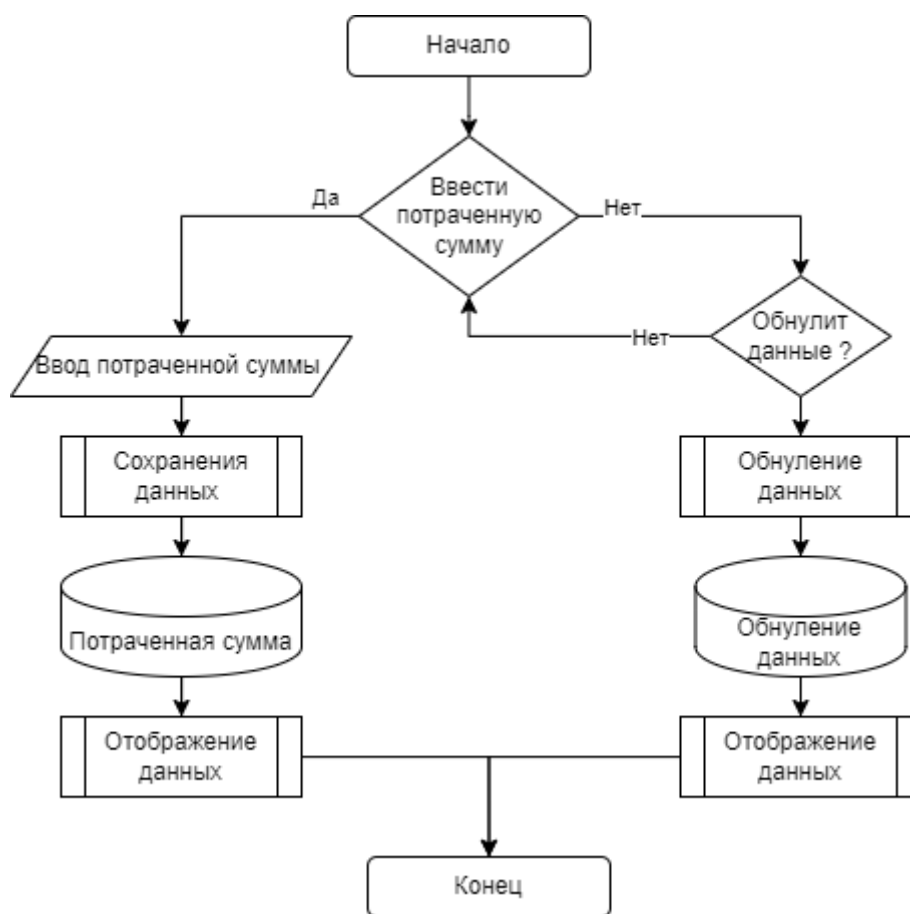


Рисунок 5. - Функциональные схемы.

Процесс проектирования базы данных включает в себя определение структурных элементов базы данных и их взаимосвязей. Результатом этого процесса может быть создание схемы базы данных, которая является абстрактным представлением структуры данных и их взаимосвязей в базе данных. Схема базы данных предоставляет полное представление о структуре данных и их взаимосвязях в базе данных, что позволяет проектировщикам и разработчикам лучше понять и организовать хранение и доступ к данным. На рисунках 6-7 приведена схема базы данных, которая включает в себя таблицы для хранения информации о потраченной сумме и пользователях. Схема представлена в нотации IDEF1X.

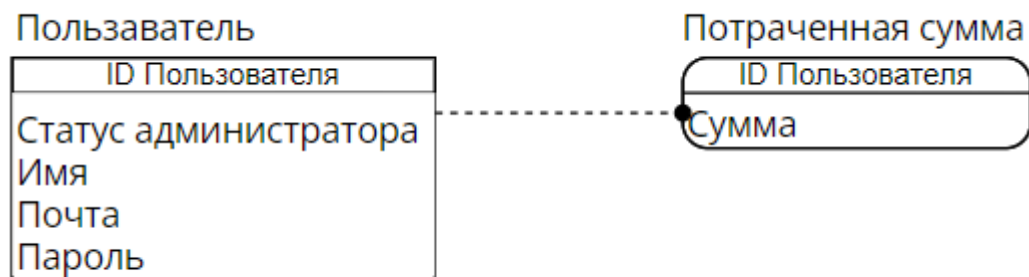


Рисунок 6. - Логический уровень.

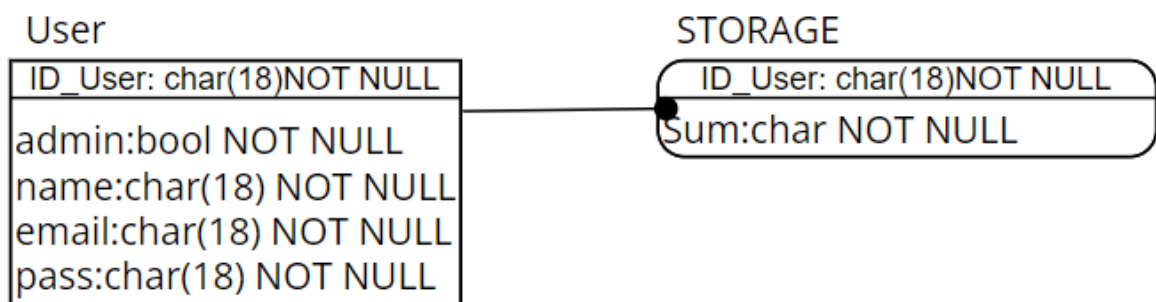


Рисунок 7. - Физический уровень.



## ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ И WIREFRAME ПРИЛОЖЕНИЯ

Пользовательский сценарий отображен в виде Use case диаграммы (рис 8).



Рисунок 8. - Пользовательский сценарий.

Вайрфрейм (Wireframe) Вайрфрейм (англ. wireframe) представляет собой простые, схематичные чертежи или макеты, используемые для проектирования веб-сайтов, мобильных приложений и других интерфейсов. Они состоят из низкоуровневых графических изображений, обычно черно-белых или серых, показывающих расположение элементов на экране и основные функциональные блоки интерфейса. Вайрфреймы позволяют дизайнерам и разработчикам быстро прототипировать концепцию интерфейса, определить его структуру, навигацию и взаимодействие элементов. Они служат для предварительного тестирования и обсуждения с заказчиком или командой разработки, помогая выявить основные проблемы и потребности пользователей на ранних стадиях проекта. Это улучшает понимание концепции и сокращает время и затраты на разработку. Пример вайрфрейма приложения показан на рис. 9.

### Вход

Имя пользователя

Пароль

Вход

Регистрация

### Регистрация

Почта

Имя пользователя

Пароль

Регистрация

### Сумма:0

Введите число

ДОБАВИТЬ

ОЧИСТИТЬ

*а) Вход*

*б) Регистрация*

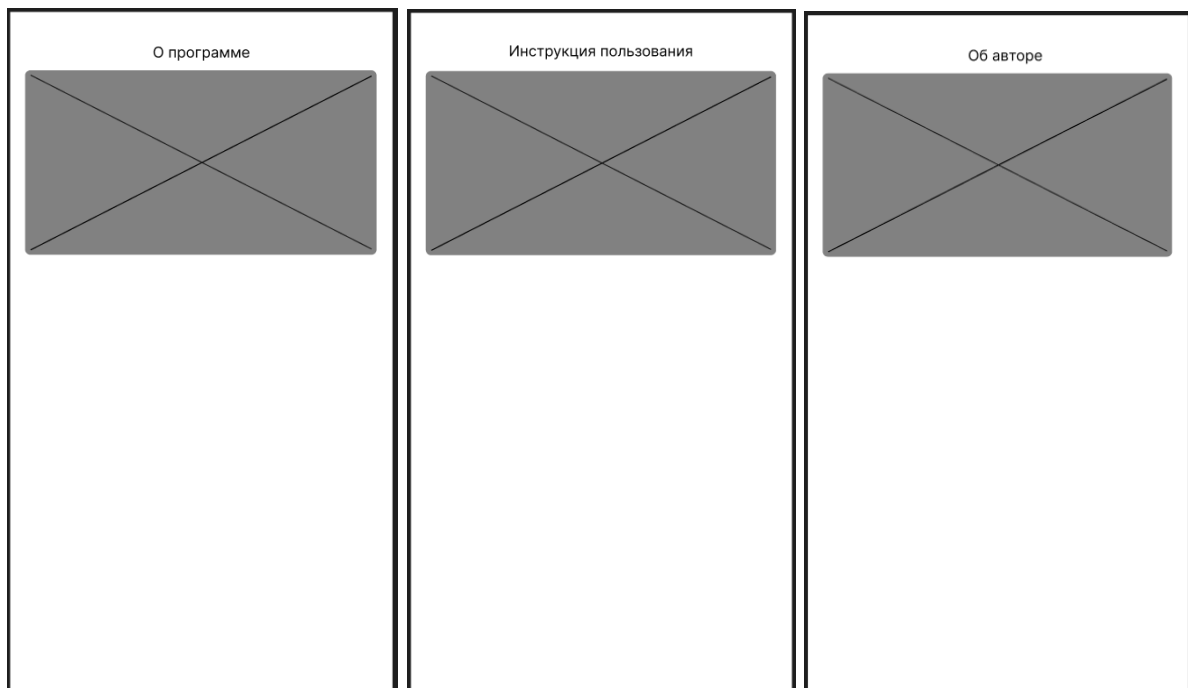
*в) Главная*

☐
О программе

☐
Инструкция пользования

☐
Об авторе

*г) Список справок и информации*



*д) О программе*

*е) Инструкция  
пользования*

*ё) Об авторе*

*Рисунок 9. - Wireframe приложения.*

## КОНТРОЛЬНЫЙ ПРИМЕР

Далее приведены все варианты использования рисунок 10.

