



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

## **Отчет по практическим работам №1-4**

по дисциплине «Системная и программная инженерия»

**Выполнили:**

Студенты группы ИКБО-11-22

Берчик А.С.  
Андрусенко Л.Д.  
Гришин А.В.  
Малкин Г.Д.  
Гоппен С.Д.

**Проверил:**

Преподаватель Михайлова Е.К.

## СОДЕРЖАНИЕ

1. ОБСЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	3
1.1. Выбор предметной области для разработки и распределение на команды.....	3
1.2. Ход работы.....	4
2. ОПРЕДЕЛЕНИЕ ФУНКЦИОНАЛЬНЫХ И ПОЛЬЗОВАТЕЛЬСКИХ ТРЕБОВАНИЙ.....	5
2.1. Назначение системы.....	5
2.2. User story.....	6
2.3. Функциональные требования.....	6
2.4. Первоначальный план работы и организация рабочего пространства...8	
3. НАЧАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ПРОЕКТА. 9	
3.1. Портрет пользователя.....	9
3.2. Целевая аудитория.....	10
3.2.1. Финансово грамотные пользователи.....	10
3.2.2. Начинающие пользователи, стремящиеся к финансовой грамотности.....	10
3.2.3. Семейные пользователи.....	11
3.3. Use Case.....	11
3.4. Событийная диаграмма.....	12
4. МАТРИЦА ТРЕБОВАНИЙ.....	14
4.1. Нефункциональные требования.....	14
4.2. Матрица требований.....	16

# 1. ОБСЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Выбор предметной области для разработки и распределение на команды

Командная работа подразумевает распределение обязанностей и ролей, каждая из которых включает в себя определенный набор задач. Один человек либо не может реализовывать крупный проект, либо реализация займет достаточно продолжительный период, в связи с большой нагрузкой. Для более грамотного распределения на команды, обозначим роли и функции, которые они в себя включают.

**Руководитель группы (менеджер проекта)** — промежуточное звено между командой и клиентом, координирует процесс разработки, решает проблемы и разногласия, следит за сроками и качеством выполнения задач. Использует трекеры (например, Trello) для распределения задач, установки сроков и мониторинга выполнения.

**Мобильный разработчик** — разработка под мобильные платформы.

**Тестировщик** — тестирует завершённые модули перед релизом, снижая риски ошибок на более поздних этапах. Не сочетается с ролью разработчика, но может работать с аналитиком.

**Аналитик** — описывает процессы в предметной области, создаёт диаграммы и спецификации, чтобы точно реализовать бизнес-логику приложения.

**Технический писатель** — формирует техническое задание и документацию, вступает в процесс реализации ПО на первых этапах.

**Дизайнер** — занимается созданием фирменного стиля, проектированием и отрисовкой интерфейсов на основе требований из основного ТЗ проекта.

## 1.2. Ход работы

Была определена команда из 5 человек со следующими ролями (Таблица 1.1).

*Таблица 1.1 – Участники команды и исполняемые роли*

<b><i>ФИО</i></b>	<b><i>Роль</i></b>
Берчик Анастасия Сергеевна	Менеджер, технический писатель
Андрусенко Лада Дмитриевна	Аналитик, тестировщик
Гришин Андрей Валерьевич	Разработчик
Малкин Глеб Дмитриевич	Дизайнер, аналитик
Гоппен Сергей Дмитриевич	Разработчик

Была выбрана тематика мобильных приложений. Тема проекта «Мобильное приложение учёт финансов».

## **2. ОПРЕДЕЛЕНИЕ ФУНКЦИОНАЛЬНЫХ И ПОЛЬЗОВАТЕЛЬСКИХ ТРЕБОВАНИЙ**

### **2.1. Назначение системы**

#### **1. Управление личными финансами.**

Система должна предоставлять пользователю инструменты для эффективного управления личными и семейными финансами. Она позволит отслеживать доходы, расходы, кредиты и ссуды, анализировать финансовые данные и прогнозировать будущие финансовые потоки. Это поможет пользователям лучше контролировать свои финансы и принимать обоснованные решения.

