

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки / специальности)

Разработка социальных и экономических информационных систем
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка информационной системы для финансовых решений в сфере ЖКХ

Обучающийся Т.Н. Волкова
(Инициалы Фамилия) (личная подпись)

Руководитель канд. пед. наук, доцент Е.В. Ерофеева
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант канд. пед. наук, доцент А.В. Егорова
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Аннотация

Название дипломной работы: «Разработка информационной системы для финансовых решений в сфере ЖКХ».

Выпускная работа состоит из введения, трех глав, заключения, списка литературы, включая зарубежные источники, и одного приложения.

Основной акцент работы сделан на концепции, проектировании и разработке информационной системы, предназначенной для финансового планирования и управления расходами на услуги жилищно-коммунального хозяйства.

Цель работы заключается в создании эффективного инструмента для оптимизации финансовых потоков, улучшения процесса принятия решений и продвижения к более экономичному и экологически ответственному подходу в использовании коммунальных ресурсов.

Результаты исследования демонстрируют успешную разработку веб-приложения, оснащенного необходимым функционалом для управления финансами и мониторинга использования коммунальных услуг, что не только позволяет контролировать свои расходы, но и способствует повышению финансовой осведомленности граждан.

В заключение хочется отметить, что разработанная информационная система имеет значительный потенциал для решения существующих задач в сфере ЖКХ, повышения качества жизни граждан и улучшения их финансовой грамотности, что делает данную дипломную работу актуальной и перспективной для дальнейшего развития и внедрения.

Abstract

The title of the graduation work is Development of an Information System for Financial Decisions in the Housing and Utilities Sector.

This graduation work delves into the intricacies of crafting a specialized information system aimed at enhancing financial management and expenditure planning within the housing and utilities sector.

The graduation work consists of an introduction, three substantive chapters, a conclusive section, a list of references including foreign sources, and one appendix.

The research is aimed at the development and implementation of a custom-built web application. The objective of this tool is not only to optimize financial flows, but also to promote a more economically and environmentally sustainable use of utility resources.

The author pays particular attention to the web application that emerged from this research, highlighting its pivotal roles in financial management and the monitoring of utility service consumption. This innovation not only empowers users to effectively manage their expenses but also plays a crucial role in fostering financial literacy across the populace. By addressing the challenges inherent within the housing and utilities domain, the developed system showcases considerable potential in enhancing citizens' quality of life while simultaneously elevating their financial acumen.

In conclusion, it should be emphasized that the developed information system has significant potential for solving existing problems in the field of housing and communal services, improving the quality of life of citizens and improving their financial literacy, which makes this thesis relevant and promising for further development and implementation.

Оглавление

Введение.....	5
Глава 1 Анализ предметной области.....	6
1.1 Описание организации.....	6
1.2 Выбор нотации для моделирования бизнес-процессов.....	7
1.3 Разработка процессной модели «Как есть»	9
1.4 Определение требований и бизнес-целей	10
1.5 Разработка процессной модели «Как должно быть»	12
Глава 2 Проектирование и разработка	14
2.1 Проектирование архитектуры	14
2.2 Проектирование интерфейса	20
2.3 Проектирование базы данных	28
2.4 Разработка системы финансовых решений.....	30
Глава 3 Тестирование.....	37
3.1 План тестирования.....	37
3.2 Тестовые случаи.....	38
3.3 Баг репорт	40
Заключение	44
Список используемой литературы и используемых источников.....	46
Приложение А Модель «Как есть» процесса регистрации показаний.....	50

Введение

В современном мире, где информационные технологии проникают во все сферы жизни человека, особое внимание уделяется оптимизации и эффективности управления различными аспектами социальной и экономической деятельности [20]. Одной из важных областей, требующих инновационных подходов и решений, является сфера жилищно-коммунального хозяйства (ЖКХ) [4].

Данная выпускная квалификационная работа (ВКР) выполнялась с целью исследования и разработки информационной системы для управления финансами в сфере ЖКХ с упором на вопросы повышения финансовой грамотности населения, осведомленности о коммунальных вопросах и просвещения в сфере ЖКХ.

Объект исследования – процессы финансового учета и планирования в сфере ЖКХ.

Предмет исследования – информационная система для финансовых решений.

Цель исследования – разработка эффективной информационной системы для оптимизации финансовых решений в сфере ЖКХ, то есть создание такой системы, которая поможет упростить и автоматизировать процессы финансового планирования и анализа в ЖКХ, повысить их эффективность и доступность для пользователей.

Задачи исследования:

- анализ требований и особенностей финансовых процессов в сфере ЖКХ,
- проектирование архитектуры информационной системы,
- проектирование интерфейса, включающего в себя основные функции,
- разработка и тестирование системы.

Глава 1 Анализ предметной области

1.1 Описание организации

«Квартплата 24» является федеральной IT компанией, специализирующейся на разработке и внедрении информационных технологий в сфере жилищно-коммунального хозяйства (ЖКХ). Основными целями компании являются повышение эффективности управления жилищно-коммунальным сектором, обеспечение прозрачности и удобства взаимодействия между участниками данного процесса, а также содействие в соблюдении законодательства.

Взаимодействие с клиентами «Квартплата 24» охватывает широкий спектр организаций и учреждений, включая товарищества собственников жилья (ТСЖ), жилищно-строительные кооперативы (ЖСК), управляющие компании (УК), ресурсоснабжающие организации (РСО), а также крупные единые информационно-расчетные центры (ЕИРЦ) на всей территории России. Более 700 клиентов пользуются сервисами «Квартплата 24», что подтверждает их доверие и эффективность предоставляемых решений.

Для того, чтобы сделать ЖКХ более удобным и прозрачным для всех его участников, была разработана и успешно внедрена экосистема облачных сервисов, которая позволяет автоматизировать процессы расчета и учета платы за жилищно-коммунальные услуги.

«Квартплата 24» является ведущим игроком на рынке IT-решений в сфере ЖКХ, предоставляющим клиентам инновационные технологии для эффективного управления жилищно-коммунальным сектором и повышения качества жизни граждан [13].

На рисунке 1 изображена организационная структура компании «Квартплата 24», которая включает несколько ключевых подразделений, таких как отдел разработки, общий отдел, служба экспертной поддержки, отдел по работе с клиентами и финансовый отдел. Каждый отдел имеет свои

обязанности и функции, специализируясь на определенных аспектах деятельности компании.



Рисунок 1 – Организационная структура компании

На вершине иерархии находится директор, который управляет всеми процессами в компании. Отдел разработки отвечает за разработку продуктов и услуг, технические инновации и поддержку информационных систем. Общий отдел занимается административно-хозяйственными вопросами, управлением персоналом и организационной структурой. Служба экспертной поддержки предоставляет специализированную поддержку и консультации в области услуг компании. Отдел по работе с клиентами ответственен за общение с клиентами, обработку обратной связи и поддержание клиентских отношений. Финансовый отдел управляет финансовыми потоками компании, бюджетированием, учетом и отчетностью [22].

Эти ключевые отделы работают совместно для обеспечения эффективного функционирования «Квартплата 24», реализации стратегии и достижения поставленных целей.

1.2 Выбор нотации для моделирования бизнес-процессов

На выбор нотации для моделирования бизнес-процессов влияют те задачи и структурные требования, которые стоят перед организацией. В рамках нашего анализа предметной области были рассмотрены три известные

нотации моделирования: IDEF0, EPC и BPMN.

IDEF (Integrated Computer Automated Manufacturing Definition) – это семейство нотаций. Одна из самых популярных в нем — IDEF0, она подходит для функционального моделирования. Ее основные элементы — прямоугольники и стрелки [24].

Нотация EPC (Event-Driven Process Chain) – цепочка процессов, управляемая событиями [5]. Она подходит, когда необходимо детально смоделировать процесс или сделать графическую инструкцию. Часто ее используют для создания подробных регламентов [24].

BPMN (Business Process Model and Notation) – одна из самых популярных нотаций для бизнес-моделирования. У нее есть возможность визуально изобразить зоны ответственности исполнителей — по диаграмме легко понять, кто и за какие действия и события отвечает. И возможность изобразить развилки — альтернативные и параллельные процессы [24].

На базе проведенного анализа вышеперечисленных нотаций была составлена таблица 1, иллюстрирующая ключевые характеристики этих нотаций:

Таблица 1 – Сравнение нотаций моделирования бизнес-процессов

Критерий	IDEF0	EPC	BPMN
Высокая степень детализации	+	-	-
Подходит для однопоточных процессов без развилки	+	-	-
Подходит для ветвящихся процессов	-	+	+
Визуально простая и наглядная для восприятия	-	+	+
Сложность чтения и интерпретации без навыков	+	-	-
Подходит для описания бизнес-процессов	-	+	+
Понятность даже для рядовых сотрудников	-	+	+
Возможность создания исполняемой модели	-	-	+
Необходимость создания модели для каждого этапа	-	+	-

Эта таблица отражает как преимущества, так и недостатки каждой нотации. Например, IDEF0 обеспечивает высокую степень детализации и подходит для описания линейных процессов, но тяжело читается и плохо

подходит для ветвящихся процессов. EPC легко воспринимается визуально и хорошо подходит для описания ветвящихся процессов, однако структура EPC может стать перегруженной из-за обилия элементов.

Можно сделать вывод, что BPMN является наиболее подходящей нотацией для моделирования бизнес-процессов в данной работе, так как поможет четко и наглядно отобразить сложные процессы, включая взаимодействие с финансовыми потоками, работу с клиентами и учет ресурсов. Это может значительно облегчить понимание логики работы системы, ее последующее тестирование, документирование и внедрение. BPMN выделяется своей простотой в освоении и эффективной визуализацией процессов. Также значительное значение имеют масштабируемость и доступность инструментария. BPMN обеспечивает необходимую степень детализации и наглядности для визуализации сложных процессов в ЖКХ, что критически важно для точной передачи информации между всеми сторонами проекта.

1.3 Разработка процессной модели «Как есть»

В рамках данного раздела рассмотрим процесс регистрации показаний приборов учета. Как было решено в прошлом разделе, разработка процессной модели «Как есть» проводится с использованием нотации BPMN, что позволяет в ясной форме представить последовательность шагов процесса.

Процессная модель «Как есть» выполняется для анализа существующего на момент исследования процесса внесения пользователями показаний. Данное исследование направлено на описание нынешних операций, производимых пользователями для регистрации своих данных через системы личного доступа.

Целью данного этапа является обеспечение точного понимания текущего механизма работы процесса внесения показаний, что необходимо для корректной разработки и функционирования предстоящего приложения.

Итак, процессная модель «Как есть» для регистрации показаний приборов учета, представленная на рисунке 1 в Приложении А, предполагает наличие трех основных пулов: «Пользователь», «Квартплата 24» и «Исполнитель ЖКУ». Пул «Квартплата 24» дополнительно разделен на две дорожки, которые представляют «Личный кабинет» и «Сервис расчета».

Процесс начинается с того, что житель входит в личный кабинет и вводит показания приборов учета. Если они некорректны, ему предложат ввести их заново. После проверки на корректность показания отправляются на расчет по актуальным тарифам. Как только расчеты выполнены, житель получает уведомление об успешном вводе данных и квитанцию в личном кабинете.

Таким образом, процессная модель «Как есть», описанная с использованием BPMN для регистрации показаний приборов учета, дает полное представление о взаимодействии между пользователем, «Квартплатой 24» и исполнителем ЖКУ, а также о ключевых шагах процесса от авторизации до получения подтверждения о зарегистрированных показаниях [3].

