



## Аннотация

Данная выпускная квалификационная работа посвящена разработке мобильного приложения для учета и управления переводами денежных средств поставщикам услуг в компании «Квартплата 24». В работе анализируются существующие подходы к учету финансовых транзакций и рассматривается роль мобильных приложений в упрощении финансовых процессов. Теоретическая часть включает выбор методов разработки мобильных приложений и описание процесса проектирования и реализации приложения. Практическая часть представляет описание разработанного мобильного приложения, его функционала и интеграции с существующими системами учета.

Целью выпускной квалификационной работы является создание функционального мобильного приложения, способного упростить учет и управление переводами денежных средств поставщикам услуг. Объектом исследования является процесс учета переводами денежных средств поставщикам услуг в компании «Квартплата 24». Предметом исследования является разработка мобильного приложения, способного автоматизировать и упростить данный процесс. Для достижения этой цели были сформулированы следующие задачи:

- провести анализ требований к приложению и определить основные функциональные возможности;
- разработать удобный и интуитивно понятный интерфейс;
- реализовать функционал для учета финансовых транзакций и взаимодействия с базами данных компании;
- провести тестирование и оптимизацию приложения для обеспечения его стабильной работы и безопасности данных.

Результатом исследования является эффективное мобильное приложение, способствующее повышению эффективности управления финансовыми потоками в компании.

## **Abstract**

The title of the final qualifying work: "Development of a mobile application for accounting for money transfers to service providers".

The final work consists of an introduction, three chapters, a conclusion, tables, a list of references, including foreign sources, and a graphic part.

The key issue of the final qualifying work is the creation of a functional mobile application that can simplify the accounting and management of money transfers to service providers. We address the issue of the accounting process for money transfers to service providers. The purpose of this study is to create a functional mobile application that can simplify the accounting and management of money transfers to service providers.

The final qualification work can be divided into the following logically interrelated parts: analysis of business processes; review of existing solutions in the field of accounting for financial transactions; definition of system requirements; setting the task for application development.

The result of the research is an effective mobile application that helps to improve the efficiency of financial flow management in the company. The work is of practical importance for optimizing accounting and financial management processes in the organization «Kvartplata 24» and can be used in the development of similar applications in other business areas.

## Оглавление

Введение.....	5
Глава 1 Анализ предметной области.....	7
1.1 Описание организации ООО «Квартплата 24» .....	7
1.2 Анализ бизнес-процессов .....	13
1.3 Обзор существующих решений в области учета финансовых транзакций.....	17
1.4 Определение требований к системе .....	24
1.5 Постановка задачи на разработку приложения.....	27
Глава 2 Проектирование и разработка мобильного «Умный Бюджет» .....	30
2.1 Обоснование выбора средств разработки .....	30
2.2 Проектирование структуры приложения.....	32
2.3 Логическое и физическое моделирование .....	37
2.4 Разработка интерфейсов и архитектурной модели приложения.....	41
Глава 3 Разработка дизайна и тестирование приложения .....	44
3.1 Реализация дизайна приложения .....	44
3.2 Реализация функциональных модулей.....	48
3.3 Тестирование приложения .....	52
3.4 Оценка эффективности проекта.....	54
Заключение .....	58
Список используемых источников.....	60
Приложение А Код модуля регистрации и аутентификации пользователей	64
Приложение Б Код модуля управления переводами.....	65

## Введение

В условиях современного информационного общества, где цифровые технологии играют ключевую роль в организации бизнес-процессов, разработка эффективных информационных систем для управления финансовыми потоками становится необходимостью. Особую значимость приобретает это в контексте компаний, оказывающих услуги и взаимодействующих с различными поставщиками.

Актуальность темы обусловлена необходимостью оптимизации процессов учета и управления финансовыми транзакциями в компаниях, в том числе и в контексте взаимодействия с поставщиками услуг. Мобильные приложения становятся все более популярным инструментом для облегчения финансовых процессов, что делает разработку подобного приложения крайне актуальной.

Целью данного исследования является создание функционального мобильного приложения, способного упростить учет и управление переводами денежных средств поставщикам услуг в организации «Квартплата 24». Для достижения этой цели были сформулированы следующие задачи:

- провести анализ требований к приложению и определить основные функциональные возможности;
- разработать удобный и интуитивно понятный интерфейс пользователя;
- реализовать функционал для учета финансовых транзакций и взаимодействия с базами данных компании;
- провести тестирование и оптимизацию приложения для обеспечения его стабильной работы и безопасности данных.

Объектом исследования является процесс учета и управления переводами денежных средств поставщикам услуг в компании «Квартплата 24». Предметом исследования является разработка мобильного приложения, способного автоматизировать и упростить данный процесс.

Для разработки мобильного приложения выбран итеративный метод. Этот выбор обоснован его гибкостью и способностью быстро реагировать на изменения в требованиях проекта. Итеративный подход позволит поэтапно разрабатывать и улучшать приложение, что является ключевым фактором для успешной реализации проекта.

Результаты данного исследования могут быть применены в практике компании «Квартплата 24» для оптимизации процессов учета финансовых транзакций и повышения эффективности взаимодействия с поставщиками услуг. Кроме того, разработанное приложение может быть использовано в других организациях, оказывающих аналогичные услуги.

Полученные результаты и разработанное приложение будут протестированы в реальных условиях работы компании «Квартплата 24» с целью оценки их эффективности и соответствия поставленным требованиям.

Данная работа состоит из введения, содержательной части, заключения и списка использованных источников. Содержательная часть включает в себя теоретический обзор, методологию и методы исследования, практическую часть с описанием разработки приложения, выводы и рекомендации.

## **Глава 1 Анализ предметной области**

### **1.1 Описание организации ООО «Квартплата 24»**

«ООО «Квартплата 24» было зарегистрировано 16 декабря 2010 года по адресу: д. 31, Приморский б-р, город Тольятти, Самарская область, индекс 445057. Регистратором организации выступает Межрайонная инспекция Федеральной налоговой службы № 24 по Самарской области. Форма собственности - частная, с уставным капиталом 10 тысяч рублей. Компания "Квартплата 24" специализируется на ИТ-услугах, в частности на решении задач, связанных с расчетом и учетом платы за жилищно-коммунальные услуги, а также приемом, распределением платежей и взысканием долгов в соответствии с законодательством для различных организаций РФ. Экосистема сервисов компании представляет собой более 10 тесно интегрированных облачных сервисов, предназначенных для управления и информационного обслуживания различных организаций по всей территории России» [16].

«Основными реквизитами организации выступают: ИНН 6321258740; ОГРН 1106320023534; КПП 632101001» [16].

Цель организации: обеспечить удобство и эффективность в управлении жилищно-коммунальными платежами клиентов через онлайн-сервис, объединяющий оплату квитанций, отслеживание задолженностей и доступ к службам поддержки.

Миссия организации: создать удобную и надежную платформу для оптимизации управления финансами, обеспечивая жителям доступ к онлайн-сервисам оплаты, отслеживания и управления жилищно-коммунальными платежами.

Основные задачи организации:

- развитие и поддержание функциональности онлайн-сервиса для оптимального управления жилищно-коммунальными платежами;

- обеспечение надежности и безопасности данных пользователей в системе;
- расширение спектра услуг, предоставляемых через платформу, с учетом потребностей клиентов;
- повышение качества обслуживания клиентов через эффективное взаимодействие с службами поддержки;
- привлечение новых клиентов и увеличение удовлетворенности текущих пользователей за счет постоянного совершенствования предоставляемых услуг;
- внедрение инновационных технологий и методов работы для оптимизации процессов управления жилищно-коммунальными платежами.

«Директор ООО «Кварплата24» несет ответственность за руководство и управление ее деятельностью. Его основные обязанности включают в себя» [16]:

- руководство всеми аспектами деятельности организации в соответствии с уставом и законодательством;
- определение стратегии развития компании и формирование ее бизнес-планов;
- обеспечение выполнения поставленных целей и задач организации;
- управление финансовой деятельностью, в том числе контроль за бюджетом, инвестициями и финансовой отчетностью;
- организация работы с персоналом, включая найм, обучение и мотивацию сотрудников;
- представление интересов организации перед сторонними органами и клиентами;
- разработка и подписание соглашений, контрактов и других юридических документов;

- организация контроля за качеством услуг и выполнением стандартов обслуживания клиентов;
- ведение диалога с акционерами и участниками организации, обеспечение прозрачности деятельности компании;
- принятие оперативных решений по текущим вопросам функционирования организации.

«Главный бухгалтер отвечает за финансовые операции компании и подчиненных ей бухгалтеров. К обязанностям относятся» [16]:

- ведение бухгалтерского учета и составление отчетности в соответствии с действующим законодательством;
- контроль за правильностью и своевременностью проведения финансовых операций;
- подготовка и представление налоговой отчетности в установленные сроки;
- разработка и внедрение системы внутреннего контроля и учета финансовых операций;
- обеспечение соблюдения налогового и бухгалтерского законодательства при взаимодействии с контрагентами и клиентами;
- оценка финансового состояния организации и разработка рекомендаций по оптимизации финансовых процессов;
- участие в аудиторских проверках и обеспечение соблюдения рекомендаций аудиторов;
- обеспечение своевременного предоставления финансовой информации руководству для принятия управленческих решений;
- управление бухгалтерским отделом, включая найм и обучение персонала;
- сотрудничество с внешними консультантами и аудиторскими фирмами по вопросам финансового учета и отчетности.

«Бухгалтер-юрист, выполняет ряд обязанностей, связанных с оказанием качественных услуг своим клиентам. Ниже приведены основные обязанности бухгалтера-юриста в ООО «Кварплата24» [16]:

- обеспечение соблюдения бухгалтерского и налогового законодательства при ведении бухгалтерского учета и подготовке финансовой отчетности;
- предоставление юридического консультирования в области налогообложения, финансового права и бухгалтерского учета;
- разработка и анализ договоров, контрактов и других юридических документов, связанных с финансовой деятельностью организации;
- участие в аудиторских проверках и обеспечение соблюдения юридических норм и требований при проведении аудита;
- подготовка и представление юридических позиций организации в судебных и административных процедурах;
- оценка юридических рисков и разработка стратегий их минимизации в финансовой деятельности организации;
- взаимодействие с внешними юридическими консультантами и адвокатами по вопросам, касающимся финансовых операций и учета;
- обучение сотрудников бухгалтерского отдела по вопросам юридического аспекта финансовой деятельности;
- подготовка ответов на запросы налоговых и финансовых органов, а также внутренних запросов руководства по юридическим вопросам;
- обеспечение соблюдения конфиденциальности и защиты прав и интересов организации во всех аспектах ее деятельности.

Обязанности бухгалтера-инспектора по кадрам:

- осуществление контроля за правильностью и своевременностью ведения бухгалтерского учета и составления финансовой отчетности;

- проведение проверок и анализа финансовых операций, выявление и устранение ошибок и несоответствий в учетных данных;
- организация и проведение инвентаризаций и инспекций финансовых активов и обязательств организации;
- контроль за соблюдением законодательства и внутренних нормативов при проведении финансовых операций;
- участие в разработке и внедрении системы внутреннего контроля и аудита для обеспечения надежности и эффективности финансовой деятельности организации;
- обеспечение соблюдения процедур и правил взаимодействия с контрагентами и клиентами при проведении финансовых операций;
- подготовка рекомендаций по улучшению процессов учета и контроля финансовых операций организации;
- участие в подготовке и представлении финансовой отчетности и налоговой отчетности в соответствии с требованиями законодательства;
- взаимодействие с внешними аудиторами и инспекторами по вопросам проведения проверок и аудита финансовой отчетности;
- обучение и консультирование сотрудников по вопросам бухгалтерского учета и внутреннего контроля финансовых операций.

Древовидная организационная структура ООО «Кварплата24» представлена на рисунке 1.



Рисунок 1 – Организационная структура ООО «Квартплата24»

ООО «Квартплата 24» предоставляет клиентам возможность эффективного управления жилищно-коммунальными платежами через свой онлайн-сервис. Организация предоставляет свои услуги как через веб-сайт, так и через мобильное приложение, обеспечивая удобство и доступность для клиентов в любое время и в любом месте. Основной вид деятельности ООО «Квартплата 24» заключается в предоставлении онлайн-сервиса для управления жилищно-коммунальными платежами. Организация обеспечивает клиентам возможность оплаты квитанций за коммунальные услуги, отслеживания задолженностей, внесения показаний счетчиков, хранения документов и получения уведомлений через личный кабинет или мобильное приложение. Дополнительные виды деятельности ООО «Квартплата 24» могут включать:

- консультационные услуги по вопросам управления жилищно-коммунальными платежами и оптимизации расходов на коммунальные услуги;

- разработка и внедрение дополнительных функциональных возможностей для улучшения пользовательского опыта на платформе;
- партнерство с управляющими компаниями и поставщиками коммунальных услуг для расширения спектра услуг, доступных через сервис;
- проведение обучающих мероприятий и вебинаров по вопросам эффективного управления коммунальными платежами и использования сервиса «Квартплата 24»;
- разработка и внедрение инновационных технологий в области управления жилищно-коммунальными платежами, таких как системы автоматизации учета и оплаты услуг.