#### **2. Повышение финансовой грамотности.**

В процессе использования системы пользователь будет сталкиваться с различными аспектами финансового планирования, что способствует повышению его финансовой грамотности. Это включает в себя понимание структуры бюджета, управление долгами и планирование будущих расходов.

#### **3. Создание персонализированного финансового опыта.**

Система будет предоставлять персонализированные отчеты и рекомендации на основе введенных пользователем данных. Это позволит каждому пользователю получать уникальные инсайты и советы, которые помогут ему улучшить свое финансовое состояние.

## 2.2. User story

В таблице 2.1 представлены пользовательские истории – описание функциональной возможности ПО простыми, общими словами, составленное с точки зрения конечного пользователя – в структурированном виде.

Таблица 2.1 – Перечень user story

<i><b>Кто?</b></i>	<i><b>Что хочет?</b></i>	<i><b>С какой целью?</b></i>
Пользователь	Регистрироваться и входить в систему	Иметь персонализированный доступ к своим финансовым данным
	Добавлять источники доходов	Отслеживать все поступления
	Добавлять расходы	Контролировать свои траты
	Категоризировать доходы и расходы	Легко анализировать свои финансы
	Видеть графики и отчеты по доходам и расходам	Понимать структуру своего бюджета
	Устанавливать лимиты на расходы по категориям	Не превышать бюджет
	Получать уведомления о приближении к лимитам расходов	Своевременно корректировать свои траты
	Добавлять информацию о кредитах и ссудах	Учитывать их в общем бюджете
	Прогнозировать будущие доходы и расходы	Планировать свой бюджет
	Экспортировать данные в различных форматах (например, CSV)	Использовать их в других программах

## 2.3. Функциональные требования

Функциональные требования, составленные на основе пользовательских историй:

### 1. Регистрация и авторизация.

- У пользователя должна быть возможность регистрироваться и входить в систему с использованием email и пароля.
- Система должна поддерживать восстановление пароля.

### 2. Управление доходами.

- У пользователя должна быть возможность добавлять,

редактировать и удалять источники доходов.

- Каждый доход должен иметь дату, сумму и категорию.

### 3. Управление расходами.

- У пользователя должна быть возможность добавлять, редактировать и удалять расходы.
- Каждый расход должен иметь дату, сумму и категорию.

### 4. Категоризация.

- У пользователя должна быть возможность создавать, редактировать и удалять категории для доходов и расходов.

### 5. Отчеты и аналитика.

- Система должна генерировать графики и отчеты по доходам и расходам за выбранный период.
- Пользователь должен иметь возможность фильтровать данные по категориям и датам.

### 6. Управление лимитами.

- У пользователя должна быть возможность устанавливать лимиты на расходы по категориям.
- Система должна уведомлять пользователя о приближении к установленным лимитам.

### 7. Управление кредитами и ссудами.

- У пользователя должна быть возможность добавлять, редактировать и удалять информацию о кредитах и ссудах.
- Система должна учитывать кредиты и ссуды в общем бюджете.

### 8. Экспорт данных.

- У пользователя должна быть возможность экспортировать финансовые данные в форматах CSV и других.

### 9. Прогнозирование.

- Система должна предоставлять инструменты для прогнозирования будущих доходов и расходов на основе исторических данных.

#### 10. Уведомления.

- Система должна отправлять уведомления пользователю о приближении к установленным лимитам расходов.

### **2.4. Первоначальный план работы и организация рабочего пространства**

Менеджером проекта был разработан первоначальный план разработки продукта. В данный план вошли следующие пункты:

1. Подготовка к разработке
  - Анализ требований и целевой аудитории..
2. Разработка прототипа
  - Создание и тестирование макетов интерфейса.
  - Сбор обратной связи и внесение изменений.
3. Разработка приложения
  - Определение архитектуры и разработка функционала.
  - Тестирование и исправление ошибок.
4. Завершение и подготовка к запуску
  - Оптимизация производительности.
  - Запуск приложения.

Также, было создано рабочее пространство на основе Jira-подобного сервиса(YouGile) для распределения задач между участниками проекта (Рисунок 2.1).



