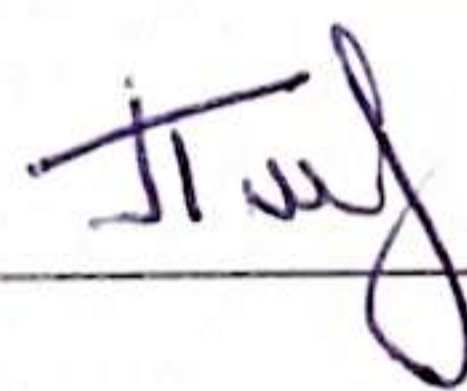


МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

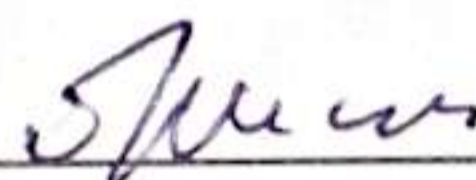
КУРСОВОЕ ПРОЕКТИРОВАНИЕ
Проектирование интеллектуальной системы оценки
доходности аренды жилой недвижимости
Б-ИН-21-169585. 8847-с. 03.000. КП

Студент
гр. БИН-21-1



П.И. Григорьева

Руководитель,
профессор кафедры ИТС,
доцент, д.т.н.



В.М. Гриняк

Владивосток 2025

Аннотация

Отчёт по курсовому проектированию содержит 33 страницы машинописного текста, 10 рисунков, 3 таблицы, 21 источник.

Цель – спроектировать интеллектуальную систему оценки доходности аренды жилой недвижимости для упрощения и оптимизации данного бизнес-процесса.

Работа состоит из введения, 4-х глав, заключения и списка использованных источников.

Первая глава. Анализ и выбор технологий для разработки.

Вторая глава. Разработка технического задания.

Третья глава. Описание структуры данных интеллектуальной системы.

Четвёртая глава. Проектирование интеллектуальной системы и разработка кликабельного прототипа.

В заключении обобщается проделанная работа и подводятся итоги полученных результатов

Содержание

Введение	4
1 Выбор технологий	5
1.1 Выбор языка программирования.....	5
1.2 Выбор библиотеки пользовательского интерфейса	7
1.3 Выбор СУБД.....	13
1.4 Выбор библиотеки для машинного обучения.....	16
2 Разработка технического задания	18
2.1 Цели и требования приложения	18
2.2 Календарный план график	19
2.3 Целевая аудитория.....	20
3 Описание структуры данных.....	22
3.1 Архитектура приложения.....	22
3.1 Реализуемые алгоритмы.....	24
4 Проектирование интеллектуальной системы	25
4.1 Разработка пути пользователя	25
4.2 Атомарный дизайн.....	26
4.3 Разработка кликабельного прототипа.....	28
Список использованных источников.....	32
Приложение А.....	34

Введение

В условиях современного рынка недвижимости, где конкуренция становится все более острой, эффективное управление арендными объектами требует применения инновационных подходов и технологий. Одним из ключевых аспектов успешного управления является оценка доходности аренды жилой недвижимости, которая позволяет собственникам и управляющим компаниям принимать обоснованные решения, минимизировать риски и оптимизировать финансовые потоки.

С развитием информационных технологий и увеличением объемов доступных данных, возникает необходимость в создании интеллектуальных систем, способных анализировать и обрабатывать информацию о рынке недвижимости [1]. Проектирование такой системы требует применения методов машинного обучения, анализа данных и разработки пользовательских интерфейсов, что делает данную задачу актуальной и многогранной.

Целью данной курсовой работы является проектирование интеллектуальной системы оценки доходности аренды жилой недвижимости, которая будет учитывать различные факторы, влияющие на стоимость аренды, такие как местоположение, состояние объекта, площадь, рыночные тенденции и другие параметры. В рамках работы будет разработан алгоритмы для оценки доходности и реализован прототип системы, который позволит компании получать актуальную информацию и рекомендации по управлению арендными объектами.

В ходе выполнения курсовой работы решаются следующие задачи:

- выбор необходимых технологий;
- разработка технического задания для разработки приложения;
- описание структуры данных;
- разработка интерфейса приложения;
- разработка кликабельного прототипа приложения.

Таким образом, данная работа не только способствует углублению знаний в области программной инженерии, но и предоставляет практическое решение для актуальной проблемы в сфере посуточной аренды, что подчеркивает ее значимость и полезность в современных условиях.

1 Выбор технологий

1.1 Выбор языка программирования

В процессе проектирования интеллектуальной системы важным этапом является выбор подходящего языка программирования. В данном разделе рассматриваются три основных кандидата: Python, C++ и Java. Эти языки были выбраны исходя из их популярности, функциональных возможностей и применимости к задачам анализа данных, машинного обучения и разработки приложений [2].

Python – это высокоуровневый язык программирования общего назначения, который широко используется в области анализа данных, искусственного интеллекта и разработки. Он отличается простотой синтаксиса, что делает его доступным даже для начинающих программистов, а также наличием обширной экосистемы библиотек и фреймворков [3].

Преимущества Python:

- простота и читаемость кода. Python имеет лаконичный и интуитивный синтаксис, что позволяет быстро писать и понимать код. Это особенно важно при разработке сложных систем, где требуется высокая степень сотрудничества между разработчиками;

- богатая библиотека для анализа данных и машинного обучения. Python предоставляет мощные инструменты для обработки данных, такие как библиотеки NumPy, Pandas и Scikit-learn. Эти библиотеки позволяют легко манипулировать данными, строить модели машинного обучения и визуализировать результаты;

- поддержка веб-разработки. Фреймворки вроде Django и Flask делают Python отличным выбором для создания веб-интерфейсов и серверных приложений. Интеграция этих инструментов с анализом данных позволяет создавать полноценные решения, доступные через интернет;

- широкая поддержка сообщества. Python обладает одним из самых активных сообществ разработчиков, что обеспечивает постоянную поддержку и обновление библиотек и инструментов.

Недостатки языка программирования Python:

- производительность. Python интерпретируемый язык, поэтому он может уступать по производительности компилируемым языкам, таким как C++ или Java, особенно в вычислительно интенсивных задачах;

- типизация. Динамическая типизация Python может привести к ошибкам времени выполнения, что требует тщательного тестирования и проверки кода.