Вход	Регистрация	Сумма:0
<div>Имя пользователя</div> <div>Пароль</div> <div>Войти</div> <div>Регистрация</div>	<div>почта</div> <div>Имя пользователя</div> <div>Пароль</div> <div>Регистрация</div>	<div>Введите число</div> <div>ДОБАВИТЬ</div> <div>ОЧИСТИТЬ</div> <div>&gt;</div>

*а) Вход*

*б) Регистрация*

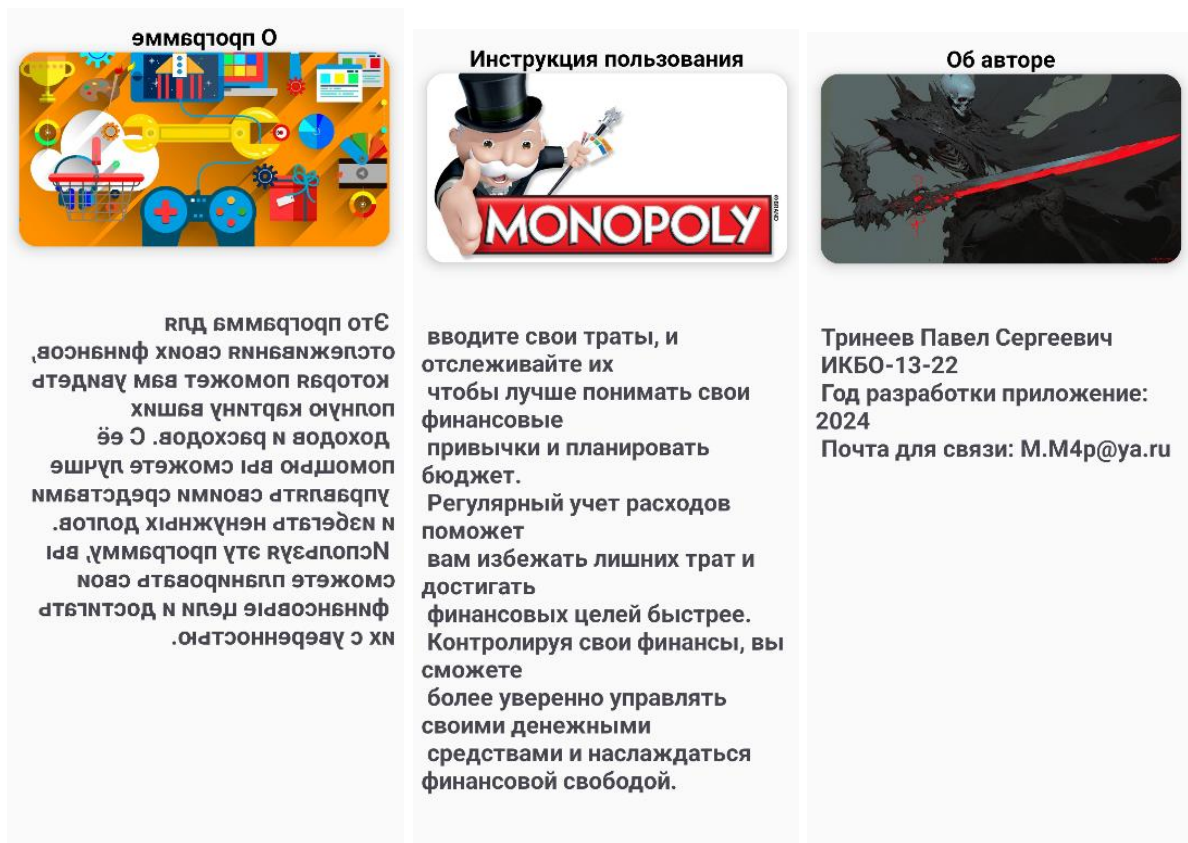
*в) Главная*

О программе

Инструкция пользования

Об авторе

*г) Список справок и информации*



д) О программе

е) Инструкция  
пользования

ё) Об авторе

Рисунок 9. - Wireframe приложения.

## **ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЮ**

В приложении есть 2 пункта меню.

1. Главная страница: здесь содержится основная информация о потраченной сумме. Также возможность добавлять и очищать информацию о сумме.
2. Информационный список: здесь содержится список всех сведений, таких как: <<О программе>>, <<Инструкция пользования>>, <<Об авторе>>.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы успешно были решены все поставленные задачи, что подтверждает достижение цели разработки мобильного приложения для отслеживания своих трат.

Первоначально была проведена глубокая аналитика предметной области, что позволило более полно понять специфику и особенности отслеживания личных финансов. Затем был осуществлен обзор существующих аналогов, что помогло выделить уникальные особенности и преимущества разрабатываемого приложения.

Сформулированное техническое задание на разработку программы в соответствии с ГОСТ 19.201-78 стало основой для четкого определения требований и ожиданий от продукта. Далее была разработана архитектура программной системы, включая структурные и функциональные диаграммы, а также схему базы данных, что обеспечило эффективное проектирование и последующую реализацию приложения.