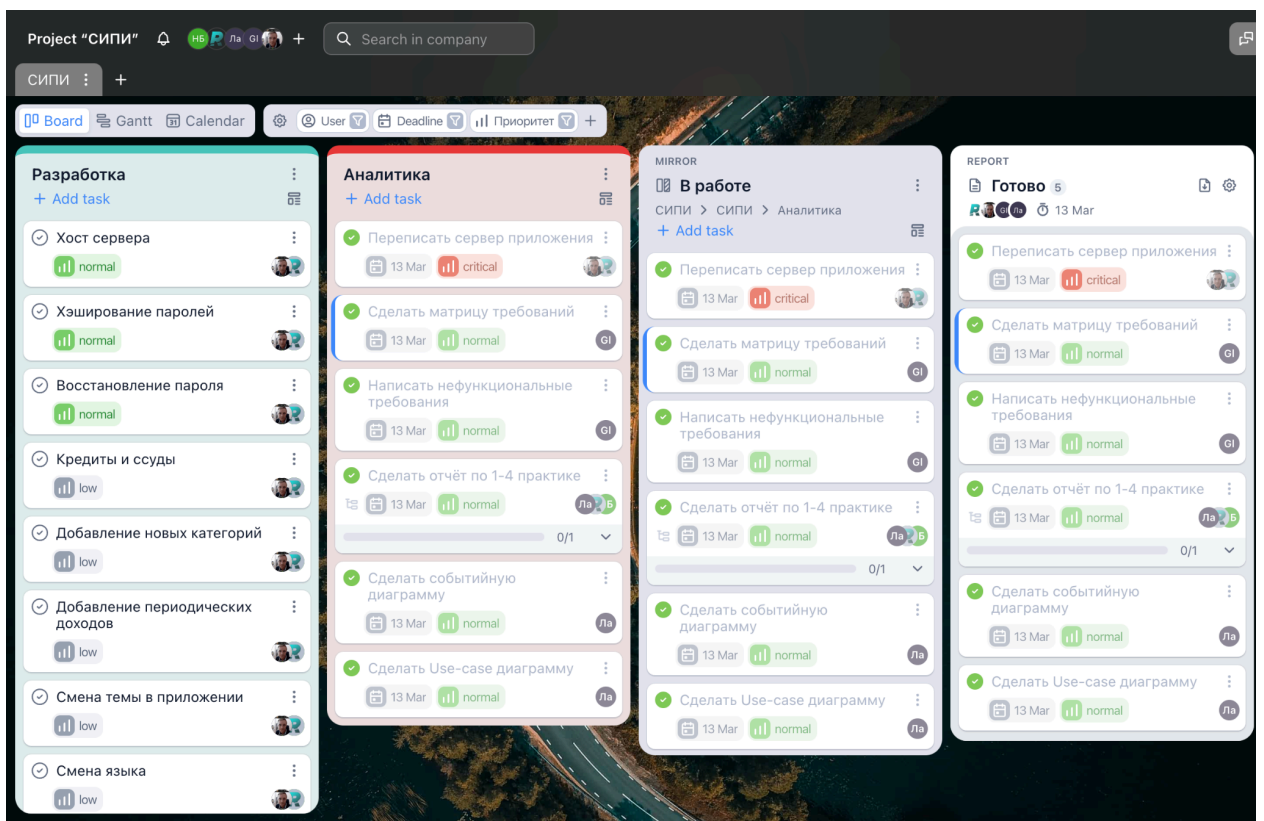


Рисунок 2.1 – Страница рабочего пространства команды

### **3. НАЧАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ПРОЕКТА**

#### **3.1. Портрет пользователя**

Платформа для учёта бюджета ориентирована на пользователей, которые стремятся к финансовой стабильности и осознанному управлению своими доходами и расходами. Это аудитория, которая воспринимает управление бюджетом не только как необходимость, но и как инструмент для достижения финансовых целей и улучшения качества жизни.

Основной пользователь — человек, который ценит контроль над своими финансами и стремится к их оптимизации. Он привык к использованию цифровых инструментов для управления личными данными и готов инвестировать время в анализ и планирование своего бюджета. Это может быть как начинающий пользователь, который только начинает осваивать основы финансовой грамотности, так и опытный пользователь, который ищет продвинутые инструменты для анализа и прогнозирования.