Основным видом деятельности ООО «Квартплата 24» можно определить как "Предоставление онлайн-сервиса по управлению жилищно-коммунальными платежами и связанными с ними услугами". «Кроме того, организация оказывает дополнительные услуги, которых насчитывается 7 видов. ООО «Квартплата 24» действует на рынке свыше 10 лет, что говорит о ее значительном опыте и стабильности. За это время компания завоевала доверие клиентов и утвердилась как надежный и профессиональный партнер в области управления жилищно-коммунальными платежами» [16]. «Компания постоянно совершенствует свои сервисы и внимательно следит за рыночными изменениями, чтобы предоставлять своим клиентам наиболее актуальные и выгодные условия» [16].

## **1.2 Анализ бизнес-процессов**

Бизнес-процессы – это последовательность действий, выполняемых в организации для достижения ее целей. Они являются фундаментом функционирования любой компании и могут включать в себя такие процессы,

как производство товаров, обработка заказов, управление финансами, маркетинг и другие [1].

Анализ бизнес-процессов помогает выявить сильные и слабые стороны компании, а также определить пути повышения эффективности и снижения затрат.

«IDEF0 (Integration Definition for Function Modeling) — метод моделирования бизнес-процессов. Он используется для описания функций, действий и взаимосвязей внутри организации, а также для обозначения входов и выходов каждого процесса. Модели IDEF0 применяются для анализа и улучшения существующих процессов, а также для создания новых. Диаграммы IDEF0 используют прямоугольники, стрелки и другие символы для представления различных элементов процесса, таких как входы, выходы, элементы управления и механизмы» [26].

Контекстная диаграмма процесса перевода отражена на рисунке 2.

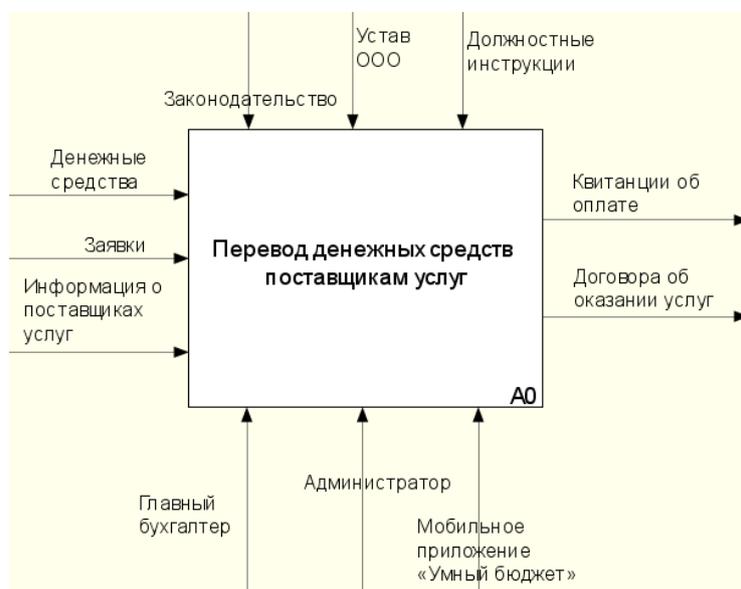


Рисунок 2 – Контекстная диаграмма работы над процессом перевода денежных средств поставщикам услуг

На рисунке 3 отражена IDEF0-диаграмма бизнес-процессов «как есть».

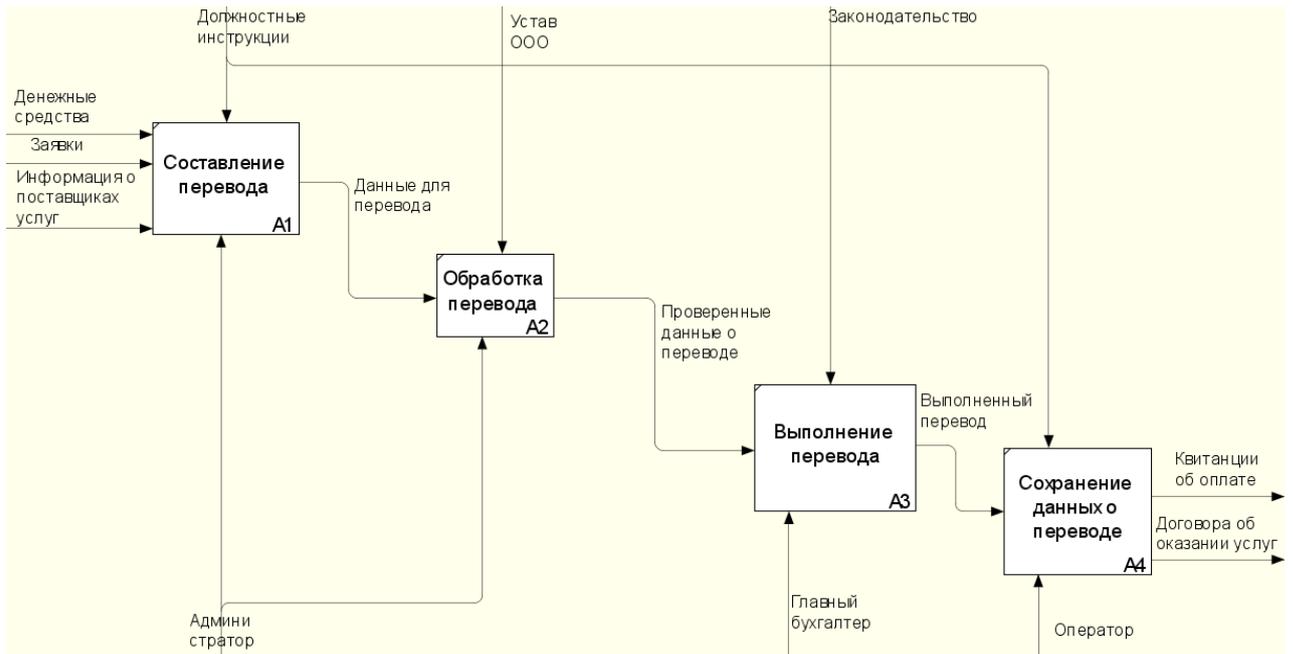


Рисунок 3 – Декомпозиция процесса перевода денежных средств поставщикам услуг (as-is)

Одной из основных проблем, выявленных в контексте разработки мобильного приложения для учета переводов денежных средств поставщикам услуг в компании «Квартплата 24», является отсутствие эффективной системы учета и управления финансовыми транзакциями внутри организации. Эта проблема порождает несколько негативных последствий [6]:

- недостаточная прозрачность и контроль за процессом перевода денежных средств поставщикам услуг, что может привести к затруднениям в отслеживании финансовых транзакций и контроле за производимыми платежами;
- не эффективное использование времени и ресурсов из-за необходимости ручного внесения данных о переводах. Это требует дополнительных усилий и времени со стороны сотрудников, что снижает их производительность и может привести к ошибкам;

- риск ошибок и потери данных в результате ручного внесения информации о переводах. Недостаточная автоматизация учета финансовых транзакций увеличивает вероятность ошибок и неточностей в финансовой отчетности [2].

Решение данной проблемы возможно через разработку мобильного приложения, которое позволит автоматизировать учет и управление переводами денежных средств внутри компании «Квартплата 24» [7]. Внедрение такого приложения улучшит прозрачность и контроль над финансовыми транзакциями, оптимизирует использование времени и ресурсов сотрудников, а также снизит риск ошибок и потери данных.

На рисунке 4 отражена диаграмма IDEF0 «как должно быть».

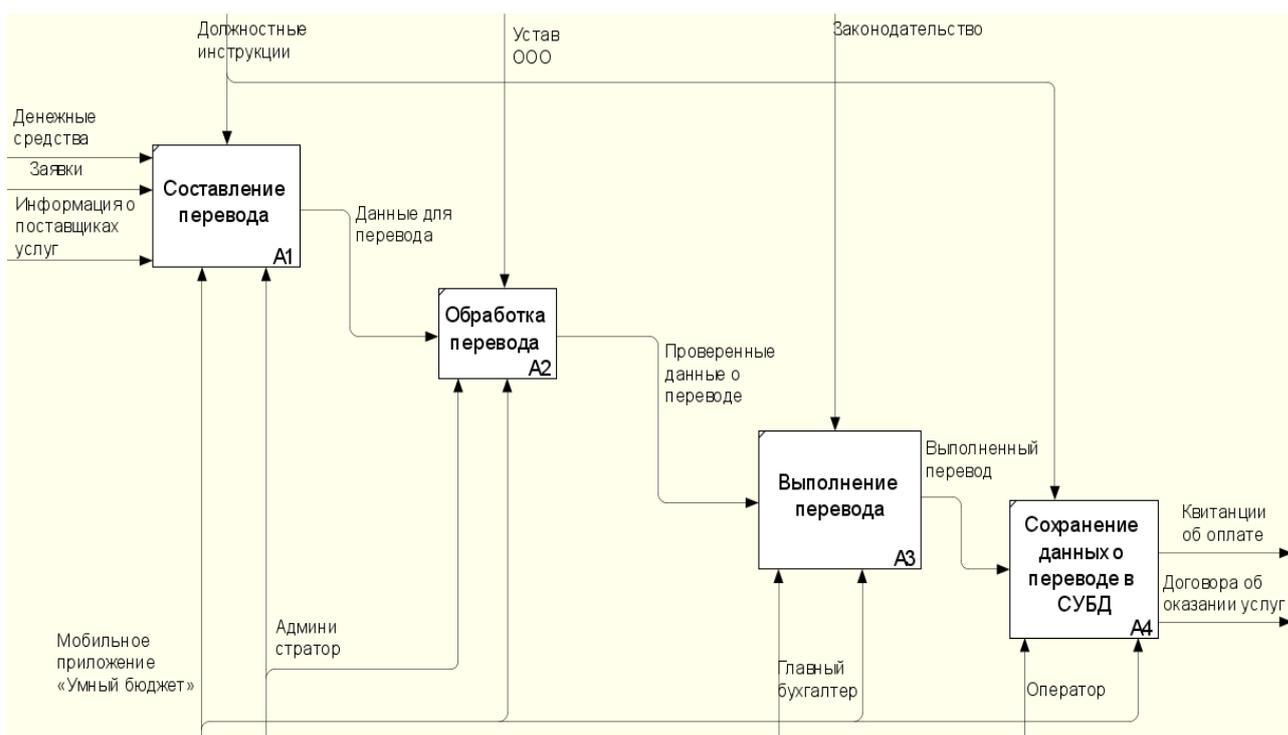


Рисунок 4 – Декомпозиция процесса перевода денежных средств поставщикам услуг (to-be)

Таким образом, «Умный бюджет» позволит сократить издержки, позволит сократить время работы над переводом.

### **1.3 Обзор существующих решений в области учета финансовых транзакций**

Необходимо рассмотреть наиболее популярные мобильные приложения для учета финансовых транзакций.

QuickBooks - это популярное программное обеспечение для управления финансами, разработанное компанией Intuit. Оно предоставляет широкий набор инструментов для ведения учета финансов бизнеса, включая учет платежей поставщикам услуг [12].

Основная цель приложения QuickBooks заключается в обеспечении удобного и эффективного способа управления финансами бизнеса [17]. Это программное обеспечение разработано для предоставления инструментов и функций, которые помогают предпринимателям и владельцам бизнеса вести учет своих финансов, контролировать расходы и доходы, управлять счетами, платежами поставщикам и многими другими аспектами финансовой деятельности [10].

QuickBooks предоставляет широкий спектр возможностей для учета финансов и управления бизнесом.

Некоторые из основных возможностей включают в себя:

- ведение бухгалтерского учета: QuickBooks позволяет пользователям вести учет доходов и расходов, создавать счета, управлять банковскими транзакциями и многое другое;
- управление счетами и счетами-фактурами: Пользователи могут создавать и отправлять счета-фактуры своим клиентам, отслеживать оплату и управлять задолженностями;
- отчетность и аналитика: QuickBooks генерирует различные финансовые отчеты, такие как баланс, отчет о прибылях и убытках, отчет о движении денежных средств и другие, что позволяет пользователям анализировать финансовое состояние своего бизнеса;

- налоговая поддержка: Приложение предоставляет инструменты для учета и подготовки налоговой отчетности, что упрощает процесс налогообложения для предпринимателей и владельцев бизнеса;
- управление инвентаризацией: QuickBooks позволяет отслеживать запасы, управлять инвентарными списаниями и вести учет товаров и услуг;
- доступ к мобильным приложениям: QuickBooks предлагает мобильные приложения для удобного управления бизнесом с любого устройства;
- интеграция с банками и платежными системами: Пользователи могут интегрировать свои банковские счета и платежные системы для автоматического импорта транзакций и упрощения процесса учета.

Эти возможности делают QuickBooks мощным инструментом для учета и управления финансами бизнеса, который может быть адаптирован под различные типы деятельности и потребности предпринимателей [8].

Дизайн данного приложения отражен на рисунке 5.