1.4 Определение требований и бизнес-целей

Для разработки нашей системы необходимо провести детальный анализ требований пользователей. Основная цель этого этапа – понять, какие функциональные и нефункциональные требования должны быть удовлетворены для успешного развертывания системы. Тщательно сформулированные требования обеспечивают необходимую основу для создания системы, отвечающей всем потребностям конечного пользователя.

Функциональные требования – это конкретные особенности и способы функционирования системы, которые ориентированы на решение задач пользователя [19]. Для нашей информационной системы функциональные требования включают:

- модуль планирования бюджета, который обеспечивает автоматический импорт данных о тарифах и расходах, и предлагает удобный инструмент для управления финансами, где можно составлять бюджет и отслеживать расходы с возможностью оптимизации [18];
- модуль рекомендаций по уменьшению потребления ресурсов, который предлагает аналитику потребления, определение аномалий в расходах и персонализированные рекомендации для снижения расходов на коммунальные услуги.

Нефункциональные требования описывают ожидания характеристики системы и ее окружения, а не поведение системы [7]. Для нашей системы они включают:

- надежность и безопасность – гарантия защиты личных данных пользователей и устойчивость работы системы при различных условиях;
- масштабируемость – способность системы адаптироваться к увеличению числа пользователей и объема обрабатываемых данных;
- производительность – быстрдействие системы, короткое время отклика на запросы пользователя;
- интуитивность и удобство использования – легкость освоения системы пользователями без специальной подготовки;
- совместимость – возможность интеграции с существующими системами учета в сфере ЖКХ [8].

Продолжая формирование плана разработки информационной системы для сферы ЖКХ, важно обратить внимание на детальное планирование бизнес-целей, которые будут достигнуты посредством реализации проекта [25].

Бизнес-цели формулируются таким образом, чтобы способствовать повышению эффективности управления в сфере ЖКХ, улучшению взаимодействия с жильцами и оптимизации расходов. Ключевые бизнес-цели включают:

- оптимизация расходов для пользователей посредством эффективного планирования и управления бюджетом на коммунальные услуги;
- повышение качества жизни пользователей за счет контроля над собственными финансами и снижения потребления ресурсов.

Для достижения вышеизложенных бизнес-целей, проект должен включать следующие шаги:

- разработка модулей для мониторинга потребления коммунальных услуг, что позволит жителям видеть реальную картину своих расходов и находить способы для их оптимизации;
- создание пользовательского интерфейса для веб-приложения, интуитивно понятного и доступного, обеспечивающего легкость использования всех функций системы.

Для поддержания постоянной обратной связи между пользователями системы и разработчиками, необходимо ввести механизмы для сбора и анализа пользовательских отзывов и предложений, что позволит периодически адаптировать систему под изменяющиеся потребности и требования конечного пользователя [17].

Этот анализ позволяет сформировать базу для дальнейшего проектирования и разработки информационной системы, которая будет наиболее точно соответствовать потребностям пользователей и требованиям сферы ЖКХ.

1.5 Разработка процессной модели «Как должно быть»

Этап разработки модели «Как должно быть» предполагает создание схемы, в которой текущий процесс передачи показаний в системе «Квартплата 24» используется как подпроцесс без внесения изменений в него самого. Мы строим связи с существующей экосистемой через API, анализируем полученные данные и предоставляем результаты пользователям, зарегистрированным в личном кабинете.

На рисунке 2 представлена модель, демонстрирующая, как данные передаются из «Квартплата 24» к «Системе финансовых решений». В этом контексте процесс «Регистрация показаний» становится начальной точкой для передачи информации, которая далее обрабатывается и визуализируется нашей системой, предоставляя дополнительные услуги пользователям.

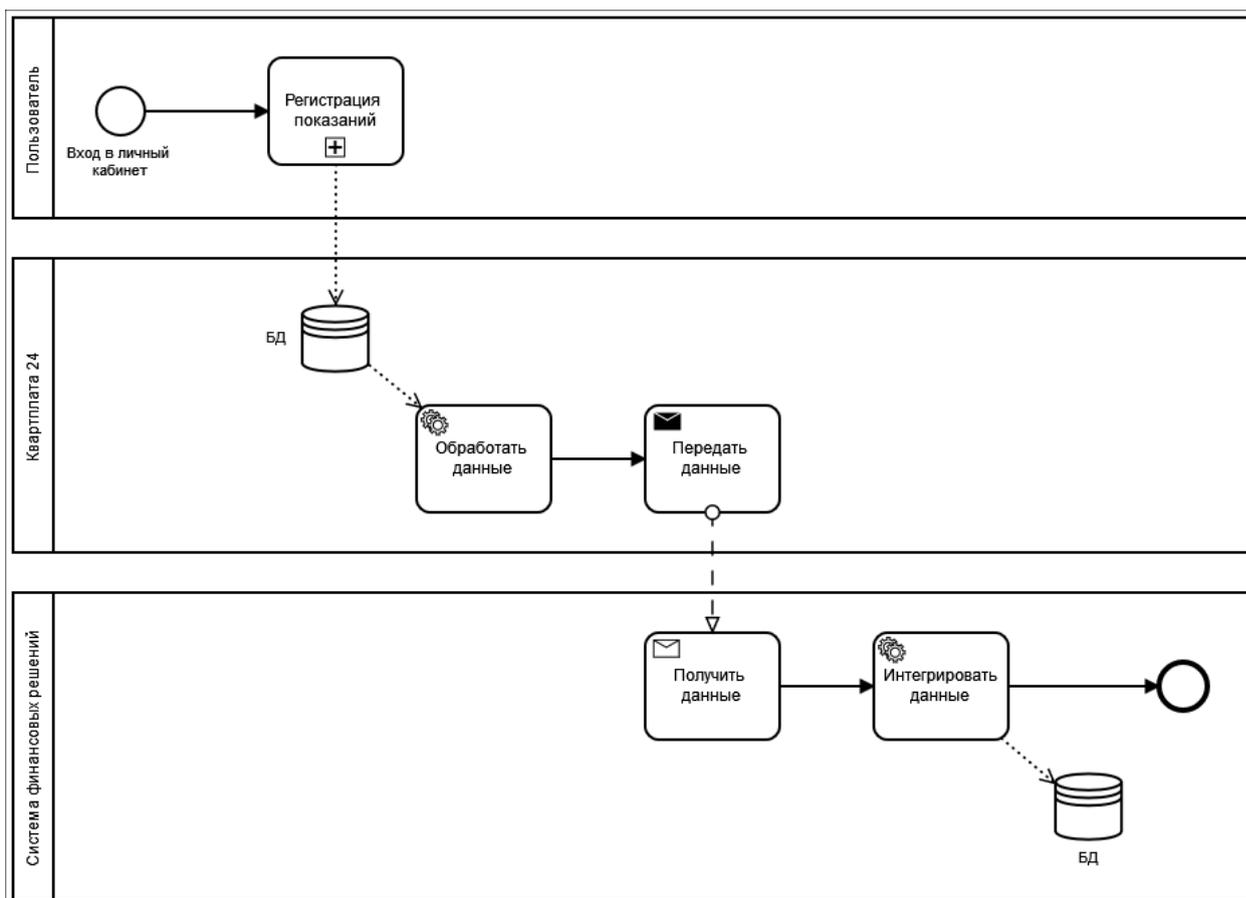


Рисунок 2 – Модель «Как должно быть» процесса передачи данных

Так, благодаря этой модели, в экосистеме «Квартплата 24» появляется новый компонент, который расширяет функционал для пользователей, действуя как дополнение существующих сервисов.

Глава 2 Проектирование и разработка

2.1 Проектирование архитектуры

После определения требований к информационной системе мы начинаем процесс проектирования ее архитектуры с помощью UML (unified model language) – языка для визуализации, специфицирования, конструирования и документирования программных систем [11]. Архитектура системы определяет структуру приложения, включая его компоненты, взаимосвязи между ними и принципы их взаимодействия. Это важный этап, который позволяет обеспечить эффективное функционирование системы и удовлетворение требований пользователей.

Первым шагом в проектировании архитектуры является создание диаграммы вариантов использования (или же прецедентов), которая служит для описания функциональных требований к программному обеспечению с привязкой к пользователям системы, в котором не предполагается спецификация реализации этих требований [11].

На рисунке 3 изображена диаграмма вариантов использования, где представлены основные функциональные возможности (прецеденты) нашей информационной системы и актеры, которые с этими прецедентами ассоциируются.

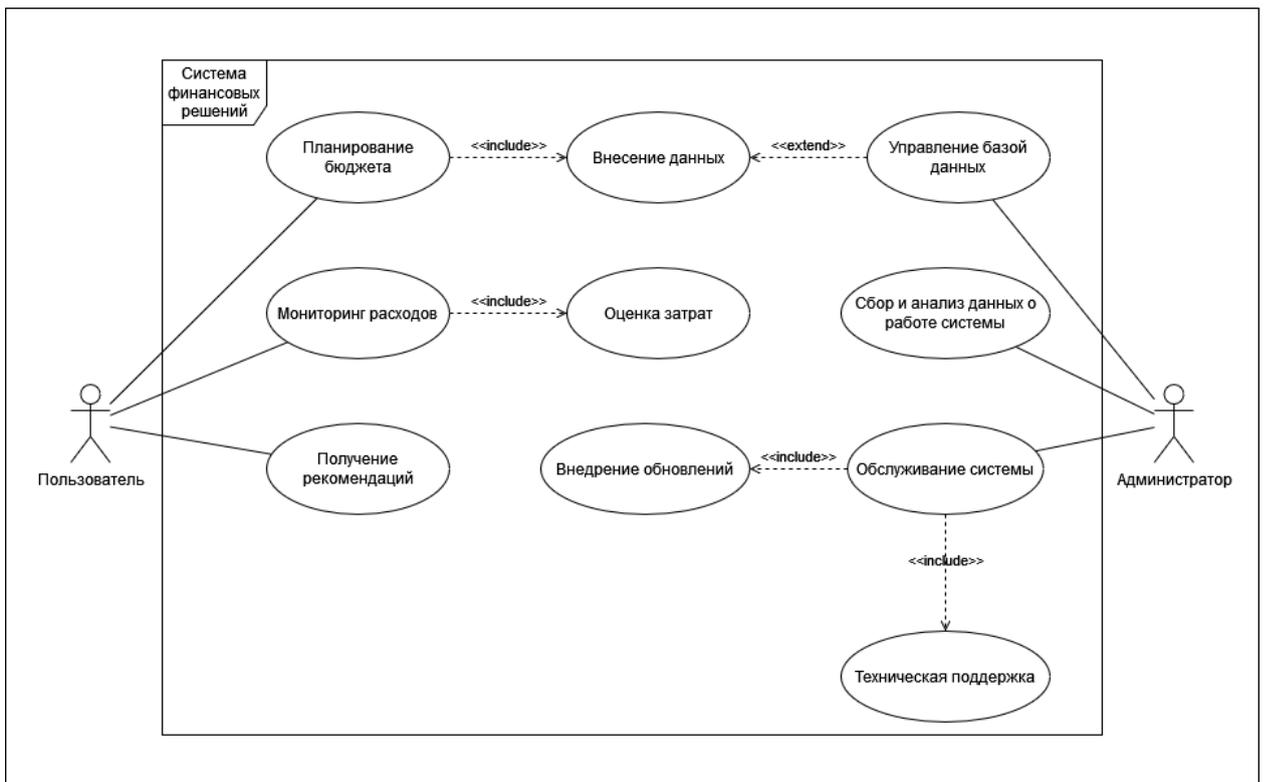


Рисунок 3 – Диаграмма вариантов использования

Данная диаграмма демонстрирует, как оба актера взаимодействуют с системой, и подчеркивает, что администраторы играют ключевую роль в поддержании оперативности и доступности сервисов для обычных пользователей, в то время как пользователю доступны действия для эффективного управления личными финансами. Эти возможности предоставляют инструменты для осознанного контроля над финансами, стимулируют ответственное потребление, а также помогают экономить на ежедневных расходах.