Концепция платформы, связанная с управлением личными финансами, привлекает пользователей, интересующихся финансовым планированием, инвестициями и достижением финансовой независимости. Это люди, которые ищут решения для эффективного управления своими доходами, расходами, кредитами и ссудами.

Платформа также найдет отклик среди тех, кто стремится к финансовой дисциплине и готов использовать современные технологии для достижения своих целей. Разнообразие функций (управление доходами, расходами, прогнозирование) делает её привлекательной для широкой аудитории, включая тех, кто ценит удобство и эффективность в управлении своими финансами.

## **3.2. Целевая аудитория**

### **3.2.1. Финансово грамотные пользователи**

Основной сегмент — активные пользователи финансовых приложений и сервисов. Это люди, которые уже имеют опыт в управлении своими финансами и ищут инструменты для дальнейшего улучшения своих навыков. Они ценят функциональность, удобство использования и возможность персонализации.

Эта аудитория предпочитает проекты, где финансовый менеджмент выходит за рамки простого учета, предлагая аналитические инструменты и прогнозирование. Они привыкли к использованию цифровых решений для управления своими финансами и ожидают высокого уровня функциональности и безопасности.

### **3.2.2. Начинающие пользователи, стремящиеся к финансовой грамотности**

Этот сегмент ориентирован на тех, кто только начинает осваивать основы управления личными финансами. Они ищут простые и интуитивно понятные инструменты, которые помогут им начать контролировать свои доходы и расходы.

Их привлекают приложения с образовательным компонентом, которые не только помогают отслеживать финансы, но и обучают основам финансовой грамотности. Они ценят советы и рекомендации, которые помогают им лучше понимать свои финансовые привычки и принимать обоснованные решения.

### 3.2.3. Семейные пользователи

Эти пользователи ищут решения для управления семейным бюджетом. Они ценят возможность совместного использования платформы с другими членами семьи и функции, которые помогают планировать и контролировать семейные расходы.

Им важны инструменты для бюджетирования, которые позволяют распределять доходы и расходы по категориям, устанавливать лимиты и отслеживать прогресс в достижении финансовых целей.

## 3.3. Use Case

Цель use case диаграммы — формирование требований к системе, понятных заказчику.

Основные элементы диаграммы:

- **Участник (actor)** — роли, взаимодействующие с системой. Это может быть человек, другая система или подсистема. Изображается в виде "человечка".
- **Прецедент (use case)** — описание последовательности действий, выполняемых системой, приводящих к результату, который наблюдает участник. Прецедент показывает, "что" выполняется, а не "как". Изображается в виде эллипса с названием.

На диаграмме отображаются:

- акторы (группы лиц или систем, взаимодействующих с системой);
- прецеденты (сервисы, предоставляемые системой актерам);
- комментарии;
- отношения между элементами диаграммы.

Схема use case можно посмотреть на рисунке 3.1.