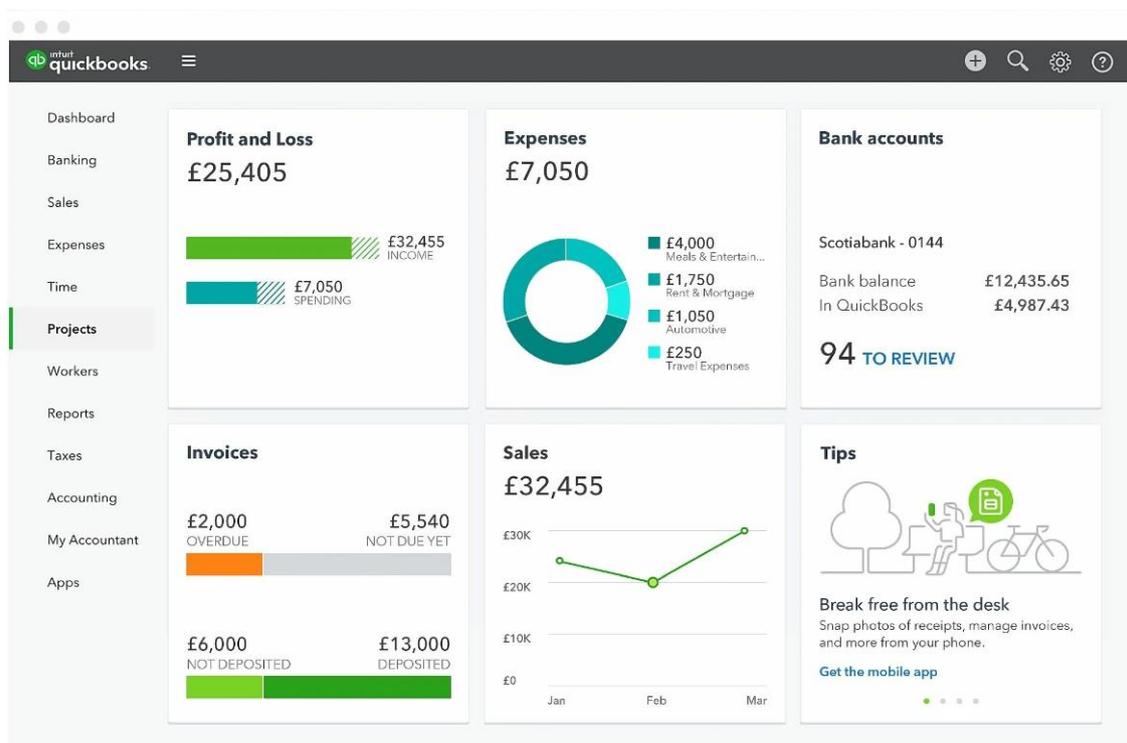


Рисунок 5 – Приложение QuickBooks

FreshBooks - Это приложение также предоставляет функционал для учета финансов и управления счетами-фактурами. С его помощью можно вести учет переводов денежных средств поставщикам услуг и управлять финансами своего бизнеса.

FreshBooks предлагает ряд ключевых функций для учета финансов и управления бизнесом, включая [10]:

- создание счетов-фактур: Позволяет быстро и легко создавать счета-фактуры для клиентов с помощью предустановленных шаблонов;
- отправка счетов-фактур: Позволяет отправлять счета-фактуры клиентам непосредственно из приложения по электронной почте или через другие каналы связи;
- отслеживание платежей: Позволяет отслеживать оплату счетов-фактур и уведомлять о просроченных платежах;
- управление расходами: Позволяет вводить и отслеживать расходы, в том числе расходы на услуги поставщиков;

- отчетность: Предоставляет широкий выбор отчетов, включая отчеты о финансовом состоянии бизнеса, отчеты о продажах, отчеты о расходах и другие;
- интеграция с банковскими счетами: Позволяет интегрировать банковские счета для автоматического импорта транзакций и учета финансовых операций;
- мобильное приложение: Обеспечивает доступ к функциям FreshBooks через мобильное приложение, что позволяет управлять бизнесом в любое время и из любой точки мира.

Эти ключевые функции делают FreshBooks удобным инструментом для учета финансов и управления бизнесом для предпринимателей и владельцев малого бизнеса.

Интерфейс отражен на рисунке 6 [9].

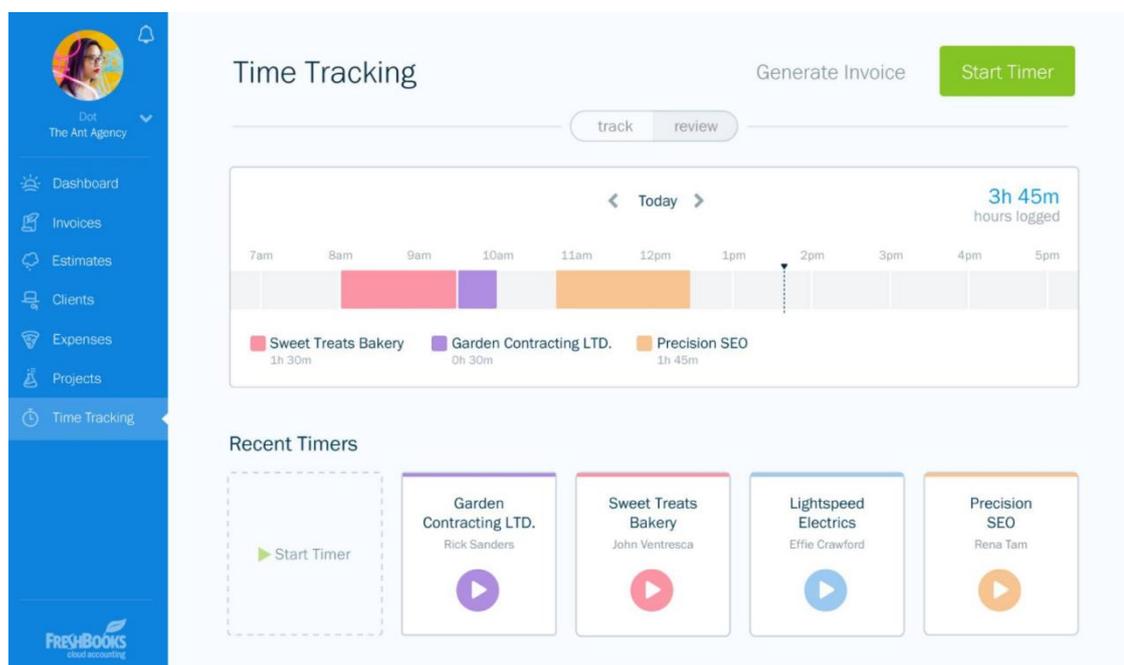


Рисунок 6– Приложение FreshBooks

Wave - предоставляет бесплатное приложение для учета финансов и управления бизнесом. Оно включает в себя функционал для создания счетов-фактур, отслеживания платежей и учета переводов денежных средств

поставщикам услуг. Мобильное приложение Wave предлагает ряд ключевых функций для управления финансами бизнеса[10]:

- учет финансов: Позволяет вести учет доходов и расходов, отслеживать счета, выставять счета клиентам и просматривать баланс;
- отчетность: Генерация различных финансовых отчетов, включая отчеты о доходах, расходах, прибыли и убытках, а также отчеты по налогам[13];
- управление счетами: Возможность добавлять и управлять банковскими счетами, кредитными картами и другими финансовыми аккаунтами;
- управление клиентами: Создание профилей клиентов, выставление счетов, отслеживание платежей и управление контактными данными клиентов;
- оплата счетов: Позволяет оплачивать счета онлайн и отслеживать статус оплаты;
- мобильный доступ: Доступ к функциям приложения через мобильные устройства, что позволяет вести учет финансов в любое время и в любом месте;
- интеграция: Возможность интеграции с другими бизнес-приложениями и сервисами для улучшения процессов управления бизнесом.

Мобильное приложение Wave предоставляет пользователям возможность интеграции с другими инструментами, улучшая их бизнес-процессы и повышая эффективность работы. Встроенные инструменты для учета финансов, отчетности и управления клиентами позволяют легко интегрировать Wave с другими сервисами и приложениями. Интерфейс приложения обеспечивает удобство и интуитивную навигацию, что делает работу с ним еще более удобной и эффективной. Интерфейс отражен на рисунке 7.

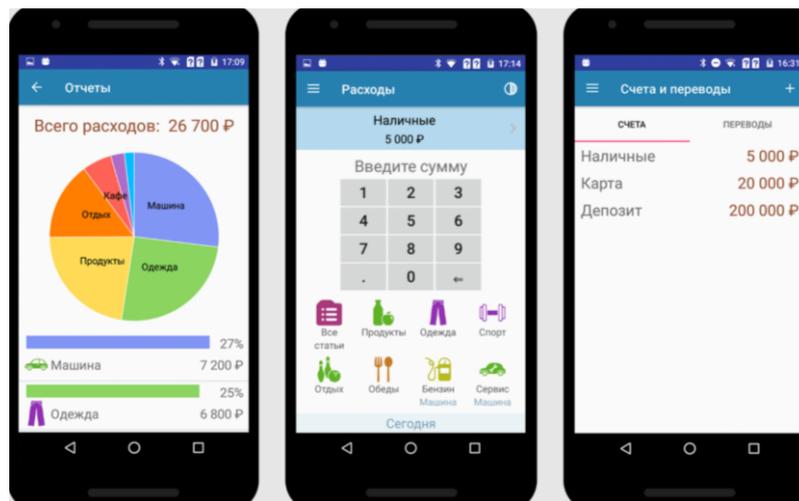


Рисунок 7 – Интерфейс Wave

Таким образом, после анализа нескольких популярных приложений для учета финансовых транзакций поставщикам услуг стало очевидным, что требуется разработка собственного приложения для учета и управления этими транзакциями. Основные причины для разработки собственного приложения включают следующие [10].

Адаптация к специфическим потребностям компании: создание собственного приложения позволит разработать инструмент, идеально соответствующий нуждам компании [14]. Опыт использования приложения поможет определить, какие функции необходимо добавить или улучшить, а также устранить неиспользуемые элементы. Улучшение эффективности и продуктивности: с помощью собственного приложения можно будет создавать персонализированные списки задач и управлять ими наиболее удобным для пользователей образом, что значительно повысит эффективность и продуктивность сотрудников.

Улучшенная безопасность данных: нативное приложение защитит данные компании от несанкционированного доступа и обеспечит их хранение на серверах, соответствующих высоким стандартам безопасности.

Экономия времени и ресурсов: Использование вашего собственного приложения упростит и ускорит процессы ведения бухгалтерского учета и управления финансовыми транзакциями [16].

Так же возможности для монетизации: Приложение может быть предложено широкому кругу пользователей, что дает дополнительные возможности для монетизации [19]. Сравнительный анализ приложений отражен в таблице 1.

Таблица 1 – Сравнительный анализ рассмотренных приложений

Критерии	QuickBooks	FreshBooks	Wave
Цена	Платные планы от \$15/месяц	Платные планы от \$15/месяц	Бесплатное приложение
Функциональность	Управление финансами, учет расходов и доходов, генерация отчетов	Управление счетами, создание и отправка счетов, отслеживание расходов, генерация отчетов	Управление финансами, учет расходов и доходов, генерация отчетов
Интерфейс и удобство	Интуитивный интерфейс, персонализация настроек	Простой и интуитивный интерфейс, настройка и персонализация	Простота использования, наличие дополнительных функций
Интеграции	Интеграция с банковскими счетами, CRM, платежными системами и т. д.	Интеграция с другими сервисами и программами	Интеграция с банковскими счетами, платежными системами
Мобильная поддержка	Доступны мобильные приложения для iOS и Android	Доступны мобильные приложения для iOS и Android	Доступны мобильные приложения для iOS и Android
Функциональность	Управление финансами, учет расходов и доходов, генерация отчетов	Управление счетами, создание и отправка счетов, отслеживание расходов, генерация отчетов	Управление финансами, учет расходов и доходов, генерация отчетов
Безопасность данных	Высокий уровень защиты данных и конфиденциальной информации	Защита данных и безопасность информации пользователей	Защита конфиденциальных данных и информации
Клиентская поддержка	Техническая поддержка для пользователей	Доступность технической поддержки	Техническая поддержка для пользователей

Разработка собственного приложения для учета денежных переводов поставщикам услуг может стать отличным инструментом для оптимизации рабочих процессов, повышения эффективности и повышения безопасности данных компании.

#### 1.4 Определение требований к системе

Определение требований к системе является одним из ключевых этапов разработки программного обеспечения. Этот этап включает в себя сбор, анализ и документирование требований, которые система должна удовлетворять для успешного выполнения поставленных задач.

В контексте разработки мобильного приложения для учета переводов денежных средств поставщикам услуг, определение требований включает в себя следующие аспекты. Требования к системе отражены в таблице 2.

Таблица 2 – Основные требования к системе

Требования	Статус	Полезность	Риск	Стабильность
Функциональные требования				
Совместимость с Apache Web Server 2.0 и выше	Одобренная	Высокая	Средний	Средняя
Непрерывное подключение к базе данных MySQL	Одобренная	Критичная	Средний	Средняя
Совместимость с Android версии 6.0 и выше, совместимость с iOS 8 и выше	Одобренная	Важная	Низкий	Высокая
Хранение выполненных задач в архиве	Предложенные	Средняя	Низкий	Средняя

Продолжение таблицы 2

Требования	Статус	Полезность	Риск	Стабильность
Поддержка разрешений экрана (qHD, HD, FHD, QHD, WXGA, WQXGA)	Одобренная	Высокая	Низкий	Высокая
Отчетность и аналитика выполненных переводов	Предложенные	Высокая	Средний	Средняя
Требования к надежности				
Аутентификация пользователя	Одобренная	Очень высокая	Низкий	Средняя
Автономное создание резервной копии базы данных	Предложенные	Критичная	Средний	Средняя
Требования к удобству использования				
Современный интуитивный Пользовательский интерфейс	Одобренная	Очень высокая	Средний	Очень высокая
Возможность совместной работы над проектом	Одобренная	Очень высокая	Высокий	Средняя
Требования к производительности				
Автономная функциональность	Одобренная	Высокая	Средний	Средняя
Время отклика приложения не должно превышать 5 секунд	Предложенные	Высокая	Средний	Средняя
Требования к поддержке				
Интеграция с другими инструментами	Предложенные	Средняя	Средний	Средняя
Максимальное время полного восстановления работы приложения – 24 часа	Предложенные	Высокая	Высокий	Низкая
Ограничения проектирования				
Язык программирования Java	Одобренная	Высокая	Низкий	Средняя

Ниже описаны общие требования для мобильного приложения «Умный Бюджет».