C++ – это мощный объектно-ориентированный язык программирования, известный своей высокой производительностью и возможностью низкоуровневого доступа к аппаратным ресурсам.

Он часто используется в системах реального времени, играх и высокопроизводительных приложениях [4].

Преимущества языка C++:

- высокая производительность. C++ компилируется в машинный код, что дает ему значительное преимущество в скорости исполнения по сравнению с интерпретируемыми языками, такими как Python;

- низкоуровневое программирование. Возможность прямого управления памятью и другими ресурсами компьютера делает C++ идеальным выбором для разработки систем, требующих максимальной эффективности;

- многофункциональность. C++ поддерживает множество парадигм программирования, включая процедурное, объектно-ориентированное и обобщенное программирование, что позволяет решать широкий спектр задач.

Недостатки C++:

- сложность. Синтаксис C++ значительно сложнее, чем у Python, что увеличивает порог входа для новых разработчиков и усложняет процесс отладки и поддержки кода;

- отсутствие встроенных средств для анализа данных и машинного обучения. Хотя существуют библиотеки для анализа данных и машинного обучения на C++, они менее развиты и популярны, чем аналогичные средства в Python.

Java – это объектно-ориентированный язык программирования, созданный с акцентом на безопасность и переносимость. Он широко используется в корпоративных приложениях, мобильных устройствах и веб-сервисах [5].

Преимущества Java:

- переносимость. Благодаря виртуальной машине Java (JVM), программы, написанные на Java, могут выполняться на различных платформах без необходимости перекомпиляции;

- безопасность. Язык имеет строгую систему типов и механизмы защиты памяти, что снижает вероятность ошибок и уязвимостей;

- масштабируемость. Java хорошо подходит для разработки крупных распределённых систем благодаря поддержке многопоточности и наличию мощных инструментов для управления памятью.

Недостатки Java:

- медленная инициализация. Запуск JVM может занимать значительное время, что делает Java менее подходящим для небольших скриптов и одноразовых задач;

- более сложный синтаксис. По сравнению с Python, синтаксис Java более громоздкий и требует большего количества строк кода для достижения тех же целей.

Таблица 1 – Сравнение языков программирования для разработки

Критерий	Python	C++	Java
Производительность	Низкая	Высокая	Средняя
Простота синтаксиса	Очень высокая	Низкая	Средняя
Машинное обучение	Отличное	Ограниченное	Хорошее
Количество библиотек и доп. инструментов	Большое	Среднее	Среднее

Проанализировав таблицу 1, можно сделать вывод, что наиболее подходящим языком программирования является Python. Он предлагает работу с использованием алгоритмов машинного обучения, лаконичный и понятный синтаксис. Таким образом, Python станет оптимальным инструментом для реализации данного проекта, обеспечивая баланс между производительностью, удобством разработки и доступностью необходимых библиотек и инструментов.

1.2 Выбор библиотеки пользовательского интерфейса

Одним из ключевых этапов проектирования системы является создание удобного и интуитивно понятного пользовательского интерфейса (UI). Для этого необходимо выбрать соответствующую библиотеку, которая позволит эффективно реализовать требуемую функциональность. Рассмотрим несколько библиотек для разработки UI на языке Python.

На сегодняшний день существует множество библиотек для создания графических пользовательских интерфейсов (GUI) на языке Python [6]. Среди наиболее известных и используемых можно выделить следующие:

- Tkinter. Стандартная библиотека для создания GUI в Python. Поставляется вместе с интерпретатором Python и не требует дополнительной установки. Имеет простой и понятный API, что делает её удобной для быстрого создания простых интерфейсов;

- wxPython. Основан на кросс-платформенном инструментарии wxWidgets. Обеспечивает более продвинутые возможности по сравнению с Tkinter, но требует установки дополнительных пакетов. Хорошо подходит для создания сложных и профессиональных интерфейсов;

- PyQt/PySide. Основаны на популярной библиотеке Qt. Предоставляют широкие возможности для разработки сложных интерфейсов, но требуют установки дополнительных зависимостей. Поддерживают создание современных и визуально привлекательных интерфейсов;

– Kivy. Кросс-платформенная библиотека для создания мобильных и настольных приложений. Подходит для разработки интерактивных интерфейсов, но может быть сложноватой для новичков. Ориентирована на создание мультимедийных и игровых приложений;

– Dear PyGui. Относительно новая библиотека, ориентированная на быстрое создание GUI. Предлагает высокую производительность и удобство использования. Идеально подходит для научных вычислений и визуализации данных.

Каждая из этих библиотек имеет свои сильные стороны и предназначена для различных сценариев использования. Однако для данного проекта необходимо выбрать одну из них, основываясь на конкретных требованиях и целях.

Перед тем как перейти к выбору конкретной библиотеки, определим основные требования к ней:

– простота использования: Библиотека должна иметь простой и понятный API, чтобы разработчики могли быстро освоить её и начать использовать;

– кросс-платформенность: Интерфейс должен корректно работать на различных операционных системах (Windows, macOS, Linux);

– наличие базовых элементов управления: Библиотека должна предоставлять стандартные элементы управления (кнопки, текстовые поля, списки и т.д.), необходимые для создания полноценного интерфейса;

– возможность расширения функционала: Должны быть доступны способы добавления дополнительных функций и модулей, если это потребуется в будущем;

– совместимость с библиотеками для анализа данных и машинного обучения: Поскольку наша система будет использовать данные и алгоритмы машинного обучения, важно, чтобы выбранная библиотека поддерживала интеграцию с соответствующими инструментами (например, Pandas, Scikit-learn);

– минимальные зависимости. Желательно, чтобы библиотека имела минимальное количество внешних зависимостей, чтобы упростить процесс установки и настройки.

Теперь рассмотрим каждую из перечисленных выше библиотек в контексте требований нашего проекта.

Библиотека Tkinter является одной из самых популярных и широко используемых библиотек для создания графических пользовательских интерфейсов (GUI) в языке программирования Python. Она предлагает множество преимуществ, которые делают её идеальным выбором как для начинающих, так и для опытных разработчиков.