Проектирование интерфейса мобильного приложения было выполнено с акцентом на удобство и простоту использования для конечных пользователей. Затем был написан и отлажен код программы на языке высокого уровня Java, а также проведено тестирование, что обеспечило стабильную и надежную работу приложения.

Таким образом, выполнение всех поставленных задач позволило достичь цели курсовой работы и создать мобильное приложение, предоставляющее пользователям функционал для эффективного отслеживания своих трат.

## СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.
2. Рудаков А.В. Технология разработки программных продуктов. Практикум: учеб. пособие для студ. - 4 -е изд. – М.: Издательский центр «Академия», 2014. – 192 с.
3. Цуканова О. А. Методология и инструментарий моделирования бизнес - процессов: учебное пособие – СПб.: Университет ИТМО, 2015 – 100 с.
4. Колисниченко Д.Н. Программирование для Android. Самоучитель. - 3-е изд. – СПб.: БХВ-Петербург, 2021. – 288 с
5. Эванс Б., Флэнаган Д. Java. Справочник разработчика. - 7-е изд. – СПб.: ООО «Диалектика», 2019. – 592 с.
6. Клифтон Я. Проектирование пользовательского интерфейса в Android. 2-е изд. – М.: ДМК-Пресс, 2017. – 452 с.
7. К 88 Кумагина, Е.А., Неймарк, Е.А. Модели жизненного цикла и технологии проектирования программного обеспечения: учебно-методическое пособие / Е.А. Кумагина, Е.А. Неймарк. – Нижний Новгород: Изд-во ННГУ, 2016 – 41 с.
8. Sparx Systems User Interaction & Experience Изд-во Enterprise Architect 2022 – 43 стр.
9. Github: Where the world builds software [Электронный ресурс]. URL: <https://github.com/> (дата обращения 18.05.2024).
10. Android Developers [Электронный ресурс]. URL: <https://developer.android.com/> (дата обращения 16.05.2024).
11. The Destination for Java Developers [Электронный ресурс]. URL: <https://dev.java/> (дата обращения 16.05.2024).
12. Figma [Электронный ресурс]. URL: <https://www.figma.com/design/GB0vonHRtsRQYUMyUzZpJe/Untitled?no>



[de-id=0-1&t=M03pITpYACH8h4Mm-1](#) (дата обращения 04.06.2024).

13. Stack overflow [Электронный ресурс]. URL: <https://stackoverflow.com/>  
(дата обращения 04.06.2024).

## ПРИЛОЖЕНИЕ

Проект доступен по ссылке: [https://github.com/VVV-VVV1/Financial\\_Manager.git](https://github.com/VVV-VVV1/Financial_Manager.git)

*Листинг 1. – activity\_main.xml.*

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:background="@color/background_material_light"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/signin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Вход"
        app:layout_constraintVertical_bias="0.057"
        android:textStyle="bold"
        android:textSize="35dp"
        android:gravity="center"
        android:layout_margin="50dp"/>

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/email"
        android:layout_below="@+id/signin"
        android:hint="Имя пользователя"
        android:layout_margin="10dp"
        android:padding="20dp"
        android:background="#30808080"
        android:drawableLeft="@drawable/baseline_attribution_24"
        android:drawablePadding="20dp"
        />

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/password"
        android:layout_below="@+id/email"
        android:hint="Пароль"
        android:layout_margin="10dp"
        android:padding="20dp"
        android:background="#30808080"
        android:drawableLeft="@drawable/twotone_apps_24"
        android:drawablePadding="20dp"
        />

    <com.google.android.material.button.MaterialButton
        android:layout_width="135dp"
        android:layout_height="wrap_content"
        android:id="@+id/loginbtn"
        android:layout_below="@+id/password"
        android:text="Войти"
        android:backgroundTint="@color/cardview_dark_background">
```

```

        android:layout_centerHorizontal="true"
        android:layout_margin="20dp"
    />

    <com.google.android.material.button.MaterialButton
        android:id="@+id/regbtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/loginbtn"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="1dp"
        android:layout_marginTop="1dp"
        android:layout_marginEnd="1dp"
        android:layout_marginBottom="1dp"
        android:backgroundTint="@color/cardview_dark_background"
        android:text="Регистрация" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:text=""
        android:layout_below="@+id/regbtn"
        android:layout_centerHorizontal="true"
    />

</RelativeLayout>

```

*Листинг 2. – MainActivity.java.*

```

package com.example.myapplication;

import android.content.Intent;
import android.content.SharedPreferences;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.example.myapplication.databinding.ActivityMainBinding;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.snackbar.Snackbar;
import com.google.android.material.textfield.TextInputEditText;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class MainActivity extends AppCompatActivity {
    MediaPlayer zvuk_in_login;
    private static final String TAG = "MyApp";
    ActivityMainBinding binding;

```

```

private FirebaseAuth auth;
private TextInputEditText emailField;
private TextInputEditText passwordField;
DatabaseReference usersRef =
FirebaseDatabase.getInstance().getReference().child("Users");