Интуитивно понятный пользовательский интерфейс: интерфейс приложения должен быть простым и понятным для пользователя, что обеспечит удобство использования и повысит уровень удовлетворенности пользователей.

Безопасное хранение данных пользователей: приложение должно обеспечивать надежное хранение и защиту конфиденциальных данных пользователей, чтобы предотвратить несанкционированный доступ к информации [20].

Непрерывная доступность к базе данных: приложение должно иметь непрерывный доступ к базе данных для быстрой загрузки данных и обеспечения плавной работы приложения.

Возможность синхронизации данных между устройствами: пользователи должны иметь возможность синхронизировать свои данные между различными устройствами, чтобы иметь доступ к ним в любое время и в любом месте.

Поддержка различных разрешений экрана [21]: приложение должно быть оптимизировано для различных разрешений экрана, чтобы обеспечить правильное отображение на различных устройствах.

Оптимизация производительности и быстродействия приложения: приложение должно работать быстро и эффективно, минимизируя задержки и подвисания, чтобы пользователи могли эффективно выполнять свои задачи.

Регулярные обновления и поддержка со стороны разработчиков: разработчики приложения должны регулярно выпускать обновления, исправлять ошибки и предоставлять поддержку пользователям для решения любых возникающих проблем.

Интеграция с другими сервисами и приложениями: приложение должно иметь возможность интеграции с другими сервисами и приложениями для улучшения функциональности и расширения возможностей.

Возможность резервного копирования данных: пользователи должны иметь возможность создавать резервные копии своих данных, чтобы предотвратить потерю информации в случае сбоев или ошибок.

Поддержка различных языков и региональных настроек: приложение должно поддерживать различные языки и региональные настройки, чтобы быть доступным для пользователей из разных стран и культур [16].

Соблюдение всех нормативных требований безопасности данных и законодательства о защите персональной информации: приложение должно соответствовать всем нормативным требованиям по защите данных пользователей и соблюдать законодательство о защите персональной информации.

### **1.5 Постановка задачи на разработку приложения**

Как упоминалось ранее, разработка мобильного приложения «Умный Бюджет» представляет собой актуальное решение для эффективного управления рабочими процессами в организации.

Далее следует рассмотреть ключевые модули, которые необходимо включить в приложение:

- модуль аутентификации и управления пользователями: Этот модуль позволит пользователям регистрироваться, входить в свои аккаунты, восстанавливать пароли и управлять своими профилями;
- модуль управления задачами и проектами: Здесь будут реализованы функции создания, редактирования, удаления и отслеживания задач и проектов. Пользователи смогут просматривать список своих задач, назначать сроки выполнения, присваивать приоритеты и отмечать выполненные задачи;

- модуль отчетности и аналитики: Здесь пользователи смогут создавать отчеты о выполненных задачах, анализировать свою продуктивность, а также получать статистику по проектам;
- модуль настройки и персонализации: Этот модуль позволит пользователям настраивать интерфейс приложения в соответствии с их предпочтениями, например, выбирать цветовые схемы, настраивать уведомления и прочее.

Постановка задач в отношении интерфейса приложения:

- разработка пользовательского интерфейса (UI): Создание привлекательного и интуитивно понятного интерфейса, который будет легко восприниматься пользователями. Это включает в себя разработку макетов экранов, выбор цветовой схемы, шрифтов, иконок и других элементов дизайна;
- адаптация интерфейса под различные устройства и разрешения экранов: Учитывая разнообразие устройств на рынке, включая смартфоны, планшеты и компьютеры, необходимо обеспечить адаптивность интерфейса для оптимального отображения на всех устройствах и разрешениях экранов;
- оптимизация пользовательского опыта (UX): Главная задача здесь - создать приятный и удобный пользовательский опыт, обеспечивая простоту навигации, интуитивные действия и минимальное количество шагов для выполнения задач;
- разработка анимаций и переходов: Добавление анимаций и переходов между экранами может улучшить восприятие интерфейса и сделать приложение более привлекательным для пользователей [21].

После разработки приложения требуется провести его тестирование. Этот процесс включает проверку на наличие ошибок, выявление проблем с пользовательским интерфейсом и оценку производительности.

После локального тестирования приложение должно стать доступным для пользователей, которым следует предоставить инструкции по его использованию [18].

На заключительном этапе важно отслеживать проблемы приложения на всех этапах его использования. При необходимости приложение должно обновляться и дополняться новыми функциями.

#### Выводы по главе 1

В ходе первой главы настоящей работы была представлена общая картина предметной области, начиная с описания организации ООО «Квартплата 24» и заканчивая постановкой задач на разработку приложения.

Анализ бизнес-процессов позволил выделить ключевые аспекты работы компании, а обзор существующих решений в области учета финансовых транзакций позволил определить текущие подходы и их преимущества [16].

Определение требований к системе является ключевым этапом, который позволяет уточнить функциональные и нефункциональные характеристики разрабатываемого приложения, обеспечивая его соответствие потребностям организации.

В результате этой главы были сформулированы основные задачи и цели разработки приложения, что является важным шагом к успешному выполнению проекта.

## **Глава 2 Проектирование и разработка мобильного «Умный Бюджет»**

### **2.1 Обоснование выбора средств разработки**

Существуют различные интегрированные среды разработки, в которых вы можете создавать приложения для разных платформ [22]. Чтобы объяснить, почему Android Studio была выбрана для разработки недорогих приложений, мы рассмотрим 3 наиболее распространенные среды: Microsoft Xamarin, Eclipse и Android Studio [28].

«Microsoft Xamarin - это платформа, предназначенная для создания кроссплатформенных мобильных приложений с использованием языка C#. Дек. Вы можете использовать общий код на C# для разработки приложений для Android, iOS и Windows. Xamarin также интегрируется с Visual Studio для Mac и Visual Studio Express, что делает его более полезным и продуктивным для разработчиков» [35].

«Eclipse - это свободно распространяемая интегрированная среда разработки, используемая для создания мобильных приложений на платформе Android. Eclipse предоставляет инструменты для разработки, отладки и тестирования Java-приложений. Эта среда популярна благодаря своей гибкости и широкой поддержке сообщества разработчиков» [31].

Android Studio: Эта интегрированная среда разработки (IDE) является официальным инструментом для создания мобильных приложений на платформе Android. Android Studio предлагает разработчикам широкий спектр возможностей, включая редактор дизайна, поддержку языка программирования Kotlin, встроенный генератор кода.

Эти среды разработки предлагают различные возможности и подходы к созданию мобильных приложений, и выбор оптимальной среды зависит от требований проекта, опыта разработчиков и других факторов.

Поэтому среда разработки Android Studio была выбрана для разработки интеллектуальных бюджетных приложений. Этот выбор обусловлен

максимальной поддержкой разработки мобильных приложений на платформе Android, наличием мощного набора инструментов и функций, а также обширной поддержкой Google.

Язык программирования Java был выбран для реализации приложения, потому что он обеспечивает высокую скорость разработки, повышенную безопасность и четкий синтаксис.

Для разработки приложений на платформе Android можно использовать несколько языков, таких как Java и Kotlin.

Наглядное сравнение этих языков программирования представлено в таблице 3.

Таблица 3 – Сравнение языков Kotlin и Java

Критерий	Java	Kotlin
Сложность изучения	Простой	Более сложный
Экосистема	Широкая, множество библиотек и инструментов	Менее развитая, но активно растущая
Поддержка	Долгая и стабильная история, большое сообщество разработчиков	Относительно новый язык, поддерживается Google
Производитель	Высокая	Хорошая
Безопасность	Отсутствует нулевой указатель	Поддержка нулевого указателя, снижает вероятность ошибок
Совместимость	Более широкая, множество фреймворков и библиотек	Некоторые фреймворки и библиотеки могут быть менее совместимы
Поддержка Android	Имеет множество инструментов и ресурсов для разработки Android-приложений	Поддерживается Google, активно используется для разработки Android-приложений

Итак, по сравнению с Kotlin, язык Java имеет преимущества в области сложности изучения, богатой экосистемы, долгой и стабильной истории

поддержки, производительности и широкой совместимости с различными фреймворками и библиотеками.

Однако Kotlin обладает своими преимуществами, такими как безопасность и поддержка нулевого указателя, что делает его привлекательным выбором для некоторых проектов.

Для разработки приложения «Умный Бюджет» был выбран язык программирования Java. В качестве архитектурного решения используется клиент-серверная модель, которая распределяет функциональность приложения между клиентской и серверной частями.

Клиент выступает в роли программы или устройства, запрашивающего услуги или ресурсы от сервера, тогда как сервер предоставляет эти услуги или ресурсы клиенту.

Серверная часть будет работать на процессоре AMD Ryzen 5 5600H и оснащена 16 ГБ оперативной памяти, что обеспечивает достаточную вычислительную мощность для будущего роста аудитории приложения[29]. Операционная система, используемая на сервере, — Windows 10 Pro 22H2.

Хранение данных осуществляется в реляционной базе данных MySQL 8.4.0 LTS, что обеспечивает надежное и эффективное управление данными о финансовых транзакциях.

## **2.2 Проектирование структуры приложения**

### **2.2.1 Концептуальная модель**

Необходимо разработать концептуальную схему приложения, представленную на рисунке 8 [23].

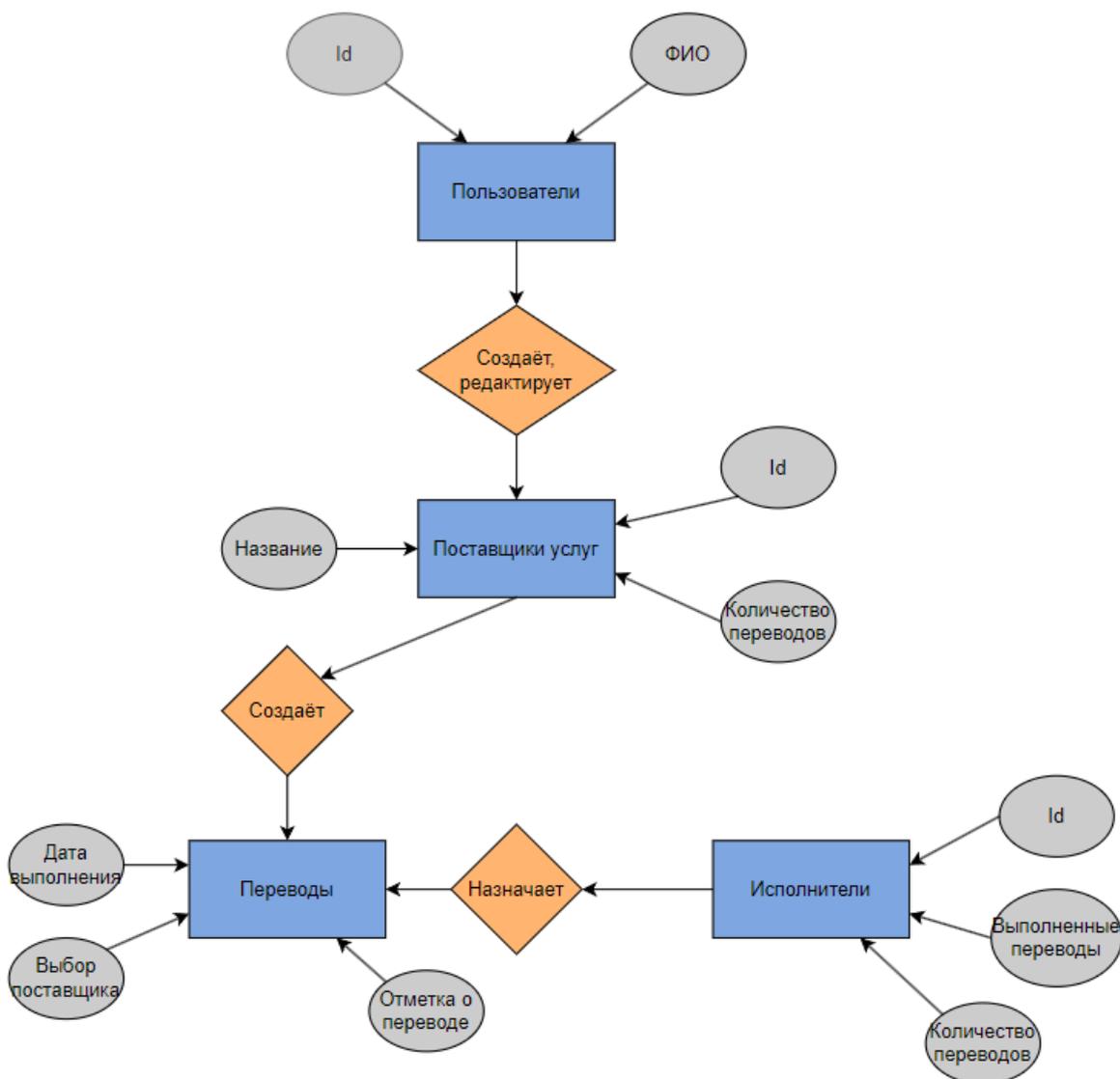


Рисунок 8 – Концептуальная модель

Концептуальная модель приложения «Умный Бюджет» отображает ключевые сущности и их взаимосвязи [24]. В нее входят следующие.