Одним из главных достоинств Tkinter является его простота. Библиотека имеет очень интуитивно понятный API, что позволяет разработчикам быстро осваивать её и создавать

простые интерфейсы без особых усилий. Это особенно важно для тех, кто только начинает свой путь в программировании, так как позволяет сосредоточиться на логике приложения, а не на сложностях интерфейса.

Библиотека предоставляет все базовые элементы управления, необходимые для большинства приложений. Это включает в себя кнопки, текстовые поля, метки и другие элементы, которые позволяют создавать функциональные и удобные интерфейсы. Благодаря этому разработчики могут легко реализовывать различные функции и взаимодействия в своих приложениях.

Помимо этого, Tkinter также предлагает возможность подключения сторонних библиотек и модулей, что позволяет значительно расширить функциональность приложений. Это означает, что разработчики могут интегрировать дополнительные возможности, такие как работа с базами данных, сетевое взаимодействие и многое другое, что делает приложения более мощными и многофункциональными.

Еще одним важным аспектом является совместимость Tkinter с аналитическими библиотеками, такими как Pandas и NumPy. Это позволяет легко интегрировать графические интерфейсы с мощными инструментами для анализа данных, что особенно полезно для разработчиков, работающих в области науки о данных и анализа информации.

Наконец, стоит отметить, что Tkinter входит в стандартную поставку Python. Это означает, что пользователям не нужно беспокоиться о необходимости установки дополнительных пакетов или зависимостей, что упрощает процесс разработки и развертывания приложений. Это делает Tkinter доступным для всех, кто уже имеет установленный Python, и позволяет сосредоточиться на создании качественного программного обеспечения.

Библиотека wxPython является одной из популярных библиотек для создания графических пользовательских интерфейсов (GUI) в языке программирования Python. Она предлагает множество возможностей, которые могут быть полезны разработчикам, стремящимся создать профессиональные и функциональные приложения. Однако, как и любая другая библиотека, wxPython имеет свои особенности, которые стоит рассмотреть более подробно.

wxPython поддерживается на всех основных операционных системах, таких как Windows, macOS и Linux. Это позволяет разработчикам создавать приложения, которые будут работать на различных платформах без необходимости вносить изменения в код. Однако стоит отметить, что для полноценной работы wxPython может потребоваться установка дополнительных пакетов, что может усложнить процесс настройки для некоторых

пользователей. Тем не менее, кросс-платформенность остается одним из значительных преимуществ этой библиотеки.

Одной из сильных сторон wxPython является богатый набор элементов управления, который она предлагает. Это позволяет разработчикам создавать сложные и профессиональные интерфейсы, которые могут удовлетворить потребности даже самых требовательных пользователей. Благодаря разнообразию доступных элементов управления, таких как кнопки, текстовые поля, списки и другие, разработчики могут легко реализовывать различные функции и взаимодействия в своих приложениях, что делает их более удобными и функциональными.

Библиотека wxPython также предоставляет широкий спектр возможностей для расширения функционала приложений. Это означает, что разработчики могут интегрировать дополнительные модули и библиотеки, что позволяет значительно улучшить производительность и функциональность создаваемых приложений. Возможность расширения функционала делает wxPython привлекательным выбором для тех, кто хочет создать мощные и многофункциональные приложения.

Стоит упомянуть о зависимостях, связанных с использованием wxPython. Для полноценной работы библиотеки может потребоваться установка дополнительных пакетов, что может усложнить процесс настройки для новичков. Это может стать определенным барьером для тех, кто только начинает свой путь в программировании. Тем не менее, для опытных разработчиков это не должно стать серьезной проблемой, так как они обычно знакомы с процессом установки и настройки необходимых зависимостей.

Библиотека PyQt/PySide, как и вышеперечисленные, является одной из самых популярных библиотек для создания графических пользовательских интерфейсов (GUI) на языке программирования Python [7]. Она предлагает разработчикам множество возможностей и инструментов, которые позволяют создавать как простые, так и сложные приложения. Но она имеет свои особенности.

Первым аспектом является простота использования. Хотя PyQt/PySide может показаться более сложной в освоении по сравнению с Tkinter, она все же считается более интуитивно понятной, чем wxPython. Разработчикам потребуется некоторое время, чтобы привыкнуть к особенностям этой библиотеки, но после этого они смогут воспользоваться всеми её преимуществами. Важно отметить, что наличие хорошей документации и активного сообщества пользователей значительно упрощает процесс обучения.

Кросс-платформенность — это ещё одно важное преимущество PyQt/PySide. Библиотека поддерживает все основные операционные системы, такие как Windows, macOS и Linux. Это позволяет разработчикам создавать приложения, которые будут работать на

различных платформах без необходимости вносить изменения в код. Однако стоит учитывать, что для полноценной работы библиотеки может потребоваться установка дополнительных зависимостей, что может усложнить процесс настройки для некоторых пользователей. Тем не менее, кросс-платформенность остаётся одним из значительных достоинств этой библиотеки.

Расширение функционала — ещё одна сильная сторона PyQt/PySide. Библиотека предоставляет гибкость и возможность подключения дополнительных модулей, что позволяет значительно улучшить производительность и функциональность создаваемых приложений. Это особенно важно для разработчиков, которые стремятся создать мощные и многофункциональные приложения, способные удовлетворить потребности пользователей. Возможность расширения функционала делает PyQt/PySide привлекательным выбором для многих разработчиков.

Совместимость с аналитическими библиотеками также является важным аспектом. PyQt/PySide хорошо работает с основными библиотеками для анализа данных, такими как Pandas и NumPy. Это открывает новые горизонты для разработчиков, работающих в области науки о данных и анализа информации. Однако стоит учитывать, что интеграция может потребовать дополнительных настроек, что может стать препятствием для некоторых пользователей. Тем не менее, для опытных разработчиков это не должно стать серьёзной проблемой, так как они обычно знакомы с процессом настройки и интеграции различных библиотек.