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivityMainBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    auth = FirebaseAuth.getInstance();
    emailField = (TextInputEditText) binding.email;
    passwordField = (TextInputEditText) binding.password;

    Log.d(TAG, "Activity created");
    binding.loginbtn.setOnClickListener(v -> onLoginClick());

    binding.regbtn.setOnClickListener(v -> {
        Intent intent = new Intent (MainActivity.this,
ThirdActivity.class);
        startActivity(intent);
    });

    private void onLoginClick() {
        String email = emailField.getText().toString();
        String pass = passwordField.getText().toString();

        if (email.isEmpty() || pass.isEmpty()) {
            Toast.makeText(this, "Почта/Пароль не могут быть пустыми",
Toast.LENGTH_LONG).show();
            return;
        }

        auth.signInWithEmailAndPassword(email, pass)
            .addOnSuccessListener(new OnSuccessListener<AuthResult>() {
                @Override
                public void onSuccess(AuthResult authResult) {
                    FirebaseUser user =
FirebaseAuth.getInstance().getCurrentUser();

usersRef.child(user.getId()).addListenerForSingleValueEvent(new
ValueEventListener() {
                        @Override
                        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                            if (dataSnapshot.exists()) {
                                boolean isAdmin =
dataSnapshot.child("admin").getValue(Boolean.class);

                                // Сохранение значения isAdmin в
SharedPreferences

                                SharedPreferences sharedPreferences =
getSharedPreferences("MySharedPref", MODE_PRIVATE);
                                SharedPreferences.Editor myEdit =
sharedPreferences.edit();

                                myEdit.putBoolean("admin", isAdmin);
                                myEdit.apply();
                            }
                        }
                    });
                }
            });
    }
}

```

```

    }

    @Override
    public void onCancelled(@NonNull DatabaseError
error) {

        // Обработка ошибок
    }

});

Intent intent = new Intent(MainActivity.this,
SecondActivity.class);
startActivity(intent);
finish();
}

}))

.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure (@NonNull Exception e) {
        Snackbar.make(binding.textView,
            "Authorisation Error: " + e.getMessage(),
            Snackbar.LENGTH_SHORT).show();
    }
});
}
}

```

Листинг 3. – activity\_second.xml.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/background_material_light"
    tools:context=".SecondActivity">

    <TextView
        android:id="@+id/sumDisplay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0"
        android:textSize="40dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:textStyle="bold"

    />

    <EditText
        android:id="@+id/numberInput"
        android:layout_width="match_parent"
        android:layout_height="58dp"
        android:layout_below="@id/sumDisplay"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="37dp"
        android:hint="Введите число"
        android:inputType="numberDecimal" />

```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="100px"
    android:id="@+id/add_button"
    android:layout_below="@+id/numberInput"
    android:text="Добавить"
    android:backgroundTint="@color/cardview_dark_background"
    android:layout_centerHorizontal="true"
    android:layout_margin="20dp"
    android:textColor="@color/LightGrey"
    android:background="@drawable/button_style"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="100px"
    android:id="@+id/remButton"
    android:layout_below="@+id/add_button"
    android:text="Очистить"
    android:backgroundTint="@color/cardview_dark_background"
    android:layout_centerHorizontal="true"
    android:layout_margin="5px"
    android:textColor="@color/LightGrey"
    android:background="@drawable/button_style"/>

<ImageButton
    android:id="@+id/nextActivity"
    android:layout_width="30dp"
    android:layout_height="100dp"
    android:layout_below="@+id/remButton"
    android:layout_alignParentEnd="true"
    android:layout_margin="20dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="14dp"
    android:layout_marginBottom="20dp"
    android:backgroundTint="@color/cardview_dark_background"
    android:src="@drawable/baseline_navigate_next_24"
    android:textColor="@color/LightGrey" />

<!--      <ImageButton-->
<!--      android:id="@+id/AnalisActivity"-->
<!--      android:layout_width="30dp"-->
<!--      android:layout_height="100dp"-->
<!--      android:layout_below="@+id/remButton"-->
<!--      android:layout_margin="20dp"-->
<!--      android:layout_marginStart="20dp"-->
<!--      android:layout_marginTop="30dp"-->
<!--      android:layout_marginEnd="14dp"-->
<!--      android:layout_marginBottom="20dp"-->
<!--      android:backgroundTint="@color/cardview_dark_background"-->
<!--      android:src="@drawable/baseline_navigate_before_24"-->
<!--      android:textColor="@color/LightGrey" />-->

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:text=""
    android:layout_below="@+id/remButton"
    android:layout_centerHorizontal="true"
/>

```

```
</RelativeLayout>
```

#### Листинг 4. – *SecondActivity.java*

```
package com.example.myapplication;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.example.myapplication.databinding.ActivitySecondBinding;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.HashMap;

public class SecondActivity extends AppCompatActivity {
    private static final String TAG = "MyApp";
    ActivitySecondBinding binding;
    private EditText numberInput;
    private Button addButton;
    private TextView sumDisplay;
    private float sum = 0;
    SharedPreferences sPref;
    final String SAVED_TEXT = "saved_text";
    DatabaseReference users;