Для визуализации последовательности взаимодействия объектов в системе используется диаграмма последовательности [9]. Она показывает обмен сообщениями между объектами в определенном сценарии использования. Это помогает понять, какие шаги выполняются для реализации определенного действия и какие объекты при этом участвуют.

На рисунке 4 представлена диаграмма последовательности.

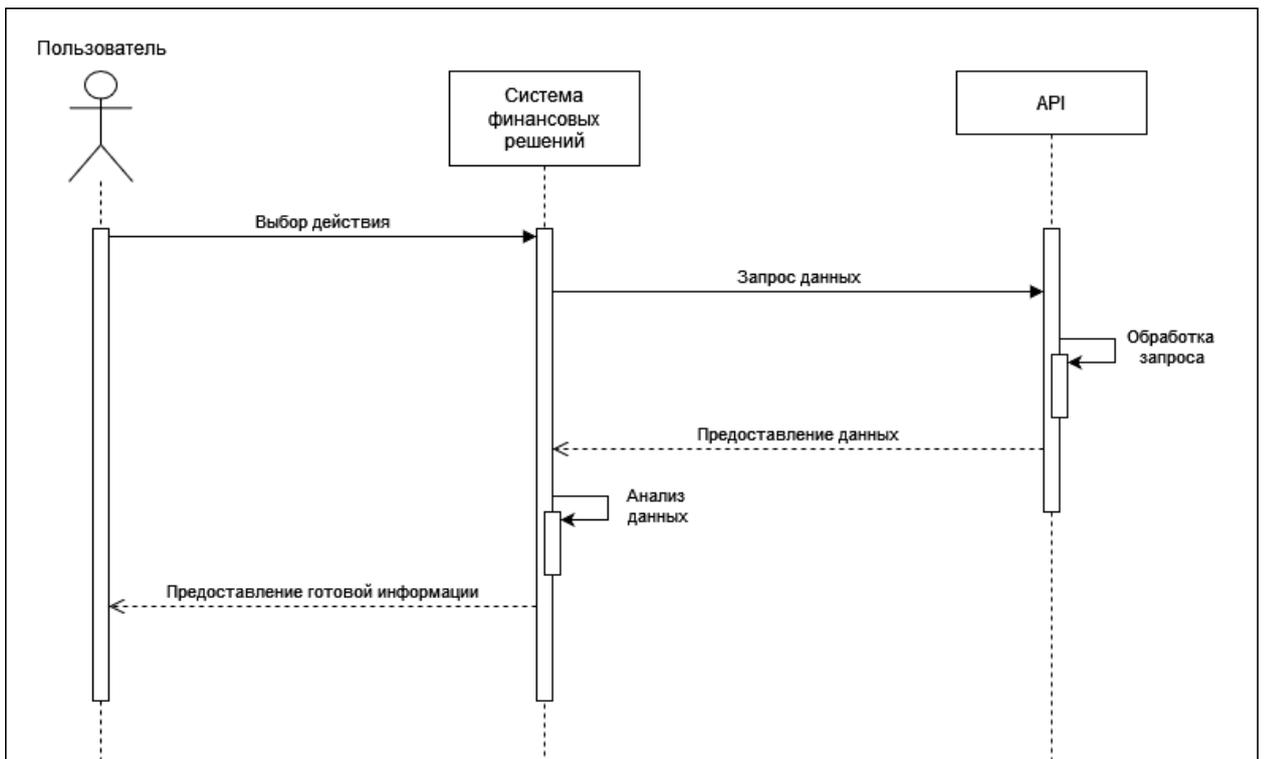


Рисунок 4 – Диаграмма последовательности

Данная диаграмма изображает процесс взаимодействия пользователя с приложением, который включает в себя цикл выбора действия, запроса к API, обработки данных и получения результата.

Далее разработаем диаграмму компонентов, которая отображает разбиение программного кода на крупные блоки (структурные компоненты) и показывает зависимости между ними. Диаграмма компонентов отображает также способы доставки сообщений (через порты и интерфейсы) [9].

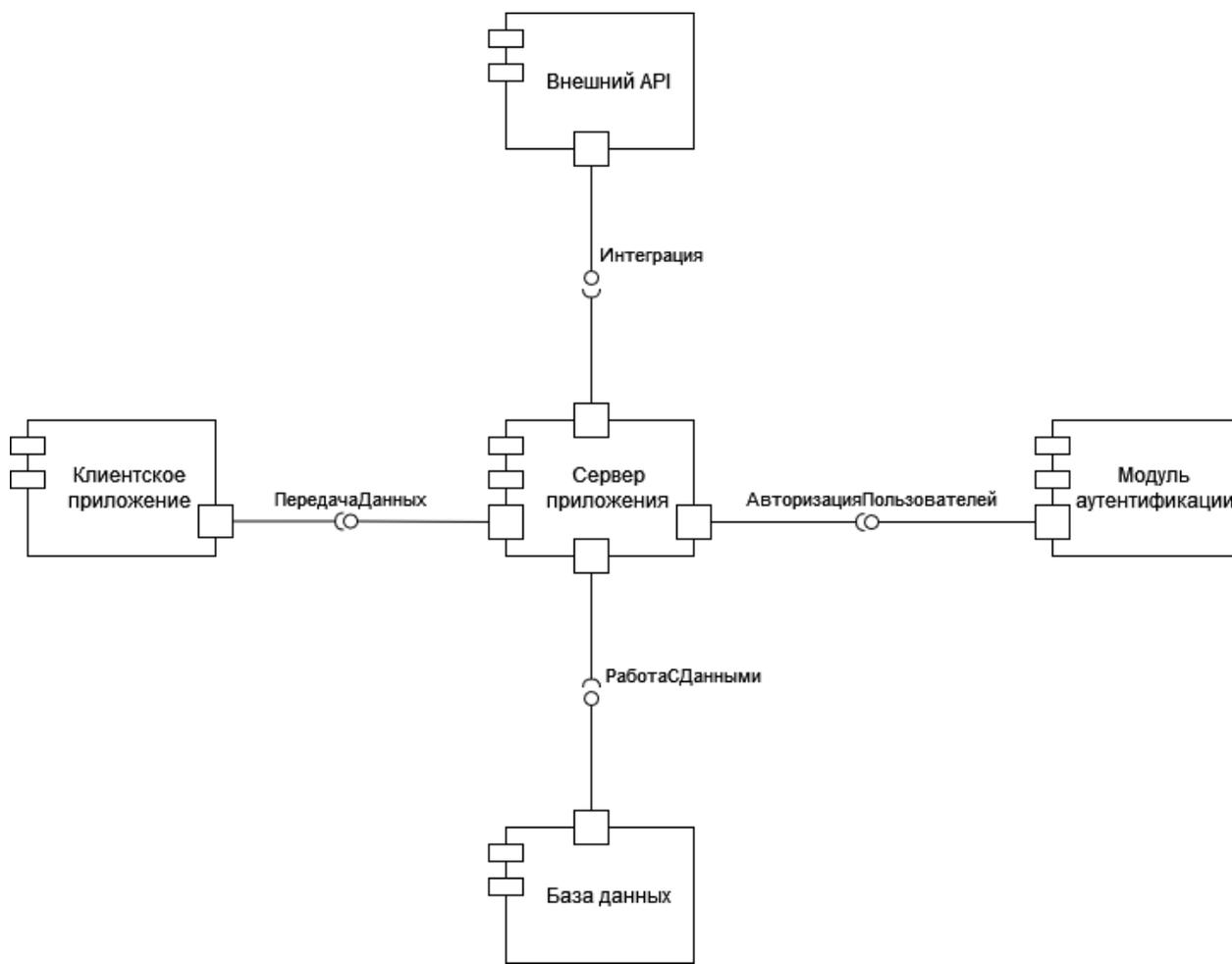


Рисунок 5 – Диаграмма компонентов

На диаграмме компонентов, изображенной на рисунке 5, представлены пять основных элементов: клиентское приложение, сервер приложения, модуль аутентификации, внешний API, а также база данных.

Клиентское приложение – это интерфейс, с которым напрямую взаимодействует пользователь. Клиентское приложение предоставляет пользовательский интерфейс и часть бизнес-логики. Оно отвечает за отправку запросов на серверное приложение и получение от него данных.

Сервер приложений является центральным узлом системы. Он обрабатывает запросы от клиентского приложения, выполняет бизнес-логику, взаимодействует с базой данных и другими сервисами, такими как модули аутентификации и внешние API. Он отвечает за координацию всех операций, обеспечение безопасности и интеграцию компонентов системы.

Внешний API (программный интерфейс приложения) – это интерфейс, предоставляемый сервисом «Квартплата 24». В него входят:

- данные об исполнителе услуг,
- баланс лицевого счета,
- сведения о приборах учета и дате их следующей проверки,
- информация о начислениях с детализацией,
- расчет пени с детализацией,
- доступ к платежным документам,
- платежи и чеки об оплате услуг,
- заявки в аварийно-диспетчерскую службу и информация об их статусах,
- сведения о пользователях личного кабинета жителя.

База данных – это централизованное хранилище данных для системы, в котором сохраняется и используется информация, такая как пользовательские данные, настройки системы и так далее.

Модуль аутентификации – это компонент, отвечающий за проверку подлинности пользователей. Его задача включает в себя идентификацию пользователя (аутентификацию) и предоставление информации о его правах и ролях в системе (авторизация).

Таким образом, в нашей системе клиентское приложение напрямую связано с сервером приложения. В свою очередь, сервер приложения взаимодействует с тремя другими компонентами: проводит аутентификацию через модуль аутентификации, обращается к базе данных для исполнения запросов данных и использует внешний API для доступа к данным из экосистемы «Квартплаты 24».

Наконец, приступим к разработке диаграммы классов (рисунок б), с помощью которой (через классы, их атрибуты, методы и зависимости между классами) описываются модель предметной области и структура моделируемой системы [9].