Библиотека Kivy — это также один из мощнейших инструментов для разработки графических пользовательских интерфейсов (GUI) на языке Python [8]. Одним из её ключевых преимуществ является кросс-платформенность. Это означает, что приложения, созданные с использованием Kivy, могут работать на всех основных операционных системах, включая Windows, macOS, Linux, а также на мобильных устройствах, таких как Android и iOS. Такая универсальность позволяет разработчикам создавать приложения, которые могут быть использованы на различных устройствах без необходимости вносить изменения в код. Это значительно упрощает процесс разработки и расширяет аудиторию пользователей.

Кроме того, Kivy предлагает широкий набор элементов управления, которые ориентированы на создание интерактивных и мультимедийных приложений. Эти элементы управления включают в себя кнопки, текстовые поля, слайдеры и многие другие компоненты, которые позволяют разработчикам создавать привлекательные и функциональные интерфейсы. Благодаря этому, разработчики могут легко реализовывать различные функции и взаимодействия, что делает приложения более удобными и интуитивно понятными для пользователей. Визуальная привлекательность и интерактивность интерфейсов, созданных с помощью Kivy, могут значительно повысить пользовательский опыт.

Помимо всего, Kivy предоставляет возможность расширения функционала, что является ещё одним важным аспектом этой библиотеки. Разработчики могут добавлять дополнительные функции и модули, что позволяет значительно улучшить производительность и функциональность создаваемых приложений. Это особенно важно для тех, кто стремится создать мощные и многофункциональные приложения, способные удовлетворить разнообразные потребности пользователей. Возможность интеграции сторонних библиотек и модулей делает Kivy гибким инструментом, который может адаптироваться под конкретные задачи и требования проекта.

Библиотека Dear PyGui представляет собой мощный инструмент для создания графических пользовательских интерфейсов (GUI). Одним из её основных преимуществ является простота использования. Dear PyGui предлагает простой и интуитивно понятный API, который во многом схож с известной библиотекой Tkinter. Это делает её доступной даже для тех, кто только начинает свой путь в программировании. Благодаря понятной структуре и логике работы, разработчики могут быстро освоить библиотеку и начать создавать свои приложения без необходимости в длительном изучении сложных концепций.

Кроме того, Dear PyGui предоставляет широкий набор базовых элементов управления, которые подходят для большинства приложений. Эти элементы управления включают в себя кнопки, текстовые поля, выпадающие списки и многие другие компоненты, которые позволяют создавать функциональные и привлекательные интерфейсы. Наличие таких базовых элементов делает разработку приложений более быстрой и эффективной, так как разработчики могут сосредоточиться на логике приложения, а не на создании интерфейса с нуля. Это особенно важно для тех, кто хочет быстро реализовать свои идеи и протестировать их на практике.

Ещё одним значительным преимуществом Dear PyGui является её совместимость с аналитическими библиотеками. Библиотека хорошо интегрируется с основными инструментами для анализа данных, такими как NumPy и Pandas. Это позволяет разработчикам легко визуализировать данные и создавать интерактивные графики, что делает её отличным выбором для приложений, связанных с анализом и обработкой данных. Возможность интеграции с аналитическими библиотеками открывает новые горизонты для создания мощных и функциональных приложений, которые могут удовлетворить потребности пользователей в различных областях.

Для наглядного сравнения библиотек пользовательского интерфейса приведем следующую таблицу 2.

Таблица 2 – Сравнение библиотек пользовательского интерфейса

Критерий	Tkinter	wxPython	PyQt/PySide	Kivy	Dear PyGui
Простота	Очень высокая	Средняя	Ниже средней	Низкая	Высокая
Кросс-платформенность	Отличная	Хорошая	Отличная	Отличная	Хорошая
Элементы управления	Базовые	Продвинутые	Продвинутые	Базовые	Базовые
Расширяемость	Средняя	Высокая	Высокая	Высокая	Средняя
Аналитические библиотеки	Отличные	Хорошие	Хорошие	Сложные	Отличные
Зависимость	Нет	Есть	Есть	Есть	Минимальные

Исходя из проведенного анализа, для нашего проекта наиболее подходящей библиотекой является Tkinter. Эта библиотека соответствует всем вышеперечисленным требованиям. Таким образом, выбор Tkinter для разработки пользовательского интерфейса разрабатываемой интеллектуальной системы является оптимальным решением, учитывая требования и цели.

1.3 Выбор СУБД

Неотъемлемая часть разработки системы – это базы данных (БД). Они позволяют организовывать и структурировать информацию, её и предоставлять доступ к ней для пользователей приложения. Без БД приложение не сможет хранить и обрабатывать данные.

Для работы с БД используются специальные системы управления (СУБД), которые обеспечивают безопасность, целостность и доступность данных, а также позволяют извлекать и анализировать информацию [9]. При разработке они используются для связи приложения с данными, хранящимися в базе, и предоставления доступа к ним [10]. Существует множество различных систем управления базами данных. Рассмотрим самые популярные и выделим их достоинства и недостатки.

1) MySQL;

Одна из самых популярных и распространенных СУБД в мире, используемых в таких отраслях, как веб-разработка, бизнес-аналитика и т.д. Она реляционная, т.е. информация внутри описывается как набор связей. Распространяется как свободное программное обеспечение и отличается гибкостью, легкостью, удобством в использовании [11].

MySQL универсальна и может работать на различных операционных системах, включая Windows, Linux, Unix и Solaris. Она имеет множество расширений и плагинов, а также API для многих популярных языков программирования, включая Python. Благодаря этому она легко настраивается и может быть адаптирована под индивидуальные потребности каждой программы. Но стоит понимать, что в ней по умолчанию меньше возможностей и функций, чем в некоторых других СУБД. Это может облегчить и упростить изучение, однако для выполнения более сложных задач необходимо устанавливать плагины и расширения.

Среди остальных плюсов можно выделить высокую производительность - MySQL работает быстрее со средними и малыми базами данных, чем другие популярные СУБД. Это связано с ее внутренней структурой. Однако, в больших проектах, может столкнуться с проблемами из-за большого количества одновременных обращений к базе данных.

Справиться с этим помогает продвинутая система управления доступом - MySQL предлагает продвинутую систему контроля доступа для различных пользователей. Она включает шифрование, проверку и другие функции безопасности. Самое главное, что весь этот функционал данная СУБД предоставляет полностью бесплатно и распространяется с открытым исходным кодом, доступным для всех.