Пользователи: сотрудники и руководство, использующие приложение для финансовых переводов.

Переводы: основная сущность, содержащая информацию о сумме, сроках и статусе перевода.

Категории: организационная структура для классификации переводов с названиями и описаниями.

Исполнители: сотрудники, которые несут ответственность за

выполнение задач.

Связи между сущностями определяют, что пользователи могут создавать, просматривать и редактировать переводы, а также назначать исполнителей. Переводы могут быть привязаны к определенным категориям, и каждый перевод может иметь свой уникальный статус, такой как «в ожидании», «в работе» или «завершен».

Эта модель служит основой для разработки функциональных требований и дальнейшего проектирования приложения.

### **2.2.2 Диаграмма прецедентов и вариантов использования**

Ключевые варианты использования для приложения «Умный бюджет»:

- регистрация нового пользователя: операторы, администраторы и главный бухгалтер могут зарегистрировать новый аккаунт в приложении, предоставляя необходимую информацию;
- авторизация в системе: после регистрации пользователи могут выполнять вход в систему, используя свои учетные данные (логин и пароль);
- управление бюджетом: главный бухгалтер может создавать, просматривать, редактировать и удалять бюджеты для различных подразделений или проектов компании. Операторы могут добавлять информацию о финансовых операциях, таких как доходы и расходы, в соответствующие бюджеты;
- мониторинг финансовых данных: администраторы и главный бухгалтер могут просматривать отчеты и статистику о финансовых операциях компании, включая доходы, расходы, прибыль и т.д.;
- управление пользователями: администратор может назначать роли и уровни доступа для других пользователей, управлять их учетными записями и сбросить пароли при необходимости;

- настройки приложения: все пользователи могут настраивать различные параметры приложения, такие как язык интерфейса, формат отображения данных, уведомления и т.д.;
- уведомления о финансовых событиях: пользователи могут получать уведомления о важных финансовых событиях, таких как превышение бюджета, неоплаченные счета и т.д.;
- анализ финансовой деятельности: главный бухгалтер может анализировать финансовую деятельность компании, используя статистические данные и отчеты, предоставляемые приложением.

Пример диаграммы использования отражен на рисунке 9.



Рисунок 9 – Процесс формирования перевода

Диаграмма вариантов использования приложения «Умный Бюджет» отображает основные субъекты, события и функции(действие) между ними. Она включает следующие субъекты:

- администратор отвечает за управление системой и контроль ее работы;
- оператор осуществляет оперативную работу с данными и взаимодействует с системой для выполнения конкретных задач.

Связи между функциями отображают последовательность выполнения определенных действий.

Функции, представленные на диаграмме, включают:

- поиск поставщика в базе данных: оператор инициирует процесс поиска поставщика в базе данных для осуществления дальнейших действий по переводу;
- запрос информации о переводах: оператор запрашивает информацию о предыдущих переводах от системы для оценки их статуса и истории;
- предоставление перевода: после поиска и запроса информации, оператор предоставляет перевод для выполнения.

События, представленные на диаграмме, включают:

- возникает потребность в переводе: система обнаруживает необходимость осуществления перевода денежных средств поставщикам услуг;
- поставщик найден: система находит соответствующего поставщика в базе данных;
- администраторы получили запрос о переводе: администраторы получают уведомление о запросе на перевод, который должен быть обработан.

Переводы могут иметь различные состояния: «в ожидании», «в работе», «завершена».

Эта диаграмма помогает визуализировать процесс взаимодействия между различными элементами системы и является полезным инструментом для понимания функциональности и потоков данных в приложении «Умный Бюджет».

### **2.3 Логическое и физическое моделирование**

Далее представлено логическое моделирование приложения.

Класс «Платёж» представляет собой конкретный перевод в рамках приложения «Умный бюджет». У каждого перевода имеются следующие свойства: уникальный идентификатор (id), название (ID потребителя), сумма денежных средств (Сумма), срок выполнения перевода (Дата платежа) и текущий статус (Отметка о переводе), который может быть «В ожидании», «В процессе» или «Завершено».

Для сбора и группировки информации о потребителе услуг служит класс «Потребитель услуг». У каждого объекта данного класса имеется уникальный идентификатор (ID потребителя), так же уникальный идентификатор поставщика услуг (ID поставщика услуг), ФИО потребителя (ФИО) и о месте перевода (Номер лицевого счёта) [30].

Класс «Поставщик услуг» представляет поставщика который предоставляет некие услуги или товар. Каждый поставщик имеет уникальный идентификатор (ID поставщика услуг), название (Наименование), название банка, работающего с поставщиком (Наименование банка), уникальный банковский номер (БИК банка) и счёт на который будут переведены денежные средства (Расчётный счёт).

Класс «Перевод денежных средств» представляет полную информацию о переводе денежных средств. Это включает в себя уникальный идентификатор перевода денежных средств (ID перевода), уникальный идентификатор поставщика (ID поставщика услуг), сумму перевода денежных средств (Сумма перевода) и фиксация времени в которое поступил перевод (Дата перевода).

Логическая модель данных отражена на рисунке 10.

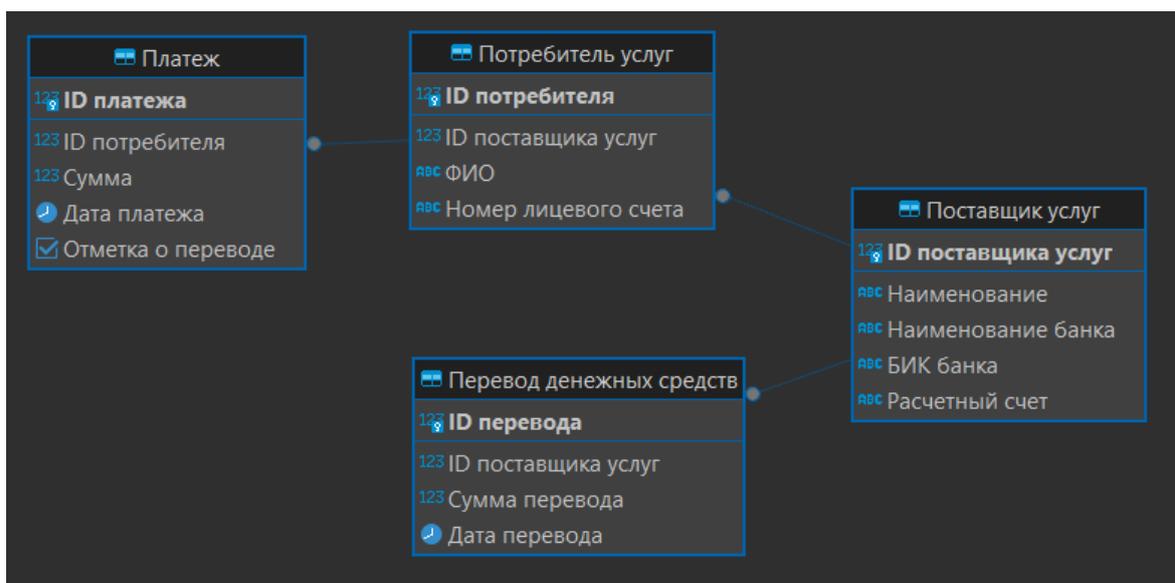


Рисунок 10 – Логическая модель данных приложения

Физическое моделирование приложения описывает структуру системы, включая ее компоненты.

Для серверной части планируется применение Spring Boot, обеспечивающего обработку серверной логики и создание API для обмена данными с клиентом. На клиентской стороне, разработанной на Java в Android Studio, предполагается использование фреймворка Android SDK для разработки пользовательского интерфейса и логики приложения. Обмен данными между сервером и клиентом будет осуществляться в формате XML для обеспечения эффективного взаимодействия [26].

Важным аспектом проектирования будет обработка ошибок и предусмотрение различных кодов статуса HTTP для корректной обработки разнообразных ситуаций. На клиентской стороне также будет использоваться Android SDK для обработки и отображения полученных данных.

Взаимодействие клиента с сервером отражена на рисунке 11.

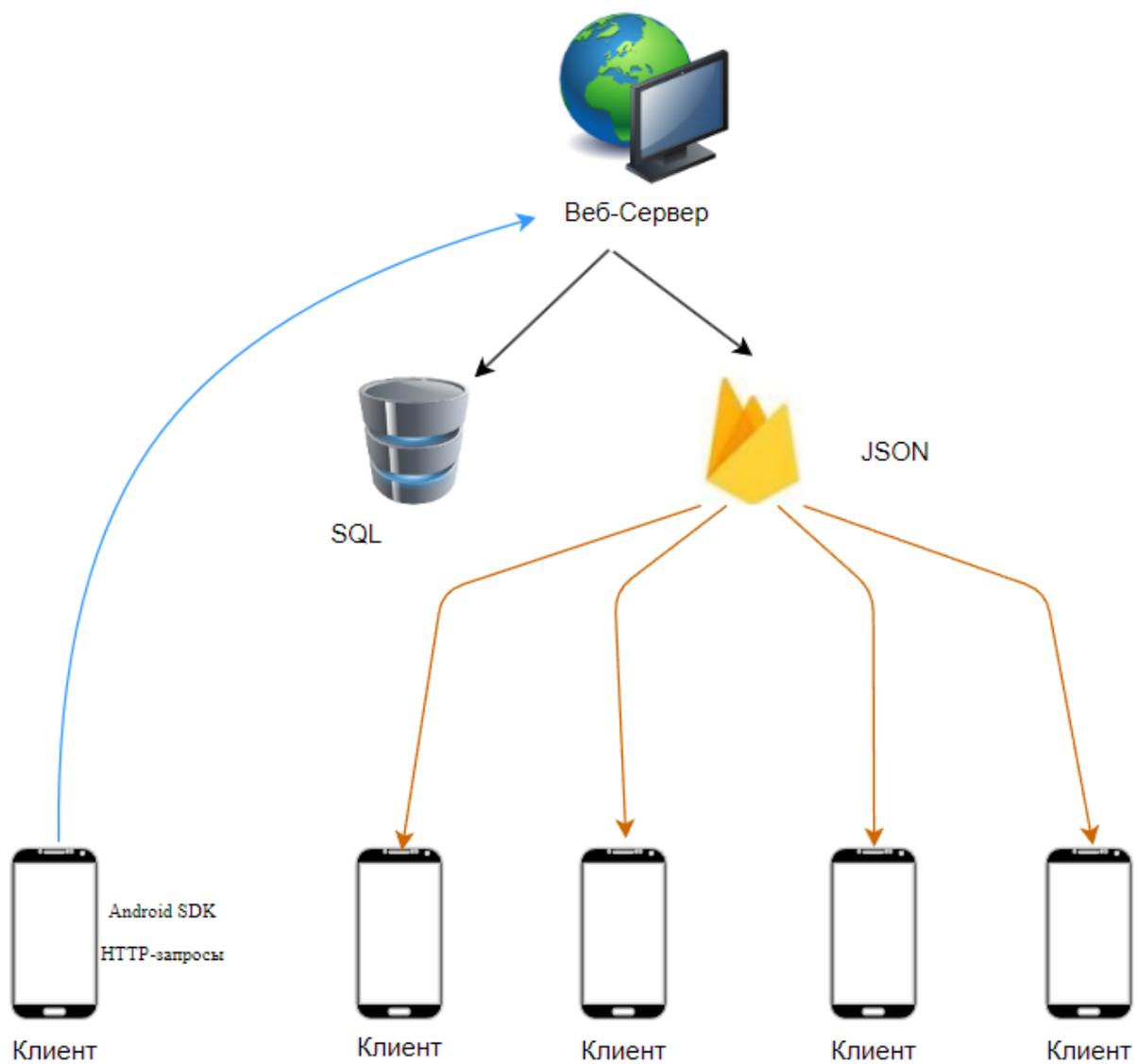


Рисунок 11 – Взаимодействие клиента с сервером

Физическая модель базы данных отражена на рисунке 12.