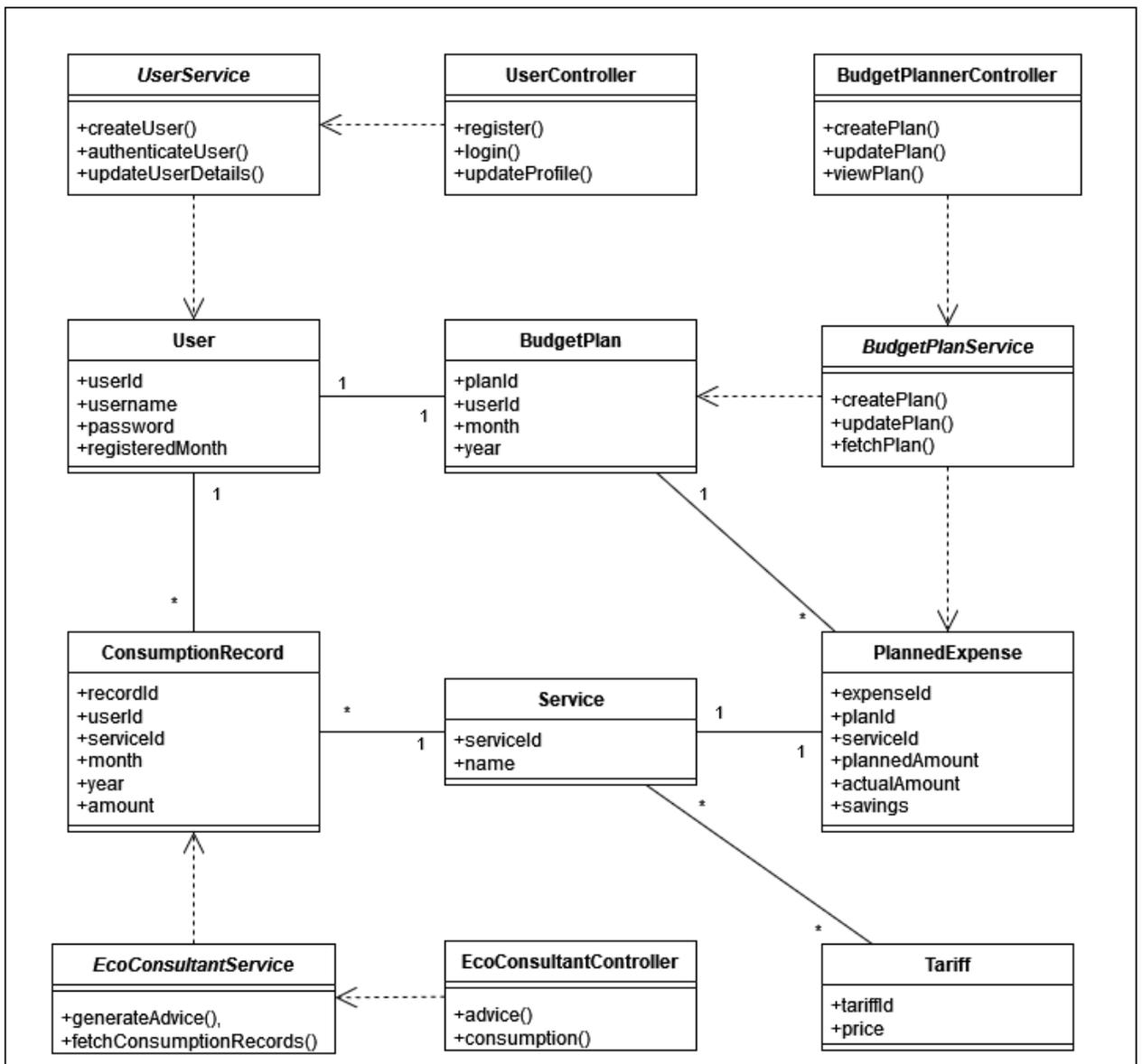


Рисунок 6 – Диаграмма классов

Диаграмма классов отражает систему финансовых решений. Она включает классы для работы с информацией о пользователях, финансовом планировании, советах по экономии и так далее.

На диаграмме мы видим два основных типа связей: ассоциация и зависимость. Зависимость (dependency) – связь использования (один класс использует информацию и сервис другого). Ассоциация (association) – структурная связь между объектами. Она отличается от зависимости тем, что эта связь симметрична. Ассоциация связывает классы, объекты которых

нуждаются во взаимодействии, но не на уровне использования в качестве параметров методов. Она соединяет объекты одного класса с объектами другого как равноправные элементы [9].

Так, на представленной диаграмме классов мы видим, что контроллеры зависят от сервисов, которые в свою очередь зависят от моделей. Модели же ассоциируются друг с другом бинарными связями «один-к-одному», «один-ко-многим» и «многие-ко-многим».

Подобная структура обеспечивает четкую организацию системы, легкость в расширении и добавлении новых функций, а также удобство в обслуживании и поддержке приложения.

Разработка разноплановых UML-диаграмм обеспечивает глубокое понимание бизнес-логики и требований к системе, что важно для точного формирования архитектуры. Диаграмма вариантов использования раскрывает взаимодействие пользователей с системой и основные сценарии использования, в то время как диаграмма последовательности иллюстрирует детальный процесс обработки информации и потоки данных между компонентами системы. Диаграммы компонентов и классов подчеркивают внутреннюю организацию системы, ее модульность и выделение ответственности между различными частями системы. В совокупности эти инструменты проектирования архитектуры способствуют созданию гибкой, масштабируемой и поддерживаемой системы, тем самым закладывая крепкий фундамент для ее дальнейшего развития и сопровождения.

2.2 Проектирование интерфейса

В этом разделе рассматривается ключевой этап в проектировании информационной системы — разработка пользовательского интерфейса (UI). Основное внимание уделяется как эстетическим аспектам дизайна, так и его функциональности и удобству использования. Проектирование интерфейса является важным этапом, определяющим, насколько легко пользователи

смогут взаимодействовать с системой и выполнять необходимые задачи.

Иницилируем процесс с определения структуры интерфейса и расположения ключевых элементов управления. В рамках этого этапа, сформируем прототипы основных экранов системы, что позволит оценить эргономику и логику взаимодействия элементов, минуя детализацию дизайна [6].

«Требования к прототипу изменяются со временем. Сначала наиболее актуальными его свойствами являются скорость создания и простота модификации. Эти свойства позволяют быстро разработать и проверить несколько версий интерфейса, при этом еще и исправить значительную часть ошибок» [14].

«Затем на первый план выходят функциональность и эстетичность, простота же модификации уже не столь важна, поскольку с каждой новой исправленной ошибкой снижается вероятность того, что прототип придется полностью переделывать при обнаружении новой ошибки» [14].

«Поэтому всегда правильно сделать прототип настолько похожим на результирующую систему, насколько позволяет самая поздняя его версия. Первый прототип стоит делать максимально примитивным» [14].

Для начала создается фрейм стандартного размера (1440 на 1024) и настраивается сетка в виде колонок, которые служат визуальными ориентирами для размещения элементов интерфейса, таких как текст, кнопки и изображения, обеспечивая их равномерное и гармоничное распределение. Результат данного этапа представлен на рисунке 7.

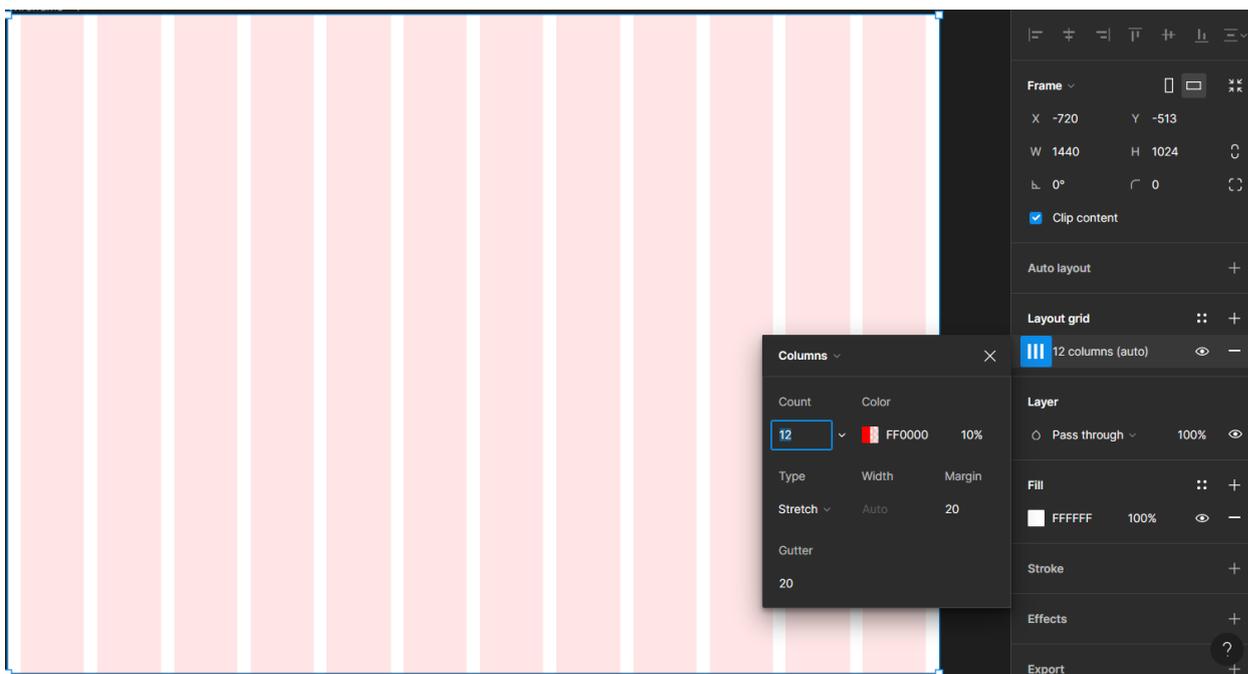


Рисунок 7 – Первый этап создания прототипа

Следующим шагом задаем расположение элементов при помощи блоков (рисунок 8). Этот этап позволяет предварительно определить структуру страницы, рассматривая элементы дизайна как составные блоки широкого спектра возможностей. Использование блоков помогает в визуализации общего компоновочного решения, обеспечивая логичное и интуитивно понятное размещение элементов управления, контента и навигационных компонентов. Создание блоков дает возможность экспериментировать с различными композиционными решениями, ища оптимальное расположение для каждого элемента, чтобы обеспечить эффективное взаимодействие пользователя с интерфейсом.

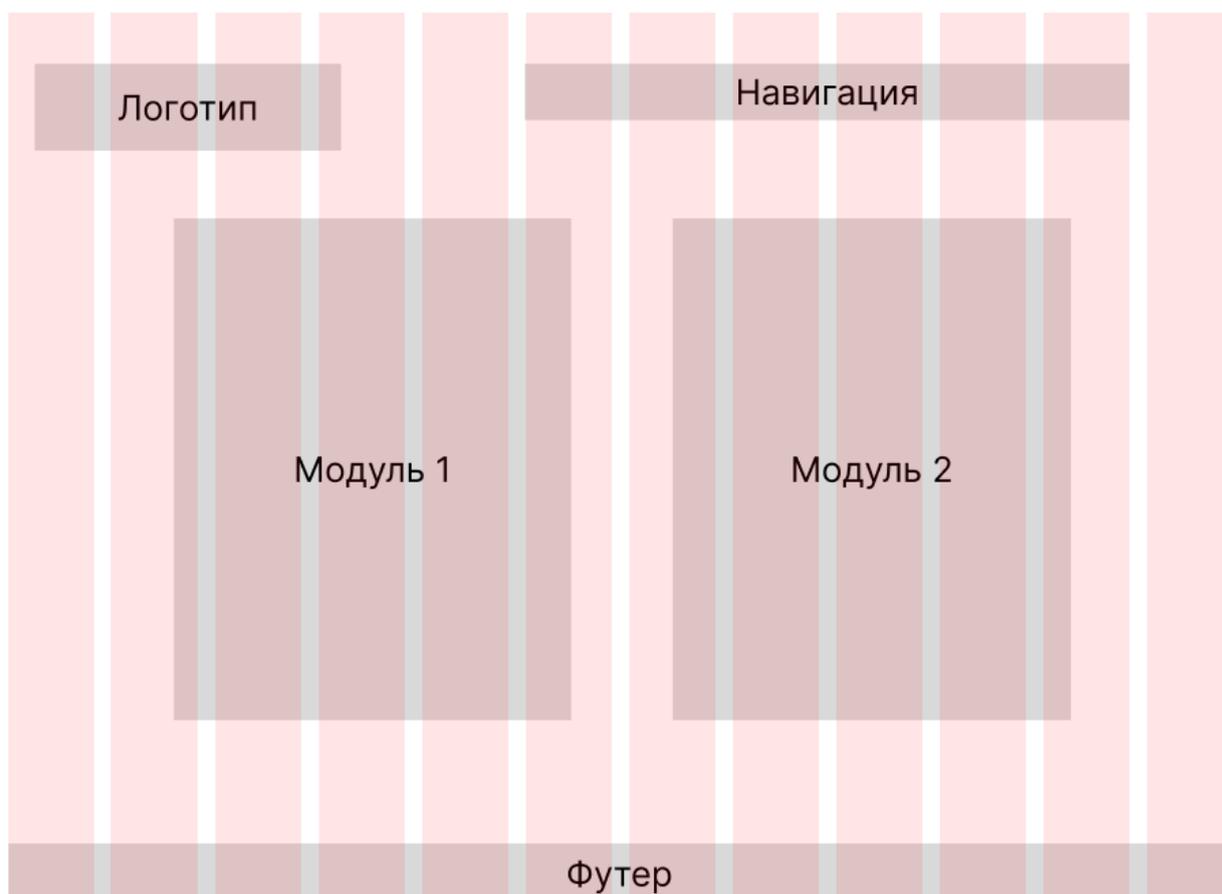


Рисунок 8 – Второй этап создания прототипа

Начнем постепенно расширять блоки и добавлять текст для большего понимания о качестве структуризации информации (рисунок 9).