Можно сказать, что MySQL — это простое, удобное и бесплатное решение с графическим интерфейсом и базовым набором инструментов для работы с данными. Такая простота удовлетворяет потребности разработчиков, не нагромождая лишних функций. А если они всё же нужны, СУБД обладает возможностью установить плагины, которые необходимы для более глубокой обработки.

2) SQLite;

SQLite — это быстрая и легкая однофайловая система управления базами данных на языке C. Она не имеет сервера и позволяет хранить все данные в локальной базе на одном устройстве. Работа с ней не требует сторонних библиотек или сервисов [12].

База данных состоит из таблиц, связей между ними, индексов и других элементов. В SQLite все они хранятся в одном файле (файле базы данных), который находится на том же устройстве, где и приложение. Благодаря такой архитектуре, СУБД работает очень быстро, занимает мало места (менее 400 КБ со всеми параметрами) и не требует сложной настройки или длительной установки. Для большинства задач даже не требуются дополнительные компоненты.

SQLite работает на разных платформах (Windows, Linux, Android) и находится в публичном доступе. Как и MySQL, он имеет открытый исходный код и распространяется бесплатно. Однако, и у данной СУБД есть свои недостатки.

Из-за особенностей своей архитектуры, она не поддерживает хранимые процедуры из-за особенностей своей архитектуры и имеет ограниченную поддержку типов данных. К тому же, в SQLite ограничена многопоточная обработка, т.е. только один процесс может записывать данные одновременно. Это делает его непригодным для многопользовательских приложений или приложений, которые записывают большие объемы данных [13].

В общем и целом, можно сказать, что SQLite — это самая простая база данных, которая будет работать на любом устройстве. Все данные в SQLite хранятся в одном файле — таблицы, служебные поля, связи и всё остальное. С одной стороны, это упрощает работу с базой и позволяет легко переносить данные из одного места в другое. Но с другой, лишает разработчика многих важных функций [14].

3) PostgreSQL;

Наряду с MySQL, является одной из самых популярных СУБД в мире. PostgreSQL объектно-реляционная, что означает поддержку концепций, присущих как реляционным базам данных, так и объектно-ориентированным языкам программирования (объекты, классы, наследование и другие) [15].

Эта СУБД является кросс-платформенной, и хорошо работает со многими популярными операционными системами, включая различные дистрибутивы Linux, macOS, Windows и Solaris. Обладает отличной масштабируемостью, содержит большой функционал и поддерживает множество различных типов и структур данных, включая сетевые адреса, текстовые данные JSON и геометрические координаты для геопозиционирования. Все эти форматы могут быть сохранены и обработаны в этой СУБД. Также PostgreSQL является бесплатным и открытым исходным кодом под лицензией BSD, что означает, что его можно использовать бесплатно, модифицировать и распространять без каких-либо ограничений.

Из-за большого количества функционала, PostgreSQL может быть сложнее в настройке и установке по сравнению с некоторыми другими СУБД, особенно для начинающих пользователей. Она также может иметь более длительное время отклика по сравнению с некоторыми другими системами из-за его более сложной структуры и возможностей, которые, в свою очередь, могут требовать больше аппаратных ресурсов, по сравнению с более простыми СУБД.

PostgreSQL на фоне остальных может показаться слишком сложной, однако это мощная, гибкая и постоянно развивающаяся система, которая подойдет разработчикам из любой сферы [16]. И в отличие от крупных коммерческих решений, PostgreSQL остаётся бесплатной и с большим функционалом.

Перед тем, как сделать выбор между СУБД, стоит сравнить их между собой. Сравнение популярных систем управления базами данных представлено в таблице 3.

Таблица 3 – Сравнение СУБД

Критерий	MySQL	SQLite	PostgreSQL
Тип СУБД	Реляционная	Реляционная	Объектно-реляционная
Производительность	Высокая	Высокая	Средняя
Кроссплатформенность	Высокая	Высокая	Высокая
Количество поддерживаемых плагинов	Большое	Маленькое	Большое
Способ распространения	Открытая лицензия	Открытая лицензия	Открытая лицензия
Количество функций	Среднее	Маленькое	Большое
Поддержка триггеров и процедур	Есть	Нет	Есть

Проанализировав таблицу 3, можно сказать, что MySQL является лучшим выбором для разработки. Она предлагает высокую производительность, масштабируемость и широкий спектр функций. Также, стоит упомянуть что она распространяется по модели открытой лицензии и имеет открытый исходный код.

Все это делает MySQL отличным выбором для разработчиков, ищущих надежную, мощную и бесплатную систему управления базами данных.

1.4 Выбор библиотеки для машинного обучения

В период проектирования ключевую роль играет выбор библиотеки для машинного обучения. Ведь именно эти технологии позволят анализировать большие объемы данных и формировать точные прогнозы, на основе которых можно принимать важные бизнес-решения [17]. Стоит разобрать два ведущих инструмента для машинного обучения: Scikit-learn и TensorFlow/Keras, а также проанализировать, что лучше всего подойдет для текущего проекта.

Scikit-learn – это наиболее популярная библиотек для машинного обучения на Python, что включает в себя множество алгоритмов классификации, регрессии, кластеризации и уменьшения размерности [18]. Известна своей простотой использования и эффективностью для решения задач среднего уровня сложности.

TensorFlow/Keras менее известная, но все еще мощная библиотека для глубокого обучения, разработанная Google. Keras — высокоуровневая библиотека, работающая поверх TensorFlow, которая упрощает создание и обучение моделей. Обе библиотеки известны своими возможностями для построения глубоких нейронных сетей и решения сложных задач машинного обучения.

Теперь следует более подробно изучить обе библиотеки.

Библиотека Scikit-learn известна своим интуитивно понятным API, что делает её доступной даже для тех, кто только начинает знакомиться с миром машинного обучения. Легкие в использовании примеры, предоставляемые библиотекой, позволяют быстро освоить основные концепции и начать применять их на практике.

Кроме того, Scikit-learn поддерживает множество классических алгоритмов, необходимых для решения различных задач. В библиотеке представлены такие алгоритмы, как линейная регрессия, деревья решений и случайные леса. Это делает её универсальным инструментом для разработчиков, которые могут использовать её для решения широкого спектра задач, от простых до более сложных. Эффективность Scikit-learn в задачах прогнозирования также заслуживает внимания: она демонстрирует высокую точность для задач среднего уровня сложности, что делает её подходящей для многих проектов, включая наш.