    FirebaseDatabase db = FirebaseDatabase.getInstance();
    DatabaseReference usersRef =
    FirebaseDatabase.getInstance().getReference().child("Users");

    FirebaseAuth auth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivitySecondBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

        binding.addButton.setOnClickListener(v -> addNumber());

        binding.remButton.setOnClickListener(v -> removeNum());

        binding.nextActivity.setOnClickListener(v ->{
            Intent intent = new Intent (SecondActivity.this,
```

```

FourthActivity.class);
        startActivity(intent);
    });

    auth = FirebaseAuth.getInstance();

    db = FirebaseDatabase.getInstance();

    users = db.getReference("Users");

    //        binding.AnalisActivity.setOnClickListener(v ->{
    //            Intent intent = new Intent(SecondActivity.this,
    FifthActivity.class);
    //            Log.d("ActivityTransition", "Navigating to SecondActivity");
    //            startActivity(intent);
    //        });

    SharedPreferences sharedPreferences =
getSharedPreferences("MySharedPref", MODE_PRIVATE);
    boolean isAdmin = sharedPreferences.getBoolean("admin", false);

    if (isAdmin) {
        binding.remButton.setVisibility(View.VISIBLE);
    } else {
        binding.remButton.setVisibility(View.GONE);
    }

}

private void addNumber() {
    String numberStr = binding.numberInput.getText().toString();
    if (!numberStr.isEmpty()) {
        float number = Float.parseFloat(numberStr);
        sum += number;

        binding.sumDisplay.setText("Сумма " + sum);
        binding.numberInput.setText("");
    }

    HashMap hashMap = new HashMap();
    hashMap.put("sum", sum);

    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

    usersRef.child(user.getId()).updateChildren(hashMap);

    Toast.makeText(SecondActivity.this, "Данные сохранены",
Toast.LENGTH_LONG).show();
}

private void loadNum(){
    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    usersRef.child(user.getId()).addValueEventListener(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            binding.sumDisplay.setText("Сумма:" +
snapshot.child("sum").getValue().toString());
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    })
}

```



```

    });
}

private void removeNum() {
    HashMap hashMap = new HashMap();
    hashMap.put("sum", 0);

    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

    usersRef.child(user.getId()).updateChildren(hashMap);

    Toast.makeText(SecondActivity.this, "Данные удалены",
    Toast.LENGTH_LONG).show();

    sum = 0;
    binding.sumDisplay.setText("");
}

@Override
protected void onStart() {
    super.onStart();
    loadNum();
}
}

```

*Листинг 5. – activity\_third.xml.*

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/background_material_light"
    tools:context=".ThirdActivity">

    <TextView
        android:id="@+id/register"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Регистрация"
        app:layout_constraintVertical_bias="0.057"
        android:textStyle="bold"
        android:textSize="35dp"
        android:gravity="center"
        android:layout_margin="50dp"/>

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/email"
        android:layout_below="@+id/register"
        android:hint="почта"
        android:layout_margin="10dp"
        android:padding="20dp"
        android:background="#30808080"
        android:drawableLeft="@drawable/baseline_alternate_email_24"
        android:drawablePadding="20dp"
        />

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/userName"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/email"
        android:layout_margin="10dp"
        android:background="#30808080"
        android:drawableLeft="@drawable/baseline_attribution_24"
        android:drawablePadding="20dp"
        android:hint="Имя пользователя"
        android:onClick="onClick"
        android:padding="20dp" />

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/password"
            android:layout_below="@+id/username"
            android:hint="Пароль"
            android:layout_margin="10dp"
            android:padding="20dp"
            android:background="#30808080"
            android:drawableLeft="@drawable/twotone_apps_24"
            android:drawablePadding="20dp"
            />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/regbtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/password"
            android:layout_centerHorizontal="true"
            android:layout_marginStart="20dp"
            android:layout_marginTop="20dp"
            android:layout_marginEnd="20dp"
            android:layout_marginBottom="20dp"
            android:backgroundTint="@color/cardview_dark_background"
            android:text="Регистрация"
            app:toggleCheckedStateOnClick="false" />

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"
            android:text=""
            android:layout_below="@+id/regbtn"
            android:layout_centerHorizontal="true"
            />
    </RelativeLayout>

```

*Листинг 6. – ThirdActivity.java.*

```

package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.widget.EditText;