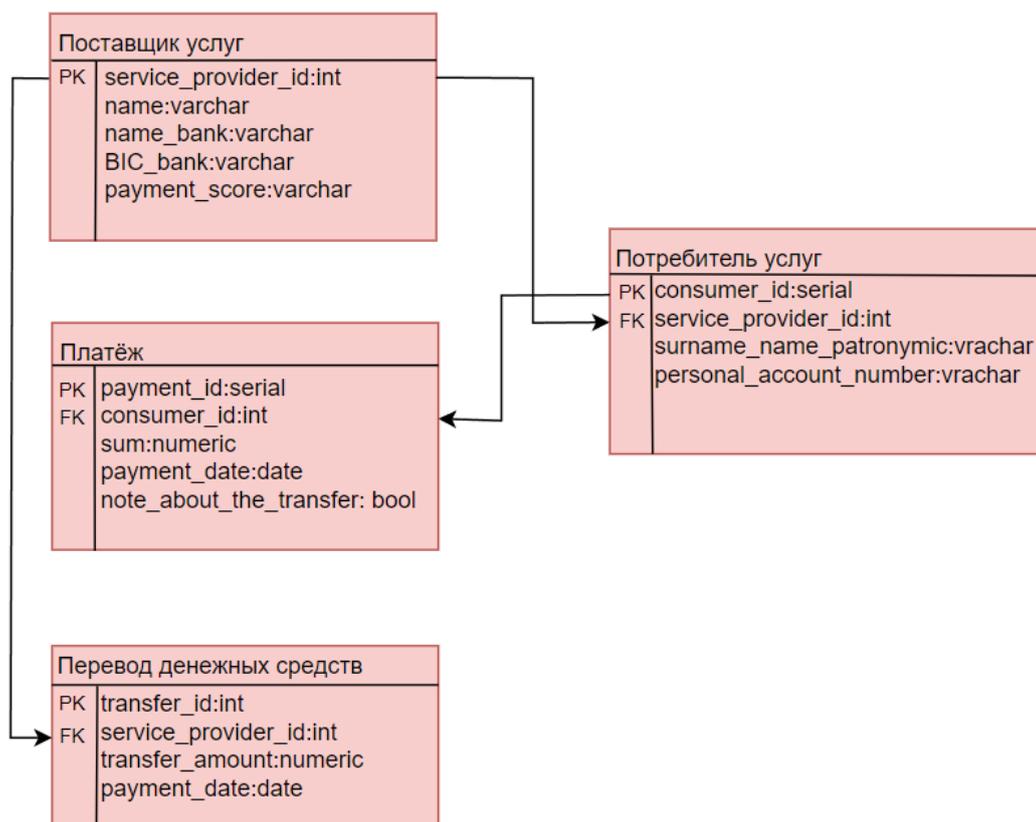


Рисунок 12 – Физическая модель базы данных

Таблица Платёж служит хранилищем данных о финансовых операциях. Каждая запись в таблице имеет уникальный идентификатор платежа (ID), который является основным ключом. Сумма платежа хранится в числовом формате (NUMERIC) с точностью до двух знаков после запятой. Для хранения даты платежа используется поле Дата платежа типа DATE. Каждый платеж также связан с определенной категорией через поле ID потребителя.

Таблица Потребитель услуг предназначена для хранения данных о клиентах. Каждая запись в таблице имеет уникальный идентификатор потребителя (ID), который служит главным ключом. Фамилия, имя и отчество клиента хранятся в текстовом поле ФИО типа VARCHAR. Каждый клиент также связан с определенной категорией через поле ID поставщика услуг.

Таблица Поставщик услуг содержит информацию о поставщиках услуг. Каждая запись имеет уникальный идентификатор поставщика (ID), который

является основным ключом. Для каждого поставщика хранится наименование, название банка, БИК банка и расчетный счет.

Таблица Перевода денежных средств используется для хранения данных о финансовых переводах. Каждая запись имеет уникальный идентификатор перевода (ID). Сумма перевода хранится в числовом формате с точностью до двух знаков после запятой. Для хранения даты перевода используется поле Дата перевода типа DATE. Каждый перевод также связан с определенной категорией через поле ID поставщика услуг.

## **2.4 Разработка интерфейсов и архитектурной модели приложения**

В приложении будут следующие модели:

- экран входа/регистрации,
- экран списка поставщиков услуг,
- экран деталей платежа,
- настройки приложения.

Каждая из этих моделей будет иметь свои свойства и методы.

Например, для модели детали платежей определены следующие свойства:

- ID (Идентификатор),
- номер Л/С (Лицевой счет),
- сумма,
- дата платежа,
- поставщик услуг,
- отметка о переводе,
- комментарий,
- дата создания.

Для отображения информации пользователю будут реализованы следующие представления:

- главный экран,
- экран создания/редактирования платежа,
- экран статистики и аналитики,
- экран списка платежей.

Каждый из этих интерфейсов содержит элементы пользовательского интерфейса (кнопки, текстовые поля и т. Д.).

Чтобы ускорить разработку и повысить производительность приложений, вы можете использовать сторонние библиотеки, такие как:

- Android Jetpack: Это набор компонентов и инструментов от Google, предназначенных для облегчения разработки приложений на платформе Android;
- Retrofit: Это библиотека для работы с сетью в приложении Android;
- Glide или Picasso: Это библиотеки для загрузки и отображения изображений в приложении Android;
- Dagger или Koin: Это инструменты для внедрения зависимостей в приложение;
- Coroutines или RxJava: Это библиотеки для асинхронного программирования в приложениях Android [4];
- Firebase: Это платформа для разработки мобильных и веб-приложений от Google.

Для создания начального экрана приложения Smart Budget в Android Studio будут использованы фрагменты [5].

Главный экран будет включать следующие фрагменты:

- для отображения списка переводов;
- для изменения имеющегося перевода;
- для редактирования информации о переводе;
- для управления справочником «Поставщик услуг»;
- для управления справочником «Потребитель услуг»;

- для ввода платежей;
- для формирования нового перевода.

Для навигации между фрагментами и создания единой структуры приложения в Android Studio будет использоваться компонент навигации [3].

#### Выводы по главе 2

Во второй главе данной работы была проведена всесторонняя разработка и анализ архитектуры мобильного приложения «Умный бюджет». Рассмотрены различные аспекты проектирования и реализации, включая выбор технологий, создание концептуальной модели, а также разработку ключевых модулей и компонентов. На основе проведенного исследования были выделены основные требования к функциональности приложения, которые включают адаптацию к специфическим потребностям компании, улучшение эффективности работы персонала, повышение безопасности данных и возможность масштабирования. Проведен выбор инструментов и технологий для разработки, таких как Android Studio, Java и MySQL, что обеспечило соответствие требованиям и потребностям проекта. Кроме того, были разработаны и протестированы модули приложения, включая модуль регистрации и аутентификации, модуль хранения данных, и модуль управления переводами. Реализация дизайна приложения с использованием принципов пользовательского интерфейса обеспечила удобство и интуитивную понятность интерфейса для конечных пользователей. Результаты работы показали, что разработанное приложение «Умный бюджет» отвечает заявленным требованиям и готово к дальнейшему этапу тестирования и внедрения. Выполненные задачи и достигнутые результаты подтверждают правильность выбранных подходов и технологий, что является основой для успешного завершения проекта.

## **Глава 3 Разработка дизайна и тестирование приложения**

### **3.1 Реализация дизайна приложения**

При разработке дизайна мобильного приложения «Умный бюджет» были использованы инструменты и методики для обеспечения его удобства и функциональности.

Было проведено исследование рынка и анализ дизайна других финансовых приложений, что позволило определить ключевые элементы, которые следует включить в дизайн и организовать на экране [25].

Мы использовали специальные инструменты дизайна интерфейса, такие как Figma и Sketch, для создания макета страницы приложения [27].

В этом процессе использовались готовые компоненты и элементы дизайна, были разработаны уникальные элементы, отражающие концепцию приложения. Это обеспечило идеальное сочетание эстетики и функциональности [11].

В результате получается высококачественный макет страницы, который полностью соответствует концепции и требованиям проекта [15].

На рисунках 13-16 отражен разработанный интерфейс мобильного приложения.

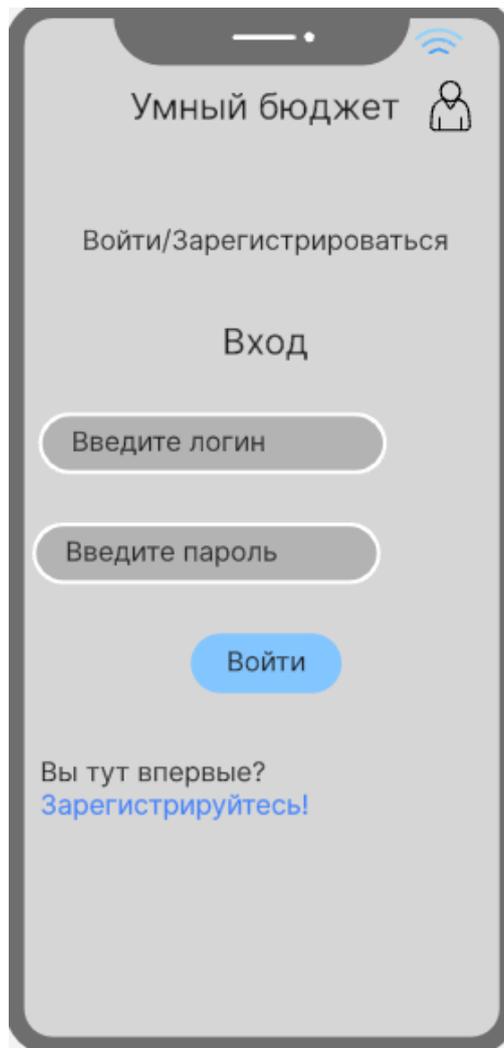


Рисунок 13 – Страница авторизации

На странице авторизации, представленной на рисунке 14, пользователи должны ввести логин и пароль, после чего данные будут сверены с базой данных, если такой пользователь есть в системе, то произойдет аутентификация и пользователь войдет в систему.

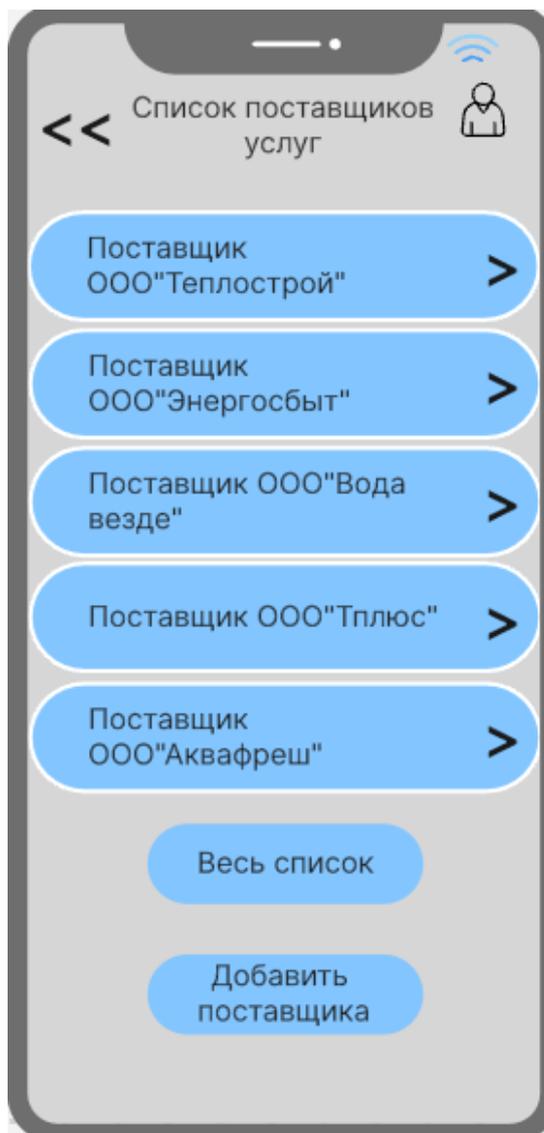


Рисунок 14 – Список поставщиков услуг

На данном экране отражен список поставщиков услуг.

На рисунке 15 изображен пример отображения информации о конкретном поставщике услуг.

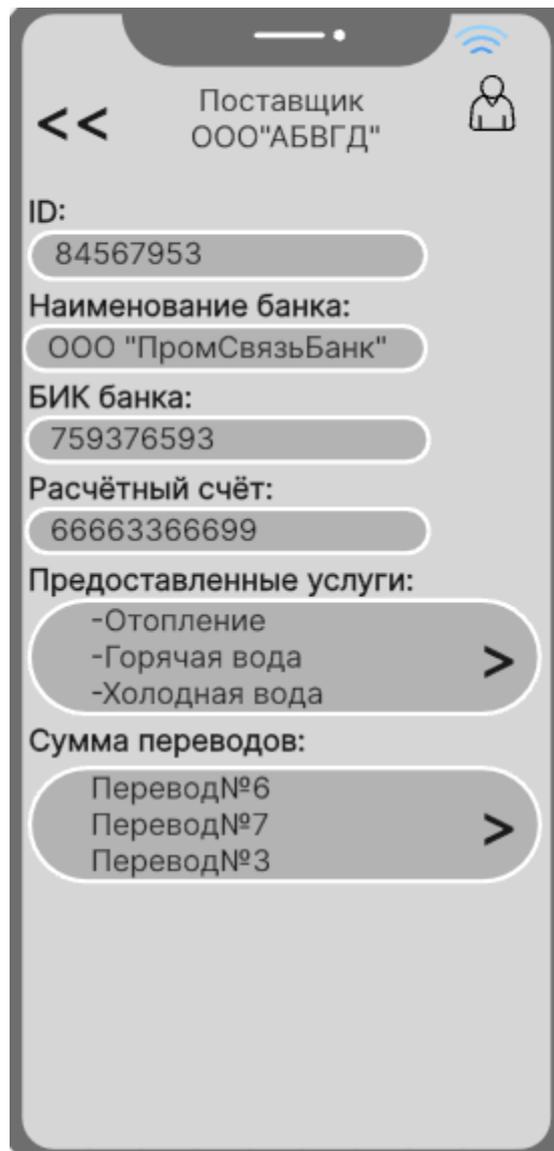


Рисунок 15 – Информация по конкретному поставщику

Расчет оплаты по счетам исходя из имеющейся суммы можно произвести на следующем экране, показанном на рисунке 16.