С другой стороны, TensorFlow и Keras представляют собой сильные инструменты для глубокого обучения, но их освоение может усложнить процесс. TensorFlow, как правило, требует больше времени на изучение из-за своей не простой структуры и многообразия возможностей. Однако Keras, работающая поверх TensorFlow, значительно упрощает процесс работы с этой библиотекой, предоставляя более удобный интерфейс. Тем не менее, даже с Keras пользователям всё равно потребуется время, чтобы полностью освоить все функции и возможности, которые предлагают эти инструменты.

Основное предназначение TensorFlow и Keras — это глубокое обучение, что делает их идеальными для решения сложных задач, требующих значительных вычислительных ресурсов. Хотя возможно применение и классических методов, таких как линейная регрессия. Поэтому, если задача не требует глубокого обучения, использование TensorFlow/Keras может оказаться избыточным и неэффективным.

Исходя из вышперечисленного, для проекта наиболее подходящей библиотекой является Scikit-learn. Она отвечает необходимым требованиям. Выбор Scikit-learn для разработки является оправданным и целесообразным шагом, позволяющим эффективно внедрить методы машинного обучения в интеллектуальную систему оценки доходности аренды жилой недвижимости.

2 Разработка технического задания

2.1 Цели и требования приложения

Целью интеллектуальной системы является предоставление пользователям возможности автоматической оценки потенциальной доходности от посуточной сдачи жилья в аренду в режиме реального времени, что будет способствовать оптимизации бизнес-процессов предприятия.

Задачи для разработки приложения:

1. Изучение предметной области;
2. Анализ существующих решений;
3. Анализ, сравнение и выбор средств разработки приложения;
4. Создание пути пользователя и разработка интерфейса;
5. Разработка прототипа;
6. Проектирование архитектуры;
7. Программная разработка;
8. Тестирование приложения.

К основным целям разрабатываемого приложения с точки зрения заказчика можно отнести оптимизацию работы сотрудников, что приведет к уменьшению трудозатрат.

К основным целям разрабатываемого приложения с точки зрения владельцев недвижимости можно отнести повышение точности расчетов предполагаемой прибыли с их объектов.

Рекомендуемыми техническими инструментами для реализации системы являются язык программирования Python, библиотека Tkinter для реализации пользовательского интерфейса, СУБД MySQL и библиотека Scikit-learn для машинного обучения модели.

Интеллектуальная система решает ряд нескольких проблем. Во-первых, сложность точного прогнозирования доходности от сдачи в аренду жилой недвижимости посуточно. Задача системы – предоставить удобную и информативную программу, в которой можно было бы узнать прогнозируемый доход в разрезе года.

И во-вторых, низкая эффективность существующих бизнес-процессов. Многие компании до сих пор используют ручные методы учета и планирования, что приводит к избыточным затратам времени и средств. Кроме того, отсутствие автоматизации часто ведет к ошибкам и недочетам, которые могут негативно сказаться на общей эффективности предприятия.

Основная функция интеллектуальной системы заключается в предоставлении пользователям возможности автоматически оценивать потенциальную прибыль от сдачи

жилья в аренду в режиме реального времени. Эта оценка должна учитывать различные факторы, такие как сезонность, местоположение объекта, состояние и другие параметры, влияющие на стоимость аренды.

Открывая интеллектуальную систему, пользователя встречает краткое описание сервиса. Из приветственного окна к основному функционалу должна вести кнопка. Следом пользователь попадает на главный экран приложения. В нем нужно непосредственно вводит данные об объекте недвижимости, позволяющие сделать расчет максимально точным, а именно: местоположение с обязательным указанием улицы, площадь квартиры, состояние квартиры следует выбрать из выпадающего списка и также нужно указать процент прибыли собственника от выручки.

В результате в следующем окне будет выведен график изменения динамики прибыли ежемесячно. Помимо этого, будет отображаться основная информация о доходности объекта недвижимости. Также у пользователя должна быть возможность просмотра более детального отчета, включающего статистику о доходи и прибыли каждой из сторон по месяцам и итоговую информацию по годовой доходности.

В системе к тому же должна быть предусмотрена функция возвращения на главный экран из каждого окна.

Приложению необходим современный и удобный дизайн. Он должен быть дружелюбным для пользователя и интуитивно понятным. Все элементы интерфейса должны быть выполнены в едином стиле, чтобы приложение выглядело профессионально и аккуратно.

Поддерживаемые языки – русский.

Результат должен быть представлен в виде работающего приложения, полностью удовлетворяющего требованиям заказчика. Сдача и приемка производится поэтапно, в соответствии с этапами разработки приложения, утвержденным с руководителем проекта.

2.2 Календарный план график

На рисунке 1 приведён календарный план график разработки проекта, который был разработан в соответствии с поставленными задачами.