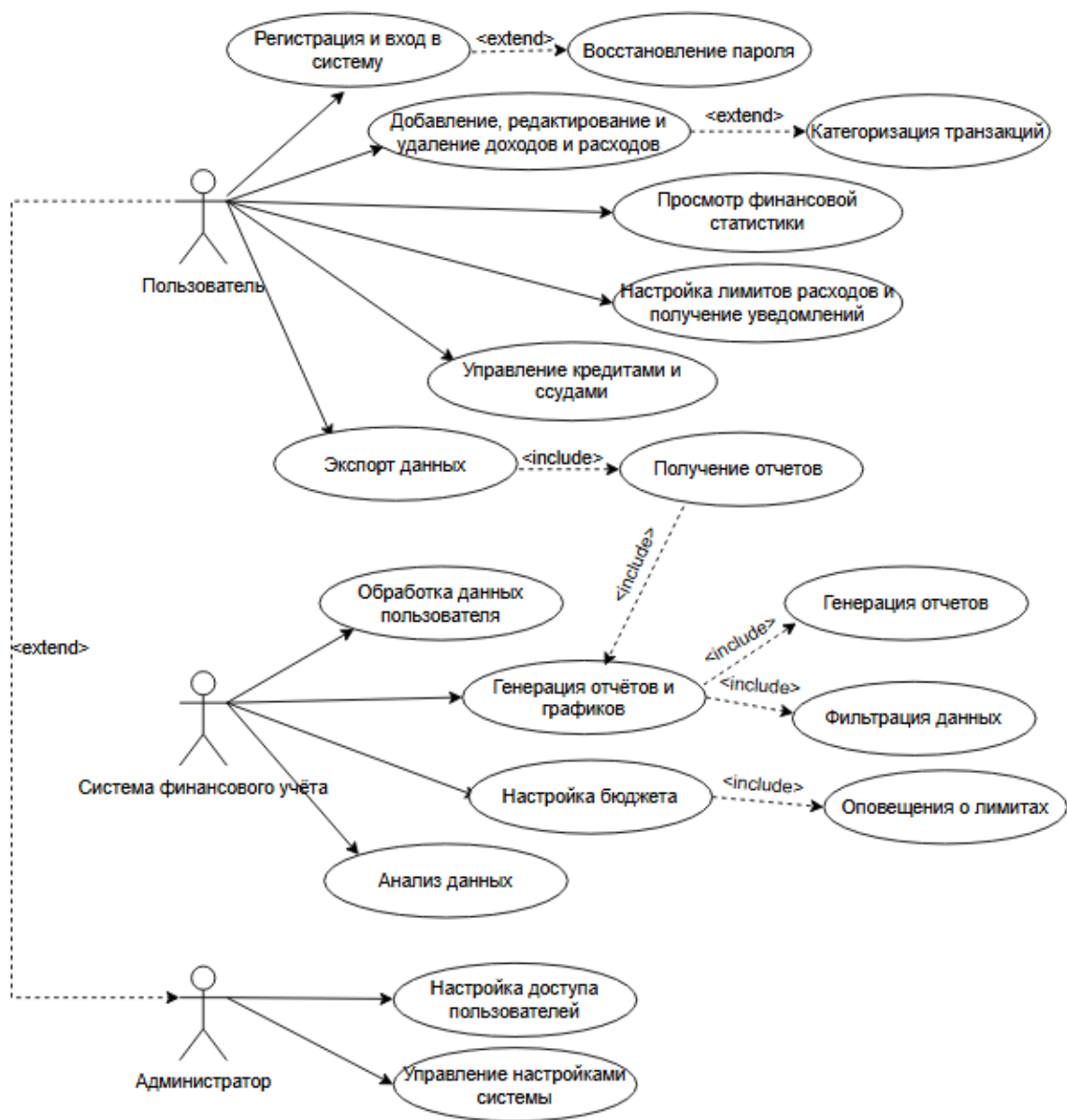


Рисунок 3.1 – Use case

### 3.4. Событийная диаграмма

Нотация EPC (Event-Driven Process Chain – событийная цепочка процессов) используется для описания процессов нижнего уровня. Диаграмма процесса в нотации EPC, представляет собой упорядоченную комбинацию событий и функций. Для каждой функции могут быть определены начальные и конечные события, участники, исполнители, материальные и документальные потоки, сопровождающие её, а также

проведена декомпозиция на более низкие уровни. Декомпозиция может производиться в нотациях EPC или BPMN.

Диаграмма реализована на рисунке 3.2.