Например, компания располагает некоторой суммой на данный момент времени для погашения задолженностей по платежам поставщикам услуг. Программа предложит перечень счетов для оплаты в рамках указанной суммы.

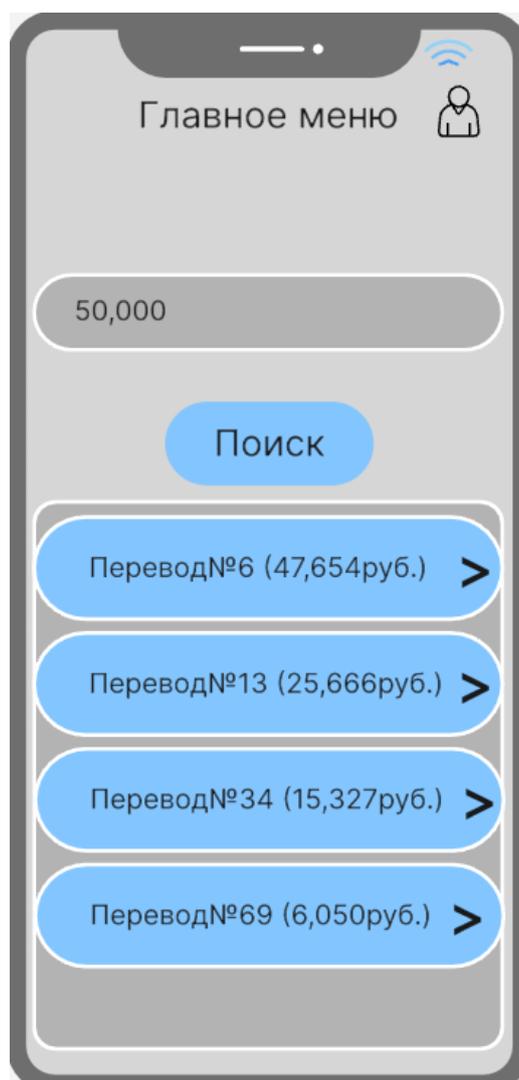


Рисунок 16 – Расчет оплаты по счетам исходя из имеющейся суммы

Таким образом, на рисунках была представлена форма регистрации и авторизации, список поставщиков услуг, расчет оплаты по счетам исходя из имеющейся суммы, а также дизайн страницы с конкретным поставщиком.

### **3.2 Реализация функциональных модулей**

Приложение «Умный бюджет» включает следующие модули:

- модуль регистрации и авторизации – отвечает за регистрацию пользователей и их идентификацию в системе [33];

- модуль хранения данных (Data storage) - будет заниматься сохранением информации о переводах денежных средств и других связанных с ними данными;
- модуль управления переводами (Transaction management) - будет отвечать за функции добавления, удаления, изменения и отображения информации о переводах денежных средств поставщикам услуг;
- модуль статистики (Statistics) - будет проводить анализ информации о переводах денежных средств и создавать статистические отчеты, помогающие пользователям оценить свои финансовые потоки и принимать обоснованные решения;
- модуль настроек (Settings) - предоставит возможность настройки параметров приложения, таких как язык интерфейса, формат отображения сумм переводов, уведомлений и других параметров, с учетом индивидуальных предпочтений пользователей [34].

Необходимо рассмотреть модуль регистрации и аутентификации пользователей. В этом модуле используется класс `User`, представляющий запись о пользователе в базе данных.

Для определения роли пользователя в таблице добавлено поле `role`, указывающее на роль пользователя: значение 0 означает, что пользователь является обычным пользователем, а значение 1 обозначает администратора. Метод `getUserByUsername` возвращает объект `User` вместе со значением поля `role`.

Код модуля регистрации и аутентификации пользователей отражен в приложении А.

Рассмотрим далее модуль хранения данных. Этот модуль содержит два основных метода: `deleteRecord`, предназначенный для удаления записи из базы данных по указанному идентификатору, и `updateRecord`, который обновляет сведения о записи с указанным идентификатором [32]. Оба метода

взаимодействуют с базой данных с помощью соответствующих методов `delete` и `update` объекта `SQLiteDatabase`.

В блоке `companion object` определены константы, которые указывают имя базы данных, её версию, имя таблицы и названия столбцов. Это упрощает доступ к значениям внутри класса, и если нужно изменить имя таблицы или столбца, можно сделать это, изменив только соответствующую константу.

Следующий модуль — модуль управления переводами. В этом модуле добавлены два дополнительных объекта базы данных: `SupplierDatabase` и `TransactionDatabase` – для управления информацией о поставщиках услуг и переводах соответственно. Метод `addTransaction` теперь принимает также идентификатор поставщика услуг и тип перевода, создавая объект перевода с указанным поставщиком. Методы `getAllTransactions` и `addTransaction` теперь также обновляют информацию о переводе с указанным поставщиком. Добавлены методы для управления поставщиками услуг и переводами: `addSupplier`, `getAllSuppliers`, `deleteSupplier`, `updateSupplier`, `addTransaction`, `getAllTransaction`, `deleteTransaction` и `updateTransaction`. Методы для добавления поставщиков услуг и переводов принимают не только их названия, но и другие параметры, такие как контактная информация для поставщика. Метод `addTransaction` также принимает идентификатор типа транзакции и поставщика услуг, связывая их с транзакцией. Код модуля представлен в приложении Б.

Метод `getAllTransactions` использует объекты `SupplierDatabase` и `TransactionDatabase` для получения соответствующей информации о поставщиках услуг для каждого перевода, затем добавляет эту информацию к транзакции.

В методе сбора статистики реализован класс `StatisticsManager`, предназначенный для анализа транзакций пользователя и формирования статистических отчетов. Основные методы этого класса включают:

- `getCountOfCompletedTransactions` и `getCountOfIncompleteTransactions`, которые возвращают

количество завершенных и незавершенных транзакций за определенный период времени соответственно;

- `getCountOfTransactionsBySupplier` и `getCountOfTransactionsByType`, которые предоставляют количество транзакций, связанных с конкретным поставщиком или типом транзакции;
- `getAverageTransactionAmount`, которая рассчитывает среднюю сумму транзакции за определенный период времени. Эти методы предназначены для создания различных статистических отчетов, которые помогают пользователям планировать свои финансы и оценивать финансовую активность.

Класс `SettingsManager` разработан в модуле настроек, который отвечает за управление настройками приложения. Основными характеристиками этого класса являются:

- `getLanguage`, `getColorScheme` и `getTimeFormat` предоставляют текущие параметры для языка интерфейса, цветовой схемы и формата времени соответственно;
- установлены новые значения для языка интерфейса, цветовой схемы и формата времени соответственно в `setLanguage`, `setColorScheme` и `setTimeFormat`;
- `isNotificationsEnabled` и `setNotificationsEnabled` соответственно возвращают и устанавливают текущее состояние уведомления. Эти функции позволяют настраивать различные параметры приложения, включая язык интерфейса, цветовую схему, формат времени и уведомления.

### 3.3 Тестирование приложения

Тестирование играет ключевую роль в качественной разработке мобильного приложения «Умный бюджет». Для этого приложения было использовано несколько методов тестирования. Модульное тестирование - такой подход позволяет независимо протестировать работу отдельных модулей приложения. Мы тщательно протестировали и отладили каждый модуль в Android Studio, чтобы выявить и исправить возможные ошибки, изолировав их от остальных компонентов. Функциональные тесты: Для проверки функционирования приложения в целом и взаимодействия его различных функций и модулей мы используем функциональные тесты.

Тестирование пользовательского интерфейса (UI) представляет собой вид тестирования, предназначенный для оценки того, насколько удобен и понятен интерфейс приложения для пользователей. Было проведено тестирование с участием реальных пользователей, чтобы определить, насколько просто им удастся выполнять различные задачи и выявить возможные проблемы. Результаты этих тестов представлены в таблице 4.

Таблица 4 – Тестирование приложения

Номер теста	Описание теста	Ожидаемый результат	Фактический результат	Статус
1	Добавление нового перевода	При добавлении нового перевода в приложении должна появляться новая запись в списке переводов	После добавления нового перевода в список переводов была добавлена новая запись	Пройден
2	Удаление существующего перевода	При удалении перевода из приложения должна исчезать соответствующая запись в списке переводов	После удаления перевода из списка переводов соответствующая запись была удалена	Пройден

Продолжение таблицы 4

Номер теста	Описание теста	Ожидаемый результат	Фактический результат	Статус
3	Скорость работы приложения	Приложение должно быстро реагировать на пользовательские действия и не тормозить при работе с большим количеством задач	Приложение быстро реагирует на пользовательские действия и работает стабильно даже при большом количестве задач	Пройден

В таблице 5 отражены устройства на которых проходили тесты и версии операционных систем.

Таблица 5 – Устройства, на которых тестировалось приложение

Устройство	Версия операционной системы
Samsung Galaxy S10	Android 11
Google Pixel 4	Android 10
OnePlus 8	Android 11
Xiaomi Mi 10	Android 12
Samsung Galaxy Note 20	Android 11

После проведения модульного тестирования на различных устройствах и версиях операционных систем, а также в различных условиях использования, были обнаружены и устранены ряд ошибок и недочетов, а также улучшена функциональность, надежность и производительность приложения.

В результате проведенного тестирования можно заключить, что приложение функционирует корректно и предлагает пользователям удобный и надежный инструмент.

### 3.4 Оценка эффективности проекта

Один из основных факторов, который оказывает влияние на успех проекта, заключается в его воздействии на бизнес-процессы предприятия. Внедрение приложения «Умный бюджет» планируется с целью улучшения процессов управления переводами и контроля их выполнения, что должно привести к повышению эффективности работы персонала и снижению количества ошибок. Это, в свою очередь, способствует повышению уровня удовлетворенности поставщиков услуг. Экономический аспект также имеет важное значение в оценке проекта. Применение открытых технологий и готовых решений помогает снизить затраты на разработку и поддержку приложения. Основной выигрыш ожидается от улучшения производительности и эффективности сотрудников при внедрении «Умного бюджета». Предполагается, что качество управления переводами улучшится, время на выполнение задач сократится, а коммуникация между сотрудниками станет более эффективной. Возможности монетизации через Google Play также представляются перспективными. Можно рассмотреть варианты введения платных функций и премиум-подписок для увеличения прибыли от приложения. Учитывая потенциальное повышение производительности на 20%, можно ожидать значительного увеличения прибыли компании.

Для расчета увеличения прибыли можно использовать следующую формулу:

$$Уп = Тп \times (1 + Кпэ) \quad (1)$$

где:

Уп - Увеличение прибыли;

Тп - текущая прибыль;

Кпэ - Коэффициент увеличения эффективности.

Учитывая, что планируем коэффициент повышения эффективности составляет 25% (0.25), можно рассчитать увеличение прибыли следующим образом:

$$Уп = 1,000,000 \times (1 + 0,25) = 1,250,000 \text{ рублей}$$

Таким образом, предполагается, что внедрение умного бюджета увеличит выручку компании на 250 тысяч рублей (1 250 000 рублей - 1 000 000 рублей). Следовательно, если проект реализован и продвигается правильно, его эффективность и успех могут превысить первоначальные затраты на разработку и внедрение.

Внедрение разумного бюджета имеет значительный потенциал для повышения эффективности работы сотрудников и улучшения управления переводами в ООО «Квартплата 24». Если проект реализован правильно, то его также можно монетизировать и получать дополнительный доход. Первым шагом была разработка дизайна приложения с использованием специальных инструментов, таких как Figma, Sketch или Adobe xD.

В процессе были использованы готовые компоненты и созданы уникальные элементы, соответствующие концепции приложения. Затем была реализована архитектура приложения и разработаны соответствующие модули, такие как хранение данных, управление переводами, конфигурирование и статистика.

Каждый модуль был создан с использованием соответствующих технологий и инструментов. Ниже приведены таблицы (Таблица 6-8) стоимости мобильного приложения «Умный бюджет».

Таблица 6 - Затраты на мобильное приложение «Умный бюджет»

Статья затрат	Сумма (руб.)
Разработка дизайна	70,000
Разработка приложения	150,000
Тестирование	50,000
Маркетинг и продвижение	30,000
Сопровождение и поддержка	20,000
Итого затраты	320,000

Таблица 7 - Ожидаемое увеличение прибыли

Показатель	Значение
Текущая прибыль	1,000,000
Коэффициент увеличения прибыли	0.25(25%)
Ожидаемая прибыль после внедрения	1,250,000
Увеличение прибыли	250,000

Таблица 8 - Итоговая таблица

Показатель	Значение
Итого затраты	320,000 руб.
Увеличение прибыли в год	250,000 руб.
Срок окупаемости	1.2 года

Таким образом, при затратах в 320,000 рублей и ожидаемом увеличении прибыли на 250,000 рублей в год, срок окупаемости проекта составит примерно 1.2 года.

### Выводы по главе 3

В третьей главе была рассмотрена реализация, тестирование и оценка эффективности мобильного приложения «Умный бюджет». Основное

внимание было уделено процессам внедрения и проверке работоспособности всех компонентов приложения.