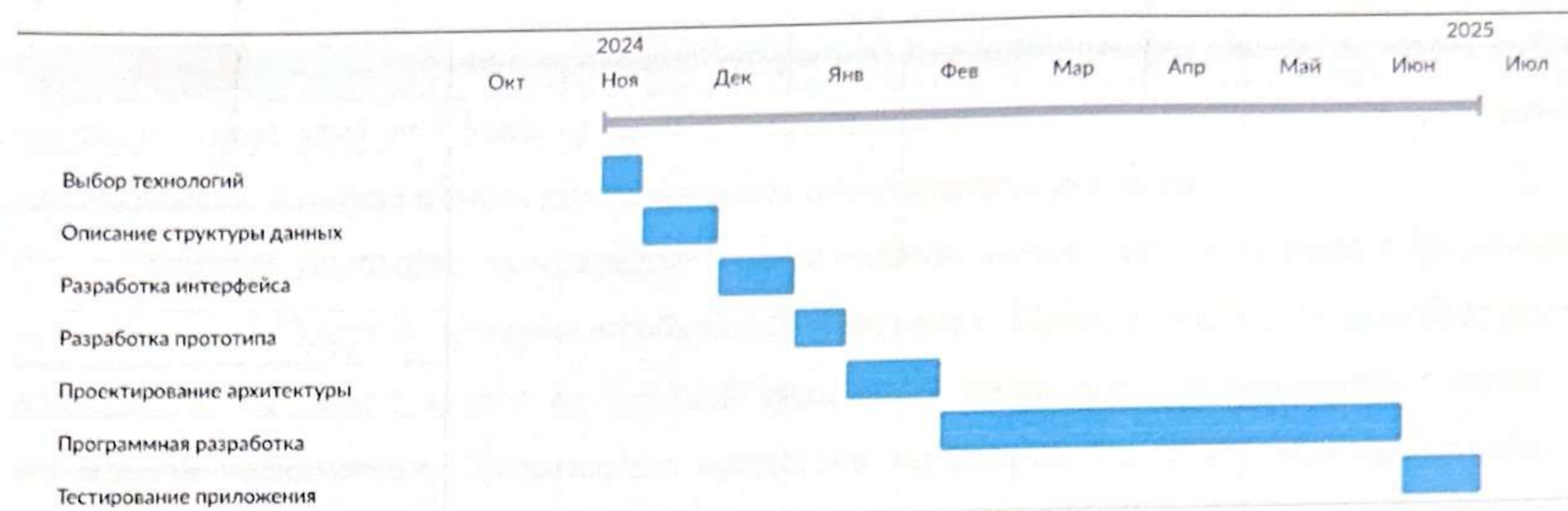


Рисунок 1 – Календарный план график разработки интеллектуальной системы

Немного разберем каждый пункт в диаграмме Ганта:

- выбор технологий - анализ, сравнение и выбор средств разработки приложения. Данный этап проходит с 11 по 21 ноября 2024 года;
- описание структуры данных: архитектура приложения и реализуемые алгоритмы. Данный этап проходит с 22 ноября по 14 декабря 2024 года;
- разработка интерфейса – разработка пути пользователя и дизайна страниц сервиса. Данный этап проходит с 15 декабря 2024 года по 3 января 2025 года;
- разработка прототипа – создание кликабельного прототипа. Данный этап проходит с 4 по 10 января 2025 года;
- проектирование архитектуры – описание будущей структуры приложения и проектирование базы данных. Данный этап проходит с 11 января по 10 февраля 2025 года;
- программная разработка – создание приложения. Данный этап проходит с 11 февраля по 12 июня 2025 года;
- тестирование – проверка функциональности, безопасности и юзабилити. Данный этап проходит с 13 по 27 июня 2025 года.

Общий временной период разработки – с 11 ноября 2024 года по 27 июня 2025 года.

2.3 Целевая аудитория

Целевой аудиторией интеллектуальной системы оценки доходности аренды жилой недвижимости являются компании, специализирующиеся на управлении объектами и сдаче в аренду жилой недвижимости на короткий срок. Эти компании стремятся максимизировать свою прибыль, минимизировать риски и повысить эффективность управления своим портфелем недвижимости. Рассмотрим подробно, кто входит в эту целевую аудиторию и какие потребности у них удовлетворяются системой.

Компании, работающие в сегменте посуточной аренды, представляют собой основную целевую аудиторию системы.

В условиях постоянно меняющегося рынка, где конкуренция растет, а потребности клиентов становятся все более разнообразными, такие компании нуждаются в надежных инструментах, которые помогут им принимать обоснованные решения.

Целевые компании, заинтересованные в использовании системы, имеют несколько основных потребностей, которые необходимо учитывать. Прежде всего, это максимизация доходности. Основной запрос от целевой аудитории заключается в получении точной и актуальной информации. Оптимизация процессов также является ключевым требованием. Компании нуждаются в автоматизации рутинных задач. Это освобождает ресурсы для стратегического управления и позволяет сосредоточиться на более важных аспектах бизнеса, таких как развитие и расширение портфеля недвижимости. Компаниям необходима возможность оперативного доступа к подробной аналитике. Точность прогнозов также играет важную роль. Это позволит компаниям более уверенно планировать свои действия и минимизировать риски.

Интеллектуальная система оценки доходности предоставляет уникальное предложение для целевой аудитории.

3 Описание структуры данных

3.1 Архитектура приложения

Архитектура приложения интеллектуальной системы оценки доходности аренды жилой недвижимости представляет собой многослойную структуру, состоящую из нескольких компонентов, взаимодействующих друг с другом для обеспечения бесперебойной работы системы. Основная цель архитектуры – это обеспечить удобство использования, производительность и масштабируемость системы, а также облегчить процесс внесения изменений и доработок в будущем.

Общая структура системы состоит из четырех основных слоев (рис. 2):

- пользовательский интерфейс (UI), который отвечает за взаимодействие с пользователем, позволяя ему вводить данные о квартире и просматривать результаты расчетов;
- бизнес-логика (Business Logic) содержит основные алгоритмы расчета доходности и прогнозирования на основе введенных данных;
- модуль хранения данных (Data Storage) управляет хранением и извлечением данных, включая информацию о квартирах, расчёты и прогнозы;
- модуль машинного обучения (Machine Learning Module) занимается подготовкой и тренировкой моделей, а также применением их для прогнозирования доходности.