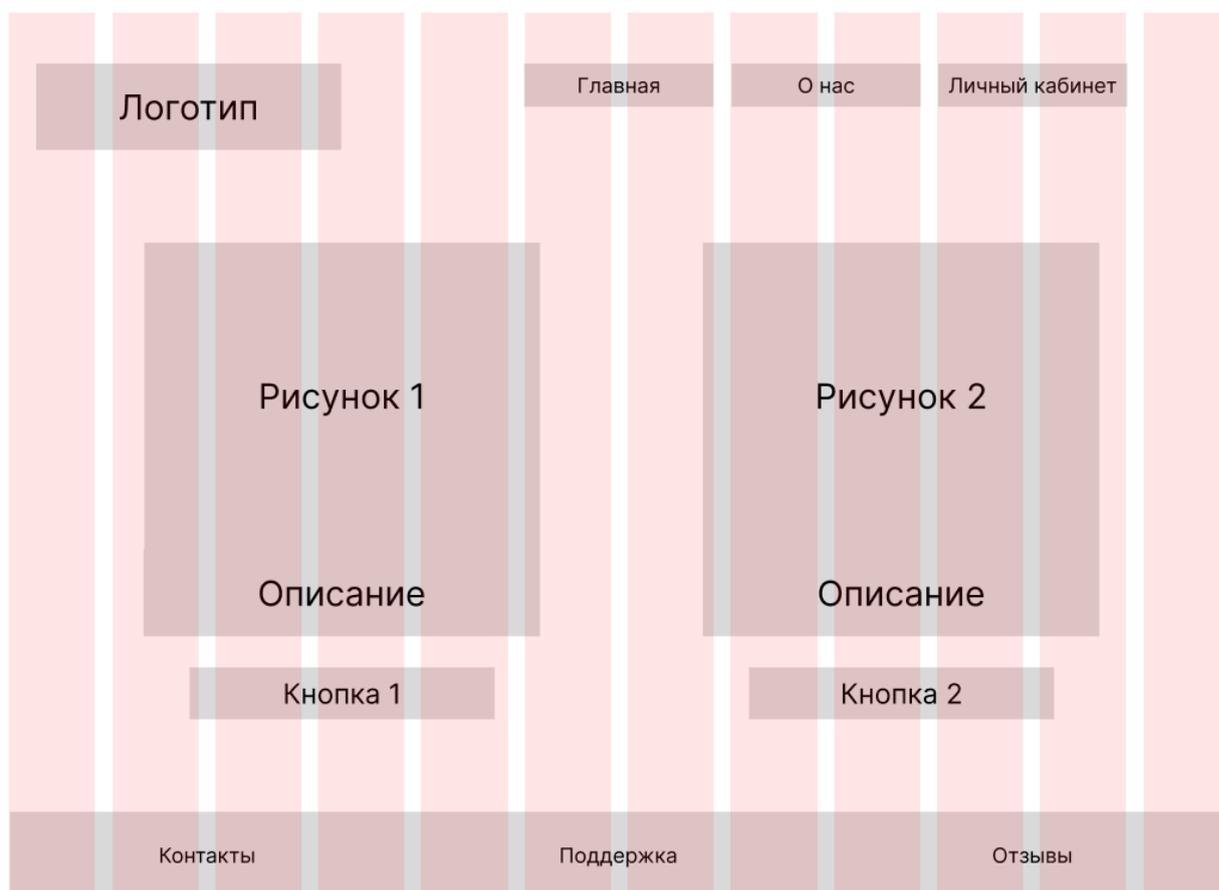


Рисунок 9 – Третий этап создания прототипа

После примитивного прототипирования переходим к высокофидельному дизайну (то есть более детализированным прототипам), который охватывает выбор цветовой схемы, шрифтов и стилей элементов управления [15].

На рисунке 10 представлена итоговая версия интерфейса главной страницы нашего веб-приложения.



Бюджетный Планировщик – ваш первый шаг к осознанному управлению личными финансами. Этот инструмент дает вам возможность не только планировать, но и анализировать свои расходы, делая ощутимым каждый шаг на пути к финансовому благополучию.

[Планировать бюджет](#)



Эко Консультант – ваш персональный гид в мире энергосбережения и устойчивого потребления. Получите полезные и практические рекомендации, которые помогут сократить расходы на коммунальные платежи и вместе с тем снизить ваш экологический след.

[Получить рекомендации](#)

Контакты

Поддержка

Отзывы

Рисунок 10 – Главная страница

Дизайн выполнен в фирменном цвете «Квартплаты 24» с использованием логотипа компании. Спроектированный интерфейс четко разграничивает функциональные зоны для упрощения восприятия. Благодаря использованию интуитивно понятного интерфейса с традиционно размещенным навигационным меню и акцентированными кнопками действий, пользователю легко идентифицировать необходимые инструменты. Визуальные элементы, подчеркивающие стилистику и значимость финансового управления и экологической осведомленности, улучшают взаимодействие и способствуют гармоничному пользовательскому опыту. В результате, интерфейс демонстрирует высокий уровень доступности и функциональности, обеспечивая эффективное взаимодействие с веб-приложением.

На заключительном этапе проделываем те же шаги в разработке

интерфейса других страниц веб-приложения. Результат мы можем видеть на рисунках 11 и 12.

Финансовые решения **24**

Главная Личный кабинет О нас

Назад **Бюджетный Планировщик**

Январь 2024

Выберите услугу: Выберите

Планируемые расходы (руб.): 0

Обновить данные

Услуги	Тарифы	Показания счетчиков	Планируемые расходы (руб.)	Фактические расходы (руб.)	Экономия (руб.)
ГВС					
ХВС					
ЭЛН					
ЭЛД					
Итого:	-	-	0,00	0,00	0,00

Перейти к рекомендациям

Контакты Поддержка Отзывы

Рисунок 11 – Интерфейс «Бюджетного Планировщика»

Данная страница содержит выпадающие меню для выбора периода и услуг, интуитивно понятные поля ввода для планирования и отслеживания расходов по коммунальным услугам, а также ясно маркированные кнопки обновления данных, которые создают легкую для понимания и использования структуру. Визуальное разграничение элементов планировщика бюджета сочетается с общим минималистичным дизайном, не перегружая пользователя, но при этом предоставляя все необходимые сведения для управления финансами. Таблица с детальным учётом расходов позволяет визуализировать и детализировать бюджетные позиции, улучшая общий опыт ведения домашней бухгалтерии. Этот интерфейс способствует более

ответственному и осознанному отношению к личным финансам.

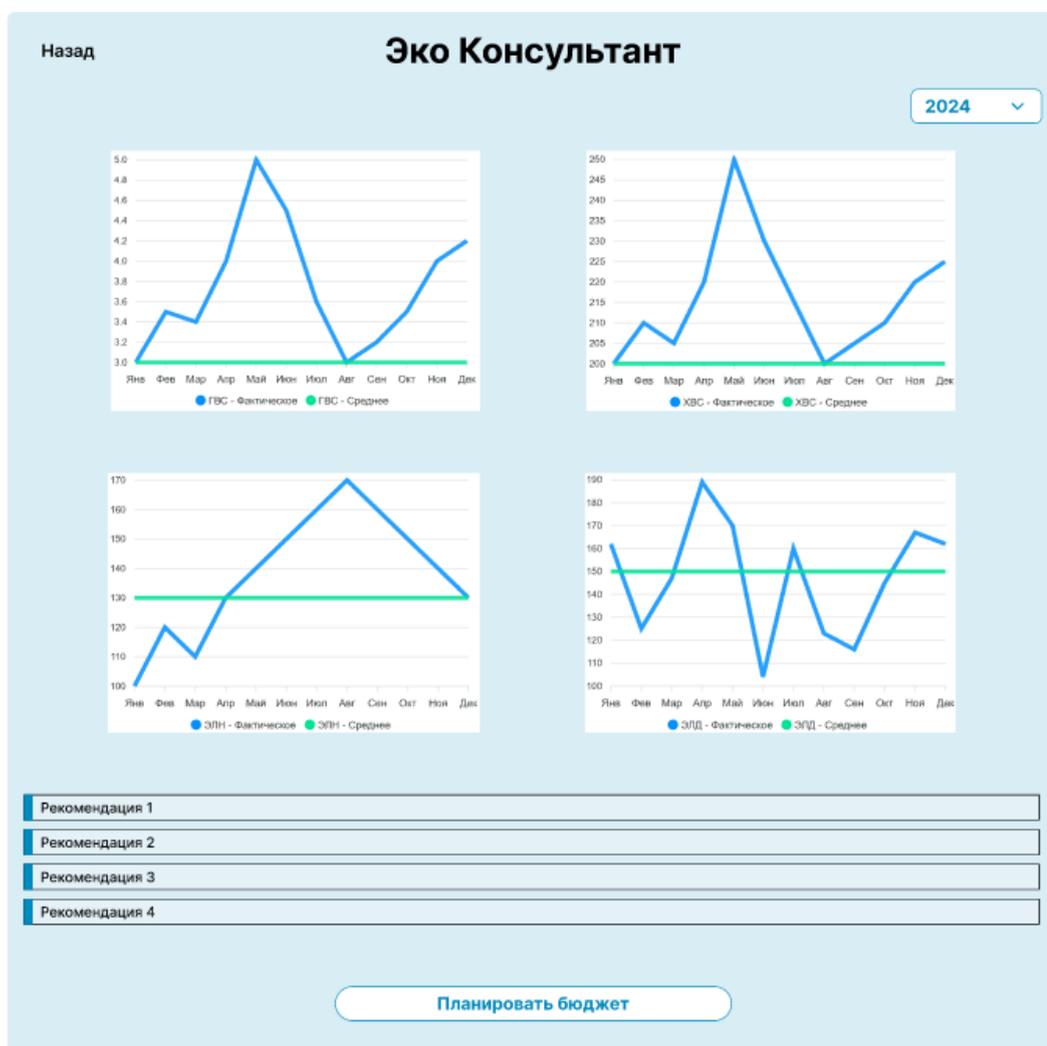


Рисунок 12 – Интерфейс «Эко Консультанта»

Интерфейс страницы «Эко Консультанта» включает в себя графическое отображение финансовых данных и рекомендаций для планирования бюджета. Общий дизайн страницы способствует легкой ориентации и созданию продуктивного пользовательского опыта, поддерживая основную концепцию эффективного финансового планирования с учетом экологической

устойчивости и осознанности.