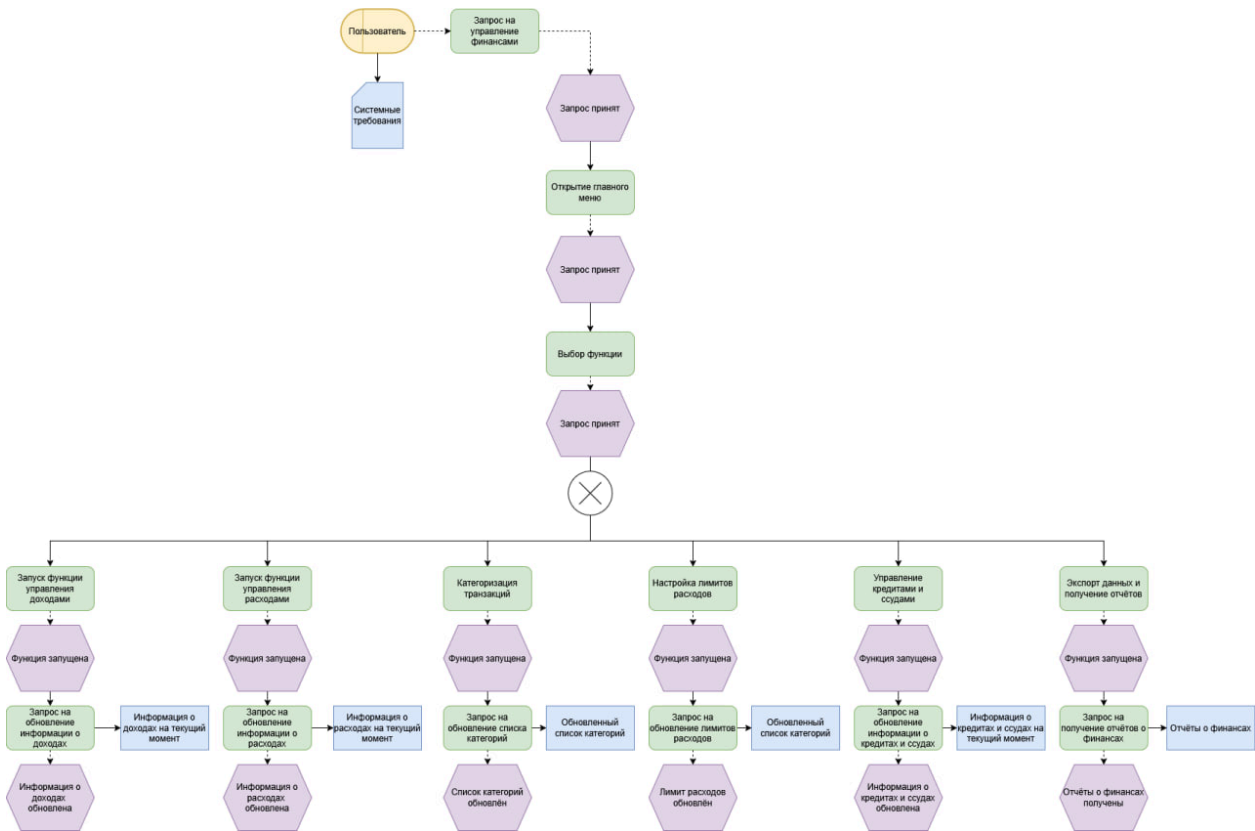


Рисунок 3.2 – Событийная диаграмма

## 4. МАТРИЦА ТРЕБОВАНИЙ

### 4.1. Нефункциональные требования

**Нефункциональные требования** — это условия, при которых продукт должен работать, и качества, которыми он должен обладать (например, производительность, надежность, масштабируемость).

Если совсем просто, то к нефункциональным относят те требования, которые не вошли в функциональные.

Нефункциональные требования должны быть измеримы. Их можно проверить и протестировать.

К нефункциональным требованиям относятся: Технические ограничения, бизнес-требования, локализация, доступность, производительность и масштабируемость, надежность, безопасность, удобство использования.

В таблице 4.3 приведены нефункциональные требования к каждому типу требований: технические ограничения, производительность, надежность и безопасность.

*Таблица 4.1 – Нефункциональные требования*

Тип требования	Содержание требования
Технические требования	Поддержка платформ: система должна быть доступна на мобильных устройствах (Android).
	Операционные системы: приложение должно поддерживать следующие версии Android: минимальная поддерживаемая версия: Android 12, рекомендуемая версия: Android 13. Также

	приложение будет обеспечивать поддержку будущих версий Android.
Производительность	Время отклика: время отклика системы на действия пользователя (например, добавление расхода) не должно превышать 2 секунд.
	Время загрузки: время загрузки основных страниц системы не должно превышать 3 секунд.
	Обработка данных: система должна обрабатывать и генерировать отчеты для пользователей с базой данных до 10,000 транзакций за 5 секунд.
Масштабируемость	Поддержка роста пользователей: система должна поддерживать увеличение количества пользователей до 100,000 без значительного снижения производительности.
	Расширение функционала: архитектура системы должна позволять добавление новых функций (например, новых типов отчетов) без необходимости полной переработки кода.
	Система должна быть способна обрабатывать и хранить до 100 Гб финансовых данных.
Надежность	Восстановление данных: в случае сбоя системы, данные пользователей должны быть восстановлены автоматически, с минимальной потерей информации.