```

```

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.example.myapplication.databinding.ActivityThirdBinding;
import com.example.myapplication.models.User;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.snackbar.Snackbar;
import com.google.android.material.textfield.TextInputEditText;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.UserProfileChangeRequest;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class ThirdActivity extends AppCompatActivity {
    ActivityThirdBinding binding;
    FirebaseAuth auth;
    TextInputEditText emailField;
    TextInputEditText nameField;
    TextInputEditText passwordField;

    FirebaseDatabase db = FirebaseDatabase.getInstance();
    DatabaseReference users;

    DatabaseReference arbitraryDataRef = db.getReference("ArbitraryData");

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityThirdBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        auth = FirebaseAuth.getInstance();

        db = FirebaseDatabase.getInstance();

        users = db.getReference("Users");
        binding.regbtn.setOnClickListener(v -> onRegistrationClick());
    }

    private void onRegistrationClick() {
        final EditText email = binding.email;
        final EditText name = binding.userName;
        final EditText pass = binding.password;

        if (TextUtils.isEmpty(email.getText().toString())) {
            Snackbar.make(binding.textView, "Enter your e-mail address",
Snackbar.LENGTH_SHORT).show();
            return;
        }
        if (TextUtils.isEmpty(name.getText().toString())) {
            Snackbar.make(binding.textView,
"Enter your name",
Snackbar.LENGTH_SHORT).show();
            return;
        }
        if (pass.getText().toString().length() < 5) {
            Snackbar.make(binding.textView,
"Enter the password at least 5 characters",
Snackbar.LENGTH_SHORT).show();

```

```

        return;
    }

    auth.createUserWithEmailAndPassword(email.getText().toString(),
pass.getText().toString())
        .addOnSuccessListener(new OnSuccessListener<AuthResult>() {
            @Override
            public void onSuccess(AuthResult authResult) {

                User user = new User();
                user.setEmail(email.getText().toString());
                user.setName(name.getText().toString());
                user.setPassword(pass.getText().toString());
                user.setAdmin(false);
                user.setSum(0);

                users.child(FirebaseAuth.getInstance().getCurrentUser().getUid())
                    .setValue(user).addOnSuccessListener(new
OnSuccessListener<Void>() {
                        @Override
                        public void onSuccess(Void unused) {
                            // Создание объекта
                            UserProfileChangeRequest и установка имени пользователя
                            UserProfileChangeRequest
profileUpdates = new UserProfileChangeRequest.Builder()
                                .setDisplayName(name.getText().toString())
                                    .build();

                            // Обновление профиля пользователя
                            FirebaseAuth.getInstance().getCurrentUser().updateProfile(profileUpdates)
                                .addOnCompleteListener(new
OnCompleteListener<Void>() {
                                    @Override
                                    public void
onComplete(@NonNull Task<Void> task) {
                                        if
(task.isSuccessful()) {
                                            Snackbar.make(binding.textView, "Успешная регистрация!",
Snackbar.LENGTH_SHORT).show();
                                        }
                                    }
                                });
                            }
                        });
                    });

                Intent intent = new Intent(ThirdActivity.this,
MainActivity.class);
                startActivity(intent);
                finish();
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                // Display the error message to the user
                Snackbar.make(binding.textView, "Registration
failed: " + e.getMessage(), Snackbar.LENGTH_LONG).show();
            }
        });
    }
}

```

Листинг 7. – *activity\_fourth.xml*.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:background="@color/background_material_light"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".FourthActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/text"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="/">

        <ListView
            android:id="@+id/lv"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_below="@+id/text"
            android:layout_marginTop="2dp"
            android:divider="@android:color/transparent"
            android:dividerHeight="10.0sp"
            android:scrollbars="vertical"
            tools:listitem="@layout/list_item" />