В итоге, были созданы и представлены детализированные прототипы, которые включают в себя распределение различных элементов, добавление текста для структуризации информации и декоративные элементы интерфейса.

2.3 Проектирование базы данных

В данном разделе мы подробно рассмотрим процесс проектирования базы данных, используя PostgreSQL, который является ведущим в мире открытым проектом по созданию СУБД.

В отличие от MySQL, которая была изначально нацелена, скорее, на решение задач поддержки веб-сайтов, PostgreSQL нацелена на поддержку более широкого класса приложений, в том числе корпоративного уровня [2].

Процесс начинается с составления детального плана, который включает в себя определение ключевых сущностей и их взаимосвязей. На этом этапе мы активно используем инструмент управления базами данных pgAdmin для создания структуры таблиц, которые отражали бы необходимую информацию для каждой сущности, такие как пользователи, счетчики, тарифы, советы по экономии энергии и оценка воздействия на окружающую среду.

В каждой таблице мы определяли поля, их типы данных, а также настраивали первичные и внешние ключи для установления связей.

После создания структуры таблиц в pgAdmin, мы перешли к визуализации этих взаимосвязей и структуры в целом с помощью инструмента ERD Tool для создания ER-диаграмм, или же диаграмм «сущность-связь», которая определяет логическую или физическую (реализацию) структуру базы данных системы [12].

Используя нотацию Мартина («Crow's Foot», она же «воронья лапка»), мы наглядно изобразили связи между таблицами, что позволило лучше понять картины потока данных и структуру базы данных в целом. Таким образом

была сформирована логическая модель данных (рисунок 13), которая поддерживает эффективное хранение и обработку данных.

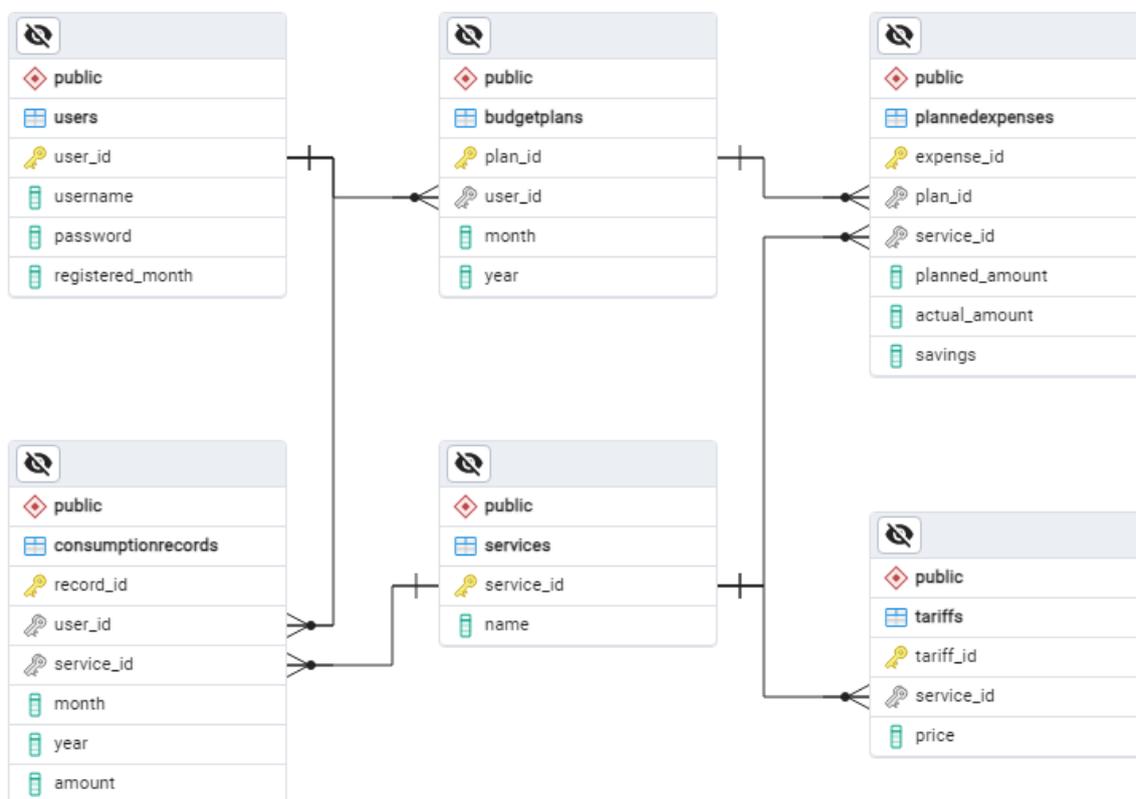


Рисунок 13 – Логическая модель данных

В итоге наша база данных поддерживает широкий спектр операций, необходимых для выполнения функций информационной системы. Созданная логическая модель способствует четкому пониманию структуры и взаимосвязей данных, что обеспечивает основу для эффективной реализации системы. Использование pgAdmin для работы с таблицами и последующее создание логической модели позволило нам не только разработать структурированную и эффективную базу данных, но и обеспечить ее гибкость для будущего расширения и адаптации под изменяющиеся требования.

2.4 Разработка системы финансовых решений

После успешного проектирования базы данных, перейдем к разработке информационной системы. Основной задачей стоит воплощение функционала, предусмотренного в техническом задании, с обеспечением необходимой гибкости и масштабируемости для последующих модификаций и улучшений.

Для реализации серверной части приложения был выбран фреймворк Spring Boot, так как это полезный проект, целью которого является упрощение создания приложений на основе Spring. Он позволяет наиболее простым способом создать веб-приложение, требуя от разработчиков минимум усилий по его настройке и написанию кода [10]. В качестве системы управления базами данных использовалась PostgreSQL для обеспечения надежного хранения данных и высокой скорости работы с ними. Для взаимодействия с базой данных применялся Spring Data JPA, который упрощает реализацию репозитория данных и улучшает производительность за счет механизма кеширования.

Фронтенд приложения был разработан с использованием HTML, который является основным языком разметки для создания структуры веб-страницы и используется для определения элементов на странице, таких как заголовки, параграфы, изображения и ссылки, CSS для определения стиля и внешнего вида веб-страницы и JavaScript, который используется для добавления интерактивности на веб-страницы [1]. Использование этих языков вместе позволяет нам создавать интерактивный и динамичный пользовательский интерфейс.

Приложение было разработано с учетом принципов RESTful API, что обеспечивает гибкость и удобство в интеграции с сервисами экосистемы «Квартплата 24».

В приложении была применена многоуровневая архитектура, разделяющая логику управления данными, бизнес-логику и представление.

Использование такой архитектуры приносит ряд преимуществ, включая защиту от SQL-инъекций, предотвращающих атаки с возможностью неправомерного влияния на базу данных, разграничение доступа к данным, возможность модификации данных перед отправкой клиенту, масштабируемость приложения для работы на множестве серверов с одной базой данных и сниженные требования к качеству соединения пользователя, поскольку обработка и фильтрация информации происходит на сервере, что уменьшает объем данных, отправляемых клиенту [10].

Разрабатываемая система включает в себя обработку данных для бюджетного планирования и предоставление рекомендаций по уменьшению потребления коммунальных ресурсов.

«Бюджетный планировщик» – это основной компонент нашего веб-приложения. Он позволяет пользователям просматривать и редактировать свой ежемесячный бюджет на коммунальные услуги. Рассмотрим несколько ключевых методов для бюджетного планировщика.

Листинг 1 – Фрагменты методов класса BudgetPlanService

```
public BudgetPlan createPlan(Long userId) {
    BudgetPlan newPlan = new BudgetPlan(userId,
    LocalDate.now().getMonthValue(), LocalDate.now().getYear());
    return budgetPlanRepository.save(newPlan);
}

public BudgetPlan updatePlan(Long planId, List<PlannedExpense> expenses) {
    BudgetPlan plan = budgetPlanRepository.findById(planId).orElseThrow(() ->
    new PlanNotFoundException(planId));
    plan.setPlannedExpenses(expenses);
    return budgetPlanRepository.save(plan);
}
```

```

public BudgetPlan fetchPlan(Long userId, int month, int year) {
    return budgetPlanRepository.findByUserIdAndMonthAndYear(userId, month,
year);
}

```

Листинг 1 демонстрирует следующие методы класса-сервиса BudgetPlanService:

- createPlan создает новый план бюджета для пользователя с идентификатором userId на текущий месяц и год и возвращает созданный план после его сохранения в базе данных;
- updatePlan обновляет план бюджета, идентифицированный по planId, добавляя или обновляя список планируемых расходов (expenses), и далее возвращает обновленный план после сохранения изменений;
- fetchPlan извлекает план бюджета для пользователя с идентификатором userId за указанный месяц и год. В случае наличия плана возвращает его, иначе может выбросить исключение о его отсутствии.

«Эко консультант» – это второй ключевой компонент разрабатываемого веб-приложения. Он предоставляет пользователям рекомендации по уменьшению потребления коммунальных ресурсов на основе их текущего потребления и бюджетного плана.

Листинг 2 – Фрагменты методов класса EcoConsultantService

```

public List<Recommendation> generateAdvice(Long userId) {
    List<ConsumptionRecord> records =
consumptionRecordRepository.findAllByUserId(userId);
}

```

```

        Map<Long, BigDecimal> totalConsumptionByService = new
HashMap<>();
        Map<Long, Integer> countByService = new HashMap<>();

        for (ConsumptionRecord record : records) {
            totalConsumptionByService.put(record.getServiceId(),
                totalConsumptionByService.getDefault(record.getServiceId(),
                    BigDecimal.ZERO).add(record.getAmount()));
            countByService.put(record.getServiceId(),
countByService.getDefault(record.getServiceId(), 0) + 1);
        }

        List<Recommendation> recommendations = new ArrayList<>();
        for (Long serviceId : totalConsumptionByService.keySet()) {
            BigDecimal averageConsumption =
totalConsumptionByService.get(serviceId)
                .divide(new BigDecimal(countByService.get(serviceId)), 2,
                    RoundingMode.HALF_UP);

            if (averageConsumption.compareTo(new BigDecimal("100")) > 0) {
                recommendations.add(new Recommendation(serviceId, ""));
            }
        }

        return recommendations;
    }

    public List<ConsumptionRecord> fetchConsumptionRecords(Long userId,
int year) {

```

```
        return consumptionRecordRepository.findAllByUserIdAndYear(userId,
year);
    }
```

Листинг 2 представляет реализацию системы генерации рекомендаций на основе анализа потребления ресурсов пользователя.

Метод `generateAdvice` начинает с извлечения всех записей о потреблении для заданного пользователя и проходит по ним, суммируя общее потребление и количество записей по каждой услуге. Затем, используя агрегированные данные, метод рассчитывает среднее потребление для каждой услуги. На основе среднего потребления метод генерирует рекомендации.

Метод `fetchConsumptionRecords` предназначен для извлечения записей о потреблении ресурсов пользователя за указанный год.

Для наглядного представления информации о потреблении коммунальных услуг наша система включает визуализацию данных, используя библиотеку `Chart.js` для создания интерактивных графиков, которые позволяют пользователям легко анализировать и сравнивать их ежемесячное потребление с средним уровнем.