	Обработка ошибок: система должна корректно обрабатывать ошибки (например, потерю интернет-соединения) и уведомлять пользователя о проблеме.
Безопасность	Защита данных: все данные пользователей должны быть зашифрованы как при хранении, так и при передаче. Для шифрования данных при хранении используется симметричное шифрование (AES-256) с защитой ключей в Android Keystore System. Конфиденциальные данные, такие как пароли, дополнительно защищаются с помощью хеширования (SHA-256). При передаче данных применяется TLS/SSL (HTTPS) для шифрования канала связи, а также дополнительное шифрование на уровне приложения с использованием асимметричных алгоритмов (RSA) для защиты конфиденциальной информации.
	Конфиденциальность: ФЗ-152 "О персональных данных"

## 4.2. Матрица требований

Матрица требований – формализованный вид всех требований к разработке проекта. Матрица требований состоит из функциональных требований и нефункциональных требований.

Необходимые свойства к требованиям в матрице требований:

1) измеримость (SLI – индикатор/показатель (можем померить), SCO – критерий (порог, критическое значение), SLA – соглашение/договор);

- 2) полнота;
- 3) идентифицируемость (возможность четкого определения каждого требования без смешивания с другими требованиями);
- 4) прослеживаемость (отслеживание реализации требования в конечном продукте).

Источники требований в матрице требований:

1. Федеральное законодательство.\*
2. Региональное и отраслевое регулирование.\*
3. Стандарты.\*
4. Документация на технологические решения.\*
5. Работа с заказчиком (анкетирование, мозговой штурм).\*
6. Моделирование (бизнес-процессы, математическое моделирование, имитационное моделирование).
7. Системы–аналоги и их анализ.

*Примечание: пункт с \* – хранилища готовых требований*

На таблице 4.2 представлена матрица требований, составленная совместно со всеми участниками команды.

*Таблица 4.2 – Матрица требований*

Составляющая	Требования	Суть	Автор	Ссылки	Критерий проверки
Регистрация и авторизация	Регистрация пользователей	Пользователь должен иметь возможность зарегистрироваться с использованием email и пароля.	Разработчик	Функциональные требования	Успешная регистрация и создание аккаунта.
	Авторизация	Пользователь должен иметь возможность входить в систему.	Разработчик	Функциональные требования	Успешный вход в систему.

	Восстановлен ие пароля	Система должна поддерживать восстановление пароля.	Разработчик	Функциональн ые требования	Успешное восстановлен ие пароля.
Управление доходами	Добавление доходов	Пользователь должен иметь возможность добавлять доходы.	Разработчик	Функциональн ые требования	Корректное добавление дохода.
	Редактировани е доходов	Пользователь должен иметь возможность редактировать доходы.	Разработчик	Функциональн ые требования	Успешное изменение данных о доходе.
	Удаление доходов	Пользователь должен иметь возможность удалять доходы.	Разработчик	Функциональн ые требования	Успешное удаление дохода.
Управление расходами	Добавление расходов	Пользователь должен иметь возможность добавлять расходы.	Разработчик	Функциональн ые требования	Корректное добавление расхода.
	Редактировани е расходов	Пользователь должен иметь возможность редактировать расходы.	Разработчик	Функциональн ые требования	Успешное изменение данных о расходе.
	Удаление расходов	Пользователь должен иметь возможность удалять расходы.	Разработчик	Функциональн ые требования	Успешное удаление расхода.