В ходе работы была проведена тщательная разработка пользовательского интерфейса, при этом особое внимание уделено удобству и интуитивной понятности для пользователей. Использование Material Design позволило создать современный и привлекательный интерфейс, что было подтверждено результатами тестирования с участием реальных пользователей. Процесс тестирования включал в себя модульное тестирование, функциональное тестирование и тестирование пользовательского интерфейса. Модульное тестирование обеспечило изоляцию и проверку каждого компонента приложения, что позволило своевременно выявить и исправить ошибки. Функциональное тестирование подтвердило корректность взаимодействия всех модулей и соответствие функциональности приложения заявленным требованиям. Тестирование пользовательского интерфейса показало высокую степень удовлетворенности пользователей и удобство использования приложения. Экономическая эффективность проекта была оценена на основе сокращения затрат на разработку и сопровождение благодаря использованию открытых технологий и готовых решений. Повышение производительности и эффективности сотрудников, достигнутое благодаря внедрению «Умного бюджета», также положительно сказалось на общей экономической оценке проекта. Потенциальные возможности монетизации приложения через Google Play, включая платные функции и премиум-подписки, открывают дополнительные перспективы для увеличения прибыли компании.

Таким образом, третья глава подтвердила успешность разработанных решений и показала, что мобильное приложение «Умный бюджет» готово к внедрению и эксплуатации, обеспечивая высокую производительность, удобство использования и экономическую эффективность.

## Заключение

В данной выпускной квалификационной работе была рассмотрена потребность в создании инновационного приложения для автоматизации учета финансовых транзакций внутри ООО "Кварплата24". Основной акцент делался на нескольких ключевых задачах, которые мог бы решать такой инструмент, включая оптимизацию ресурсов, управление транзакциями и их распределение между сотрудниками. Автоматизация процесса финансовых транзакций не только упрощает и ускоряет работу, но и снижает вероятность ошибок и повышает точность прогнозирования.

В рамках работы были выполнены поставленные задачи. Первым этапом было детальное исследование области управления финансовыми транзакциями, что включало анализ современных методологий в этой области. Это позволило лучше понять, какие подходы могут быть применены при разработке нового приложения. Затем был проведен обзор существующих приложений для учета финансовых транзакций, что помогло определить наиболее ценные функции и возможности для пользователей, а также сформулировать функциональные требования к будущему приложению.

Далее была разработана архитектура приложения и определена стратегия взаимодействия между клиентской и серверной частями. Были выбраны соответствующие технологии и инструменты для обеспечения надежной и эффективной работы приложения.

После этого был создан пользовательский интерфейс, который был адаптирован под нужды пользователей и прошел тестирование на удобство использования. Затем приложение было развернуто на сервере и протестировано на соответствие всем функциональным требованиям.

Весь процесс разработки учитывал специфику и потребности конкретной организации, что гарантировало создание конечного продукта,

который наилучшим образом соответствует требованиям пользователей и помогает повысить эффективность работы компании.

Таким образом, при правильной реализации и продвижении проекта его эффективность и успешность могут значительно превысить изначальные затраты на разработку и внедрение.

Перспективы развития приложения "Умный Бюджет" включают несколько ключевых направлений, которые помогут значительно расширить его функциональность и повысить ценность для пользователей. Во-первых, планируется интеграция с банковскими системами, что позволит автоматизировать процесс импорта данных о транзакциях, тем самым снижая необходимость ручного ввода и минимизируя ошибки. Во-вторых, развитие аналитических возможностей и внедрение инструментов прогнозирования помогут пользователям лучше управлять своими финансами и принимать обоснованные решения.

Наконец, разработка мобильной версии приложения и улучшение мер безопасности обеспечат доступ к функционалу с любых устройств и повысят доверие пользователей к приложению. Эти направления помогут укрепить позиции "Умного Бюджета" на рынке и значительно повысить его привлекательность для широкой аудитории.

## Список используемых источников

1. Ананьев И.В., Серова Е.Г. Области эффективного применения нотации IDEF0 для задач описания бизнес-процессов // Вестник Санкт-Петербургского университета. Менеджмент. 2008. №2. URL: <https://cyberleninka.ru/article/n/oblasti-effektivnogo-primeneniya-notatsii-idef0-dlya-zadach-opisaniya-biznes-protssessov-1> (дата обращения: 03.03.2023).
2. Антал М.А. Особенности планирования в современном производстве / М.А. Антал // В сборнике: Приоритетные направления развития экономики и менеджмента: теоретические и практические аспекты. Сборник научных статей. Уфа, 2021. С. 98-101.
3. Атенцио, Л. Функциональное программирование на JavaScript: как улучшить код JavaScript-программ / Л. Атенцио. – М.: Диалектика, 2018. с. 304.
4. Блох Д. Java Эффективное программирование / Д. Блох. - М.: Лори, 2016. 440 с.
5. Блох Дж. Java: эффективное программирование / Дж. Блох. - М.: Диалектика, 2019. 464 с.
6. Васильев, А. Java. Объектно-ориентированное программирование: Учебное пособие Стандарт третьего поколения / А. Васильев. – СПб.: Питер, 2013. 400 с.
7. Выявление и сбор требований к ПО — ultimate guide | Дзен URL: [https://dzen.ru/a/Ys\\_erPQlgUMeaHvt](https://dzen.ru/a/Ys_erPQlgUMeaHvt) (дата обращения: 10.02.2022).
8. Гарнаев, А. Web-программирование на Java и JavaScript / А. Гарнаев. - СПб.: BHV, 2005. 1040 с.
9. Гаст, Х. Объектно-ориентированное проектирование: концепции и программный код / Х. Гаст. - М.: Диалектика, 2018. - 1040 с.
10. Грушвицкий, Р. Проектирование систем на микросхемах с программируемой структурой / Р. Грушвицкий. - СПб.: BHV, 2006. 736 с.
11. Давыдов, С. IntelliJ IDEA. Профессиональное программирование на Java / С. Давыдов. – СПб.: BHV, 2005. 800 с.

12. Ельцова, О.М. Проектирование основной образовательной программы (на основе программы Н.В.Нищевой) / О.М. Ельцова. - СПб.: Детство-Пресс, 2016. - 256 с.
13. Зебзеева, В.А. Проектирование образовательной программы детского сада в условиях реализации ФГОС ДО / В.А. Зебзеева. - М.: ТЦ Сфера, 2015. - 128 с.
14. Зыль, С. Проектирование, разработка и анализ программного обеспечения систем реального времени / С. Зыль. - СПб.: ВНУ, 2010. 336 с.
15. Иванов, В.Б. Программирование микроконтроллеров для начинающих. Визуальное проектирование, язык С, ассемблер / В.Б. Иванов. - СПб.: Корона-Век, 2015. 176 с.
16. Квартплата24 – официальный сайт [электронный ресурс]: Режим доступа: <https://www.kvp24.ru/>
17. Круз, Р.Л. Структуры данных и проектирование программ / Р.Л. Круз. М.: Бином, 2008. - 765 с.
18. Методология IDEF0 [электронный ресурс]: Режим доступа: <https://ru.wikipedia.org/wiki/IDEF0>
19. Назаров, С.В. Архитектура и проектирование программных систем: Монография / С.В. Назаров. - М.: НИЦ Инфра-М, 2013. 351 с.
20. Нужный, А. М. Разработка мобильных приложений на языке Java с использованием Android Studio : учебное пособие / А. М. Нужный, Н. И. Гребенникова, В. В. Сафронов. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. — 93 с.
21. Олейникова, О.Н. Модульные технологии: проектирование и разработка образовательных программ: Учебное пособие / О.Н. Олейникова, А.А. Муравьева, Ю.В. Коновалова, Е.В. Сартакова. - М.: Альфа-М, Инфра-М, 2010. 256 с.
22. Ослэндер, Д.М. Управл программы для механических систем. Объектно- ориент. проектирование систем реального времени / Д.М. Ослэндер и др. - М.: Бином. Лаборатория знаний, 2004. 413 с.

23. Панюкова Т.А. Проектирование программных средств. / Т.А. Панюкова. – М.: КД Либроком, 2012. 362 с.
24. Планировщики дел: 13 лучших программ. URL: <https://amssoft.ru/amsblog/planirovshchik-zadach.php> (дата обращения: 10.02.2022).
25. Розин, В.М. Проектирование и программирование: Методологическое исследование. Замысел. Разработка. Реализация. Исторический и социальный контекст / В.М. Розин. - М.: Ленанд, 2018. 160 с.
26. Соловьев, В.В. Логическое проектирование цифровых систем на основе программируемых логических интегральных схем / В.В. Соловьев, А. Климович. - М.: РиС, 2014. 376 с.
27. Эванс, Эрик Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем / Эрик Эванс. - М.: Вильямс И.Д., 2018. 448 с.
28. Allen, D. Getting Things Done: The Art of Stress-Free Productivity. - New York: Penguin Books, 2015. 352 p.
29. Covey, S.R. The 7 Habits of Highly Effective People: Powerful Lessons in Personal Change. – New York: Simon & Schuster, 2013. 432 p.
30. Duhigg, C. The Power of Habit: Why We Do What We Do in Life and Business. – New York: Random House, 2014. 416 p.
31. Eclipse [электронный ресурс]: Режим доступа: [https://ru.wikipedia.org/wiki/Eclipse\\_\(%D1%81%D1%80%D0%B5%D0%B4%D0%B0\\_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8\)](https://ru.wikipedia.org/wiki/Eclipse_(%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8)) (дата обращения 20.02.2024)
32. Gallagher, B. The One Thing: The Surprisingly Simple Truth Behind Extraordinary Results. – Austin: Bard Press, 2013. 240 p.
33. Sinek, S. Start with Why: How Great Leaders Inspire Everyone to Take Action. – New York: Portfolio, 2011. 256 p.
34. Tracy, B. Eat That Frog!: 21 Great Ways to Stop Procrastinating and Get More Done in Less Time. – San Francisco: Berrett-Koehler Publishers, 2017. 144 p.

35. Xamarin [электронный ресурс]: Режим доступа:  
<https://learn.microsoft.com/ru-ru/previous-versions/xamarin/get-started/what-is-xamarin> (дата обращения 20.04.2024)

## Приложение А

### Код модуля регистрации и аутентификации пользователей

```
package com.example.myapplication;

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

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

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class LoginActivity extends AppCompatActivity {

    private EditText emailEditText, passwordEditText;
    private Button loginButton;
    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        mAuth = FirebaseAuth.getInstance();

        emailEditText = findViewById(R.id.email);
        passwordEditText = findViewById(R.id.password);
        loginButton = findViewById(R.id.Login);

        loginButton.setOnClickListener(v -> signIn());
    }

    private void signIn() {
        String email = emailEditText.getText().toString().trim();
        String password = passwordEditText.getText().toString().trim();

        if (TextUtils.isEmpty(email) || TextUtils.isEmpty(password)) {
            Toast.makeText(LoginActivity.this, "Пожалуйста, заполните все поля", Toast.LENGTH_SHORT).show();
            return;
        }

        mAuth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        // Вход успешен, перейти на главный экран или выполнить другие действия
                        Toast.makeText(LoginActivity.this, "Вход успешен", Toast.LENGTH_SHORT).show();
                        startActivity(new Intent(LoginActivity.this,
                            MenuActivity.class));
                    } else {
                        // Вход не удался, вывести сообщение об ошибке
                        Toast.makeText(LoginActivity.this, "Не удалось выполнить вход: "
                            + task.getException().getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
            });
    }
}
```

## Приложение Б

### Код модуля управления переводами

```
package com.example.myapplication;

import android.content.Intent;
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.appcompat.app.AppCompatActivity;

public class MenuActivity extends AppCompatActivity {
    private TextView tvMainMenu, tvSearchResults;
    private EditText etNumberInput;
    private Button btnSearch, btnSuppliers, btnConsumers;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.menu_activity);

        // Initialize UI components
        tvMainMenu = findViewById(R.id.tvMainMenu);
        etNumberInput = findViewById(R.id.etNumberInput);
        btnSearch = findViewById(R.id.btnSearch);
        tvSearchResults = findViewById(R.id.tvSearchResults);
        btnSuppliers = findViewById(R.id.btnSuppliers);
        btnConsumers = findViewById(R.id.btnConsumers);

        // Set onClickListeners
        btnSearch.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String number = etNumberInput.getText().toString();
                if (number.isEmpty()) {
                    Toast.makeText(MenuActivity.this, "Введите число для поиска", Toast.LENGTH_SHORT).show();
                } else {
                    // Здесь вы можете добавить логику поиска
                    tvSearchResults.setText("Результаты поиска для числа: " + number);
                }
            }
        });

        btnSuppliers.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Действия при нажатии на кнопку "Поставщики услуг"
                Toast.makeText(MenuActivity.this, "Переход к Поставщикам услуг", Toast.LENGTH_SHORT).show();
                // Создаем Intent для перехода к экрану "Поставщики услуг"
                Intent intent = new Intent(MenuActivity.this, PostavshikiActivity.class);
                startActivity(intent);
            }
        });

        btnConsumers.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Действия при нажатии на кнопку "Потребители услуг"
                Toast.makeText(MenuActivity.this, "Переход к Потребителям услуг", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(MenuActivity.this, PotrebiteliActivity.class);
                startActivity(intent);
            }
        });
    }
}
```