Листинг 3 – Фрагмент JavaScript-кода для создания графиков

```
function createChart(service, canvasId) {
    const ctx = document.getElementById(canvasId).getContext('2d');
    const chart = new Chart(ctx, {
        type: 'line',
        data: {
            labels: ['Январь', 'Февраль', 'Март', 'Апрель', 'Май', 'Июнь', 'Июль',
'Август', 'Сентябрь', 'Октябрь', 'Ноябрь', 'Декабрь'],
            datasets: [{
                label: 'Ежемесячное потребление',
```

```

        data: monthlyData[service],
        borderColor: 'blue',
        borderWidth: 2,
        fill: false
    }, {
        label: 'Среднее потребление',
        data:
new
Array(monthlyData[service].length).fill(averageData[service]),
        borderColor: 'red',
        borderWidth: 2,
        borderDash: [5, 5],
        fill: false
    }
    ],
    options: {
        scales: {
            y: {
                beginAtZero: true
            }
        }
    }
});

```

Листинг 3 демонстрирует фрагмент JavaScript-кода, который строит графики с двумя линиями для каждой услуги: линия ежемесячного потребления пользователя и линия среднего потребления за анализируемый период. Это позволяет пользователям оценивать их использование ресурсов относительно среднего уровня, выявлять тренды и принимать обоснованные решения по управлению их коммунальными расходами. Такой подход

способствует повышению осведомленности о потреблении и стимулирует более экономное использование ресурсов.

Интерактивность графиков обеспечивает удобство в использовании: пользователи могут наводить курсор на различные точки данных для получения более детальной информации, такой как точное количество потребленных ресурсов в конкретный месяц. Простота восприятия информации через визуальные инструменты делает наше приложение эффективным средством для контроля над личными финансами в сфере коммунальных платежей.

В целом разработка веб-приложения для бюджетного планирования и эффективного использования коммунальных ресурсов выполнялась с акцентом на функциональность и удобство. Разработанная система обладает высоким потенциалом для расширения своего функционала в будущем.

Глава 3 Тестирование

3.1 План тестирования

План тестирования предназначен для комплексной проверки системы финансовых решений. Он охватывает как функциональные, так и нефункциональные аспекты системы, гарантируя ее эффективность, безопасность и пользовательскую доступность.

Целью тестирования является проверка соответствия информационной системы функциональным требованиям и удостоверение в его надежности и стабильности работы. Тестирование направлено на выявление возможных дефектов и недочетов, которые могут повлиять на работоспособность и безопасность системы.

В процессе тестирования будут рассмотрены:

- функциональные компоненты системы,
- пользовательский интерфейс и его взаимодействие с пользователем,
- серверная часть, обеспечивающие обработку данных,
- база данных и ее структура.

Тестирование будет включать в себя следующие виды:

- модульное тестирование – проверка отдельных компонентов или модулей программного продукта на корректность работы [1];
- интеграционное тестирование – оценка взаимодействия между разными компонентами или модулями системы [1];
- функциональное тестирование – проверка выполнения программным продуктом заявленных функций и их соответствия ожиданиям пользователей [1];
- нагрузочное тестирование – оценка производительности и устойчивости системы при высоких нагрузках.

Тестирование информационной системы начинается после завершения разработки основных функций. Также к этому моменту должна быть настроена тестовая среда. Важно учесть, что все необходимые для процесса

тестирования инструменты должны быть доступны.

Тестирование считается завершенным, когда каждый из запланированных тестов был выполнен. Этот этап также предполагает, что все ошибки системы, которые могут нарушить ее основную функциональность, были обнаружены и устранены. Кроме того, производительность системы обязательно должна соответствовать всем заявленным требованиям. И, наконец, должна быть получена положительная оценка степени готовности информационной системы к эксплуатации.

В процессе тестирования стоит уделить особое внимание пользовательскому опыту при работе с системой, проверяя интуитивность интерфейса и простоту выполнения основных задач.

Для процесса тестирования будет задействована тестовая среда, включающая локальный сервер и базу данных. Помимо ручного тестирования, будут применены инструменты для автоматизации тестирования. К таким инструментам относятся JMeter, применяемый для нагрузочного тестирования, и JUnit для модульного и интеграционного тестирования.

План тестирования учитывает индивидуальный характер разработки и ориентирован на максимальное покрытие ключевых функций информационной системы с целью обеспечения ее качества и надежности.

3.2 Тестовые случаи

Процесс тестирования системы финансовых решений включал ряд тестовых случаев, то есть артефактов, описывающих совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части [16]. Основываясь на плане тестирования, были разработаны и выполнены тесты для основных компонентов системы, позволяя оценить их работоспособность и взаимодействие.

На основе проведенного тестирования была составлена таблица тест-кейсов (таблица 2), которая служит ключевым инструментом для анализа

производительности и надежности системы, а также для планирования последующих шагов по улучшению и оптимизации.

Таблица 2 – Тест-кейсы

Описание теста	Ожидаемые результаты	Фактические результаты	Статус
Расчет показаний счетчиков	Точный расчет на основе входных данных	Расчет выполнен корректно	Пройден
Подсчет итоговой суммы в таблице	Суммы по столбцам вычислены правильно	Вычисления точны	Пройден
Отрисовка графиков потребления ЖКУ	Графики отображаются корректно	Проблемы с отображением	Провален
Генерация рекомендаций	Рекомендации соответствуют анализу данных	Рекомендации актуальны	Пройден
Взаимодействие выпадающего списка с таблицей	Смена месяца/года обновляет данные в таблице	Данные обновлены верно	Пройден
Интеграция данных графиков	Выбор года влияет на отображение графиков	Графики соответствуют выбору	Пройден
Обмен данными между модулями	Бесперебойный обмен данными	Нарушение в обмене данными	Провален
Функциональность кнопки «Назад»	Возврат на главную страницу	Возврат осуществлен	Пройден
Функциональность кнопки «Перейти к рекомендациям»	Переход на страницу «Эко консультанта»	Переход выполнен	Пройден
Отображение графиков после выбора в выпадающем списке	Графики отображаются в соответствии с выбором	Отображение не соответствует выбору	Провален
Функциональность кнопки «Планировать бюджет»	Перенаправление на «Бюджетный планировщик»	Перенаправление выполнено	Пройден
Производительность при вводе данных	Система обрабатывает ввод без задержек	Задержки при обработке ввода	Провален
Производительность при просмотре графиков	Быстрая загрузка графиков при высокой нагрузке	Графики загружены быстро	Пройден
Генерация рекомендаций при нагрузке	Система быстро генерирует рекомендации	Задержки в генерации	Провален

Тестирование показало, что несмотря на успешное выполнение большинства тестов, некоторые ключевые аспекты системы требуют

дальнейшей доработки. Особое внимание будет уделено улучшению компонента отрисовки графиков, устранению проблем с производительностью и оптимизации процесса обмена данными между модулями. Эти усилия помогут повысить стабильность работы информационной системы и повысить общую удовлетворенность пользователей.

3.3 Баг репорт

В результате выполнения тестовых случаев для системы финансовых решений были идентифицированы дефекты, требующие внимания. Составление баг репортов является следующим шагом в процессе устранения выявленных проблем. Они не только фиксируют детали обнаруженных дефектов, но и служат направлением для разработчиков в процессе дефектного анализа и последующей корректировки.

Были подробно описаны шаги воспроизведения для каждого бага, представлено сравнение ожидаемых и фактических результатов, а также определены приоритет и серьезность каждой обнаруженной проблемы. Это позволяет четко идентифицировать порядок действий для исправления дефектов и способствует более целенаправленному устранению ошибок в системе. Наряду с этим, в баг репортах указаны статусы дефектов и идентификационные номера для удобства отслеживания [23].

В таблицах 3-7 представлены конкретные баг репорты, документирующие каждый из основных выявленных дефектов:

Таблица 3 – Баг репорт «Проблемы с отрисовкой графиков потребления ЖКУ»

Атрибут	Описание
ID	101
Название	Ошибка отображения графиков потребления ЖКУ

Продолжение таблицы 3

Статус	Открыт
Приоритет	Высокий
Серьезность	Критическая
Описание	Графики потребления ЖКУ не отображаются после выбора месяца и года
Шаги воспроизведения	1. Зайти на страницу потребления ЖКУ, 2. Выбрать месяц и год, 3. Нажать кнопку «Показать графики»
Ожидаемый результат	Графики отображаются корректно в соответствии с выбранными параметрами
Фактический результат	Графики не отображаются

Таблица 4 – Баг репорт «Некорректный обмен данными между модулями»

Атрибут	Описание
ID	102
Название	Ошибка синхронизации данных между «Бюджетный планировщик» и «Эко консультант»
Статус	Открыт
Приоритет	Высокий
Серьезность	Высокая
Описание	Модуль «Бюджетный планировщик» не передает данные в «Эко консультант»
Шаги воспроизведения	1. Заполнить данные в «Бюджетный планировщик», 2. Перейти в «Эко консультант», 3. Попытаться получить рекомендации
Ожидаемый результат	«Эко консультант» использует данные из «Бюджетного планировщика» для генерации рекомендаций
Фактический результат	«Эко консультант» не распознает данные, рекомендации не генерируются

Таблица 5 – Баг репорт «Ошибка отображения графиков после выбора в выпадающем списке»

Атрибут	Описание
ID	103
Название	Неактуальные данные графиков после выбора даты
Статус	Открыт
Приоритет	Средний
Серьезность	Средняя

Продолжение таблицы 5

Описание	После выбора месяца и года графики показывают неактуальные данные
Шаги воспроизведения	1. Открыть выпадающий список выбора даты на странице графиков, 2. Выбрать месяц и год, 3. Посмотреть графики
Ожидаемый результат	Графики корректно отражают данные за выбранный период
Фактический результат	Графики показывают данные за другой период или не обновляются

Таблица 6 – Баг репорт «Проблемы производительности при множественном вводе данных»

Атрибут	Описание
ID	104
Название	Замедление системы при одновременном вводе данных
Статус	Открыт
Приоритет	Средний
Серьезность	Высокая
Описание	Система реагирует с задержкой на ввод данных множеством пользователей.
Шаги воспроизведения	1. Несколько пользователей одновременно вводят данные в систему, 2. Наблюдается отклик системы
Ожидаемый результат	Безотказная работа системы при высокой нагрузке
Фактический результат	Система замедляет свою работу, возможны зависания

Таблица 7 – Баг репорт «Задержка генерации рекомендаций при высокой нагрузке»

Атрибут	Описание
ID	105
Название	Медленная генерация рекомендаций при высокой нагрузке
Статус	Открыт
Приоритет	Высокий
Серьезность	Высокая
Описание	При высоком количестве запросов система не успевает оперативно генерировать рекомендации

Продолжение таблицы 7

Шаги воспроизведения	1. Имитация высокой нагрузки на сервер запросами на генерацию рекомендаций, 2. Замер времени отклика
Ожидаемый результат	Быстрая генерация рекомендаций даже при высокой нагрузке
Фактический результат	Значительные задержки в обработке запросов.

В рамках жизненного цикла дефектов, каждый баг проходит через фазы подтверждения, назначения, исправления, повторного тестирования и закрытия [16]. Баг репорты служат не только для устранения текущих проблем, но и как историческая запись, способствующая предотвращению подобных инцидентов в будущем. Эффективное управление багами и их своевременное решение улучшают качество информационной системы и обеспечивают удовлетворение потребностей пользователей. Кроме того, они создают ценную базу знаний для улучшения процессов разработки и поддержки.

Заключение

Подводя итоги данной работы, можно сказать, что поставленная цель была успешно достигнута. Разработанная система позволяет упростить и автоматизировать ключевые процессы финансового планирования и анализа, содействуя повышению их производительности и обеспечивая пользователю гибкий инструментарий для управления финансами [21].