    </RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Листинг 8. – *FourthActivity.java*

```
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.myapplication.databinding.ActivityFourthBinding;
import com.example.myapplication.models.ListAdapter;
import com.example.myapplication.models.ListData;
```

```

import java.util.ArrayList;

public class FourthActivity extends AppCompatActivity {

    ListAdapter listAdapter;
    ArrayList<ListData> dataArrayList = new ArrayList<>();
    ListData listData;

    ActivityFourthBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityFourthBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        ListView lvMain = (ListView)findViewById(binding.lv.getId());
        final TextView txt = (TextView)findViewById(binding.text.getId());

        String[] parameters = {"О программе", "Инструкция пользования", "Об
авторе"};
        int[] text = {R.string.about_the_program,
R.string.Users_instructions, R.string.About_the_author};
        int[] image = {R.drawable.baseline_bubble_chart_24,
R.drawable.rounded_info_i_24, R.drawable.baseline_perm_identity_24};
        int[] image1 = {R.drawable.icon2,
R.drawable.icon_monopolia, R.drawable.iconc1};

        for (int i =0; i < image.length; i++){
            listData = new ListData(parameters[i], image[i]);
            dataArrayList.add(listData);
        }

        listAdapter = new ListAdapter(FourthActivity.this, dataArrayList);
        binding.lv.setAdapter(listAdapter);
        binding.lv.setClickable(true);

        binding.lv.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view,
int i, long l) {
                Intent intent = new Intent(FourthActivity.this,
DetailedActivity.class);
                intent.putExtra("parameters", parameters[i]);
                intent.putExtra("text", text[i]);
                intent.putExtra("image", image1[i]);
                startActivity(intent);
            }
        });
    }
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_marginStart="12dp"
android:layout_marginEnd="12dp"
android:layout_marginVertical="10dp"
android:layout_marginHorizontal="10dp"
app:cardCornerRadius="20dp"
app:cardElevation="8dp"
android:layout_width="match_parent"
android:layout_height="wrap_content">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        tools:ignore="MissingConstraints">

        <com.google.android.material.imageview.ShapeableImageView
            android:layout_width="60dp"
            android:layout_height="60dp"
            android:id="@+id/listImage"
            android:src="@drawable/baseline_bubble_chart_24"
            android:scaleType="centerCrop"
            android:layout_marginStart="10dp"
            android:layout_marginBottom="10dp"
            android:layout_marginTop="10dp"
            app:shapeAppearanceOverlay="@style/roundedImageViewRounded"/>

        <TextView
            android:layout_width="250dp"
            android:layout_height="30dp"
            android:id="@+id/listName"
            android:text="Name"
            android:textColor="@color/black"
            android:textSize="20sp"
            android:layout_marginTop="25dp"
            android:layout_marginStart="30dp"
            android:layout_toEndOf="@+id/listImage"/>

    </RelativeLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:background="@color/background_material_light"
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".DetailedActivity">

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/detailName"
    android:layout_marginTop="14dp"
    android:text="Recipe Name"
    android:textSize="24sp"
    android:textColor="@color/black"
    android:textStyle="bold"
    android:layout_gravity="center"/>

<androidx.cardview.widget.CardView
    android:layout_width="380dp"
    android:layout_height="200dp"
    android:id="@+id/cardImage"
    android:layout_gravity="center"
    app:cardCornerRadius="20dp"
    app:cardElevation="8dp">

    <ImageView
        android:layout_width="380dp"
        android:layout_height="200dp"
        android:id="@+id/detailImage"
        android:layout_gravity="center"
        android:src="@drawable/cat1"
        android:scaleType="centerCrop"/>
</androidx.cardview.widget.CardView>

<ScrollView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/about_the_program"
        android:layout_marginStart="10dp"
        android:layout_marginEnd="10dp"
        android:layout_marginTop="30dp"
        android:textSize="25sp"
        android:textStyle="bold"
        android:id="@+id/detailDesc"/>
</ScrollView>
</LinearLayout>

```

*Листинг 11. – DetailedActivity.java.*

```

package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import com.example.myapplication.databinding.ActivityDetailedBinding;

public class DetailedActivity extends AppCompatActivity {

    ActivityDetailedBinding binding;
}

```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivityDetailedBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    Intent intent = this.getIntent();
    if (intent != null){
        String parameters = intent.getStringExtra("parameters");
        int text = intent.getIntExtra("text",
R.string.about_the_program);
        int image = intent.getIntExtra("image",
R.drawable.baseline_bubble_chart_24);

        binding.detailImage.setImageResource(image);
        binding.detailName.setText(parameters);
        binding.detailDesc.setText(text);
    }
}
}

```

*Листинг 12. – User.java.*

```

package com.example.myapplication.models;

public class User {
    public String email, name, password;
    float sum;
    private boolean admin;
    public User(){
    }

    public User(String email, String name, String password, float sum) {
        this.email = email;
        this.password = password;
        this.name = name;
        this.sum = sum;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}

```

```

    }

    public boolean getAdmin() {
        return admin;
    }

    public void setAdmin(boolean admin) {
        this.admin = admin;
    }

    public float getSum() {
        return sum;
    }

    public void setSum(float sum) {
        this.sum = sum;
    }
}

```

*Листинг 13. – ListData.java.*

```

    package com.example.myapplication.models;

    public class ListData {
        String name;
        int image;

        public ListData(String name, int image) {
            this.name = name;
            this.image = image;
        }
    }
}

```

*Листинг 14. – ListAdapter.java*

```

    package com.example.myapplication.models;

    import android.content.Context;
    import android.view.LayoutInflater;
    import android.view.View;
    import android.view.ViewGroup;
    import android.widget.ArrayAdapter;
    import android.widget.ImageView;
    import android.widget.TextView;

    import androidx.annotation.NonNull;
    import androidx.annotation.Nullable;

    import com.example.myapplication.R;

    import java.util.ArrayList;

    public class ListAdapter extends ArrayAdapter<ListData> {
        public ListAdapter(@NonNull Context context, ArrayList<ListData>
dataArrayList) {
            super(context, R.layout.list_item, dataArrayList);
        }

        @NonNull
        @Override

```

```

    public View getView(int position, @Nullable View view, @NonNull
ViewGroup parent) {
        ListData listData = getItem(position);

        if (view == null){
            view =
LayoutInflater.from(getContext()).inflate(R.layout.list_item, parent,
false);
        }

        ImageView listImage = view.findViewById(R.id.listImage);
        TextView listName = view.findViewById(R.id.listName);

        listImage.setImageResource(listData.image);
        listName.setText(listData.name);

        return view;
    }
}

```