Категоризация	Создание категорий	Пользователь должен иметь возможность создавать категории.	Дизайнер	Функциональные требования	Успешное создание категории.
	Редактирование категорий	Пользователь должен иметь возможность редактировать категории.	Дизайнер	Функциональные требования	Успешное изменение категории.
	Удаление категорий	Пользователь должен иметь возможность удалять категории.	Дизайнер	Функциональные требования	Успешное удаление категории.
Отчеты и аналитика	Генерация отчетов	Система должна генерировать отчеты по доходам и расходам.	Менеджер проекта	Функциональные требования	Корректная генерация отчетов.
	Фильтрация данных	Пользователь должен иметь возможность фильтровать данные.	Менеджер проекта	Функциональные требования	Успешная фильтрация данных.
Управление лимитами	Установка лимитов	Пользователь должен иметь возможность устанавливать лимиты.	Разработчик	Функциональные требования	Успешная установка лимита.
	Уведомления о лимитах	Система должна уведомлять о	Тестировщик	Функциональные требования	Появление уведомлений.

		приближении к лимитам.			
Управление кредитами и ссудами	Добавление кредитов/ссуд	Пользователь должен иметь возможность добавлять кредиты/ссуды.	Разработчик	Функциональные требования	Корректное добавление данных.
	Редактирование кредитов/ссуд	Пользователь должен иметь возможность редактировать кредиты/ссуды.	Разработчик	Функциональные требования	Успешное изменение данных.
	Удаление кредитов/ссуд	Пользователь должен иметь возможность удалять кредиты/ссуды.	Разработчик	Функциональные требования	Успешное удаление данных.
Экспорт данных	Экспорт в CSV	Пользователь должен иметь возможность экспортировать данные в CSV.	Разработчик	Функциональные требования	Успешный экспорт данных.
Прогнозирование	Прогнозирование доходов/расходов	Система должна предоставлять инструменты для прогнозирования.	Менеджер проекта	Функциональные требования	Корректное отображение прогнозов.
Уведомления	Уведомления о лимитах	Система должна отправлять уведомления о лимитах.	Тестировщик	Функциональные требования	Появление уведомлений.

Поддержка платформ	Мобильные устройства	Система должна работать на Android.	Менеджер проекта	Нефункциональные требования	Работоспособность на мобильных устройствах.
Производительность	Время отклика	Время отклика не должно превышать 2 секунд.	Разработчик	Нефункциональные требования	Время отклика $\leq 2$ секунд.
	Время загрузки	Время загрузки страниц не должно превышать 3 секунд.	Менеджер проекта	Нефункциональные требования	Время загрузки $\leq 3$ секунд.
	Обработка данных	Система должна обрабатывать до 10,000 транзакций за 5 секунд.	Разработчик	Нефункциональные требования	Обработка данных в указанные сроки.
Масштабируемость	Поддержка роста пользователей	Система должна поддерживать до 100,000 пользователей.	Менеджер проекта	Нефункциональные требования	Стабильная работа при росте пользователей.
	Поддержка больших объемов данных	Система должна обрабатывать большие объемы данных.	Разработчик	Нефункциональные требования	Обработка данных без снижения производительности.
Надежность	Восстановление данных	В случае сбоя данные должны восстанавливаться автоматически.	Разработчик	Нефункциональные требования	Успешное восстановление данных.

	Обработка ошибок	Система должна уведомлять о проблемах.	Тестировщик	Нефункциональные требования	Корректное отображение уведомлений.
Безопасность	Защита данных	Все данные должны быть зашифрованы.	Разработчик	Нефункциональные требования	Шифрование данных.
	Конфиденциальность	Система должна соответствовать требованиям регуляторов.	Разработчик	Нефункциональные требования	Соответствие стандартам защиты данных.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения практических работ была проведена комплексная работа по анализу, проектированию и моделированию системы управления личными финансами. В процессе работы команда выполнила обследование предметной области, определила функциональные и пользовательские требования, создала начальные модели и разработала матрицу требований.

Разработанная система учитывает различные аспекты финансового планирования, включая управление доходами и расходами, прогнозирование финансовых потоков, анализ данных и формирование отчетности. Предложенные нефункциональные требования обеспечивают надежность, безопасность и удобство использования системы.

Работа позволила участникам углубить знания в области системной инженерии, научиться работать в команде и применять методы анализа и моделирования в рамках реального проекта. Полученные результаты могут быть использованы в дальнейшей разработке программного продукта и его внедрении.