В первой главе был проведен всесторонний анализ предметной области. Мы рассмотрели организацию «Квартплата 24», её структуру и ключевые функции. Также был проведен анализ различных нотаций для моделирования бизнес-процессов, таких как IDEF0, EPC и BPMN, и выбрана наиболее подходящая для нашей задачи — BPMN. Это позволило нам создать процессную модель «Как есть», которая описывает текущий механизм работы процесса внесения показаний, и определить требования и бизнес-цели для разработки системы.

Во второй главе было осуществлено проектирование архитектуры системы с использованием UML-диаграмм, что позволило четко определить структуру и взаимодействие компонентов системы. Были разработан пользовательский интерфейс, а также спроектирована база данных с использованием PostgreSQL. Важным этапом стало проектирование и разработка ключевых компонентов системы, таких как «Бюджетный планировщик» и «Эко консультант».

В третьей главе был разработан план тестирования, охватывающий функциональные и нефункциональные аспекты системы. Тестирование включало модульное, интеграционное, функциональное и нагрузочное тестирование, что позволило выявить и устранить возникшие недочеты. Это гарантирует надежность и стабильность функционирования информационной системы.

Система обладает значительным потенциалом для наращивания функциональных возможностей и масштабирования, что делает её

перспективной для дальнейшего развития и внедрения. Использование системы способствует не только оптимизации финансовых потоков, но и улучшению финансовой осведомлённости пользователей. Это актуально, учитывая постоянно увеличивающийся объём информации, а также необходимость принятия обоснованных и взвешенных решений.

В заключение следует отметить, что дипломная работа отражает потребность современного общества в технологических решениях, обеспечивающих эффективное управление и рациональное использование ресурсов в сфере ЖКХ. Разрабатываемое решение может оказать значительное влияние на качество жизни граждан, их удовлетворенность коммунальными услугами и на повышение общего уровня финансовой грамотности в данной сфере.

Список используемой литературы и используемых источников

1. Баланов, А. Н. Комплексное руководство по разработке: от мобильных приложений до веб-технологий : учебное пособие для вузов / А. Н. Баланов. — Санкт-Петербург : Лань, 2024. — 412 с. — ISBN 978-5-507-48841-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/394577>
2. Баранчиков, А. И. Теоретические основы реляционных баз данных : учебное пособие / А. И. Баранчиков. — Рязань : РГРТУ, 2023. — 160 с. — ISBN 978-5-7722-0367-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/380363>
3. Бедердинова, О. И. Технологии моделирования бизнес-процессов : учебное пособие / О.И. Бедердинова. — Москва : ИНФРА-М, 2022. — 102 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-111154-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1913625>
4. Богданова, Е. Н. Финансовое планирование и бюджетирование на предприятии в условиях цифровизации экономики : учебное пособие / Е.Н. Богданова, О.И. Бедердинова. — Москва : ИНФРА-М, 2023. — 110 с. — (Высшее образование: Магистратура). - ISBN 978-5-16-112202-0. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2131726>
5. Бояркин, Г. Н. Моделирование бизнес-процессов : учебное пособие / Г. Н. Бояркин. — Омск : ОмГТУ, 2020. — 94 с. — ISBN 978-5-8149-3034-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/186887>
6. Вагин, Д. В. Современные технологии разработки веб-приложений : учебное пособие / Д. В. Вагин, Р. В. Петров. - Новосибирск : Изд-во

- НГТУ, 2019. - 52 с. - ISBN 978-5-7782-3939-5. - Текст : электронный.
- URL: <https://znanium.com/catalog/product/1866926>
7. Иванова, Е. А. Управление требованиями к бизнес-приложениям : учебное пособие / Е. А. Иванова, Н. В. Ефанова. — Краснодар : КубГАУ, 2019. — 130 с. — ISBN 978-5-907294-16-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/254207>
8. Коваленко, В. В. Проектирование информационных систем : учебное пособие / В.В. Коваленко. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2021. — 357 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/987869. - ISBN 978-5-00091-637-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/987869>
9. Кугаевских, А. В. Проектирование информационных систем. Системная и бизнес-аналитика : учебное пособие / А. В. Кугаевских. - Новосибирск : Изд-во НГТУ, 2018. - 256 с. - ISBN 978-5-7782-3608-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1867932>
10. Лезин, И. А. Разработка веб-приложений с использованием Spring Boot : учебное пособие / И. А. Лезин, И. В. Лезина. — Самара : Самарский университет, 2023. — 96 с. — ISBN 978-5-7883-1989-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/406517>
11. Макшанов, А. В. Большие данные. Big Data / А. В. Макшанов, А. Е. Журавлев, Л. Н. Тындыкарь. — 4-е изд., стер. — Санкт-Петербург : Лань, 2024. — 188 с. — ISBN 978-5-507-47346-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/362318>
12. Обоснование и разработка требований к программным системам : учебное пособие / А. А. Бирюкова, А. М. Володина, К. В. Гусев, А. Н. Миронов. — Москва : РТУ МИРЭА, 2022. — 157 с. — Текст :

- электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/240089>
- 13.Официальный сайт ООО «Квартплата 24» [Электронный ресурс]. URL: <https://www.kvp24.ru/>
- 14.Проектирование интерфейса информационных систем : методические указания / составители А. М. Нужный, Н. И. Гребенникова. — Воронеж : ВГТУ, 2022. — 34 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/222746>
- 15.Роль и задачи ux/ui дизайнера в разработке [Электронный ресурс]. URL: <https://it-vacancies.ru/blog/rol-i-zadaci-uxui-dizainera-v-razrabotke/?ysclid=lvzfdsd62r996741711>
- 16.Сайт ПроТестинг [Электронный ресурс]. URL: <http://www.protesting.ru/testing/>
- 17.Чернова П.А., Шобей Л.Г. АКТУАЛЬНЫЕ ПРОБЛЕМЫ ГОСУДАРСТВЕННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ЖИЛИЩНО-КОММУНАЛЬНОГО ХОЗЯЙСТВА // Образование и право. 2022. №6. URL: <https://cyberleninka.ru/article/n/aktualnye-problemy-gosudarstvennoy-informatsionnoy-sistemy-zhilishchno-kommunalnogo-hozyaystva>
- 18.Шитов, В. Н. Организация ресурсоснабжения жилищно-коммунального хозяйства : учебное пособие / В.Н. Шитов. — Москва : ИНФРА-М, 2023. — 309 с. — (Среднее профессиональное образование). — DOI 10.12737/1002912. - ISBN 978-5-16-014757-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1916136>
- 19.Becker P. et al. Applying an improving strategy that embeds functional and non-functional requirements concepts //Journal of Computer Science & Technology. – 2019. – Т. 19.
- 20.Dmitrieva E. Digitalization of the housing and communal services:

development prospects //E3S Web of Conferences. – EDP Sciences, 2021.
– Т. 263. – С. 04037.

21. Hadjitchoneva J. Efficient Automation of Decision-making Processes in Financial Industry: Case study and generalised model //CEUR Workshop Proceedings. – 2019. – Т. 2413.
22. How should you structure your IT team? Examples & organization charts [Электронный ресурс]. URL: <https://www.pdq.com/blog/it-team-structure-and-org-chart/>
23. How to Write a Bug Report That Your Engineers Will Love [Электронный ресурс]. URL: <https://testlio.com/blog/the-ideal-bug-report/>
24. IDEF, EPC и BPMN: как выбрать нотацию для моделирования бизнес-процессов [Электронный ресурс]. URL: <https://practicum.yandex.ru/blog/notacii-modelirovaniya-biznes-processov/>
25. Information System Development and Business Process Maturity: Choosing a Solution [Электронный ресурс]. URL: <https://medium.com/simbirsoft/information-system-development-and-business-process-maturity-choosing-a-solution-fc90f0cf0935>

Приложение А

Модель «Как есть» процесса регистрации показаний

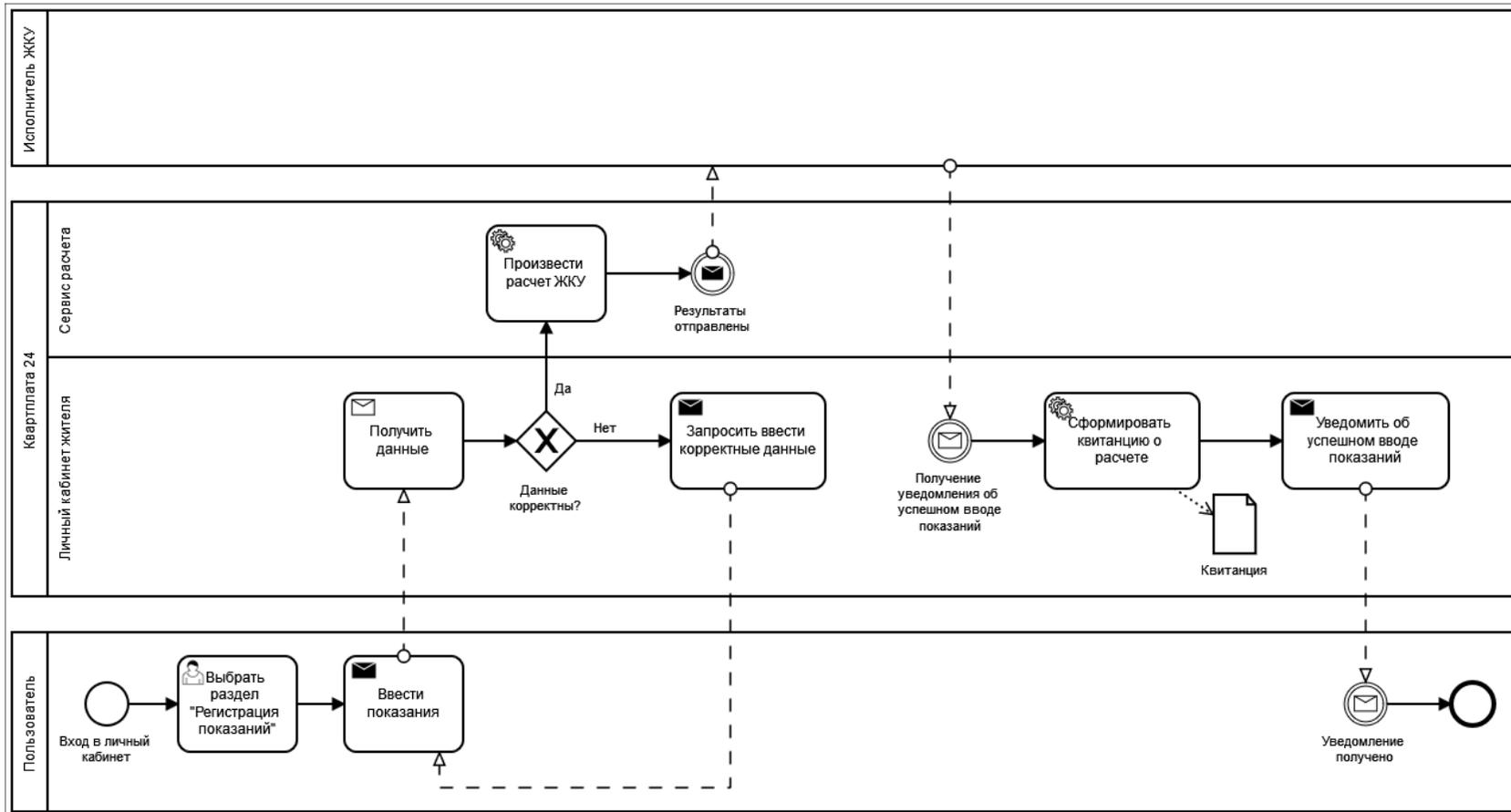


Рисунок А.1 – Модель «Как есть» процесса регистрации показаний