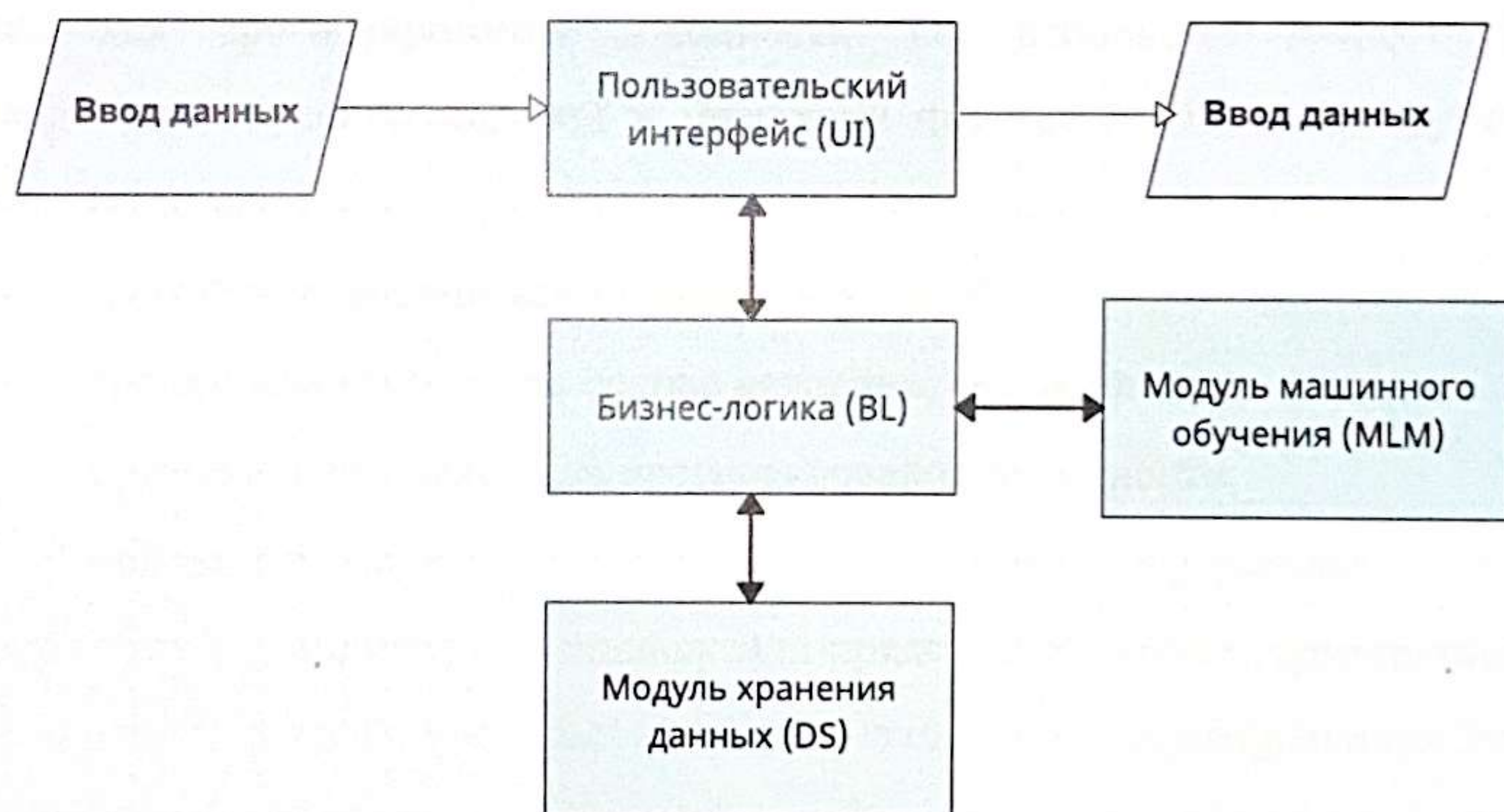


Рисунок 2 – Архитектурная диаграмма модулей

Пользовательский интерфейс предназначен для взаимодействия пользователя с системой. Он предоставляет формы для ввода данных о квартире, таких как адрес, площадь, состояние и процент собственника от дохода. Также UI отображает графики динамики доходов за год и таблицы с подробными расчётами. Пользовательский интерфейс построен с учётом удобства использования и доступности для людей без глубоких знаний в области информационных технологий.

Бизнес-логика включает в себя алгоритмы, отвечающие за обработку введённых данных и выполнение основных вычислений. Она принимает данные от пользовательского интерфейса, обрабатывает их согласно установленным правилам и передает результаты в модуль хранения данных и модуль машинного обучения. Основные функции бизнес-логики включают:

- проверку корректности введённых данных;
- расчёт доходности на основе введённой информации;
- формирование графиков и таблиц для отображения результатов.

Модуль хранения данных отвечает за сохранение и извлечение информации, необходимой для работы системы. Он использует реляционную базу данных для хранения информации о квартирах, расчётов и прогнозов. Основные функции модуля хранения данных:

- хранение информации о квартирах;
- сохранение результатов расчётов и прогнозов;
- извлечение данных для формирования отчётов и графиков.

Модуль машинного обучения отвечает за подготовку и тренировку моделей, а также их применение для прогнозирования доходности. Он использует датасет предыдущих бронирований для обучения моделей и генерации прогнозов. Основные функции модуля машинного обучения:

- подготовка данных для тренировки моделей;
- тренировка моделей на основе исторических данных;
- применение моделей для прогнозирования доходности;
- обновление моделей по мере поступления новой информации.

Таким образом, архитектура приложения представляет собой гармоничное сочетание различных уровней, каждый из которых выполняет свою уникальную функцию. Это позволяет создать мощную и эффективную систему, которая не только облегчает процесс оценки доходности аренды, но и предоставляет пользователям ценные аналитические инструменты для принятия обоснованных решений.

3.2 Реализуемые алгоритмы

Важной частью проектирования интеллектуальной системы являются реализуемые алгоритмы, которые обеспечивают обработку данных и прогнозирование доходности. Эти алгоритмы играют ключевую роль в функционировании приложения, так как именно они позволяют преобразовывать введенные пользователем данные в полезную информацию, которая может быть использована для принятия решений.

Основными алгоритмами, используемыми в системе, являются: алгоритм расчёта доходности, который используется для определения предполагаемой доходности квартиры на основе введённых данных и алгоритм прогнозирования, который основан на машинном обучении и используется для прогнозирования доходности на основе исторических данных.

Алгоритм расчёта доходности определяет ожидаемую прибыль от аренды квартиры на основе введённых данных. Основными шагами алгоритма являются:

1. Сбор данных о квартире (адрес, площадь, состояние);
2. Определение средней стоимости аренды аналогичных квартир в районе;
3. Учёт процента собственника от дохода;
4. Расчёт ожидаемого дохода за определённый период.

Основной расчет дохода собственника выглядит в виде формулы (1):

$$\text{Доход} = \text{Стоимость аренды} \times \left(\frac{\text{Процент собственника}}{100} \right), \quad (1)$$

где Доход — доходность квартиры за определенный период, Стоимость аренды — стоимость от сдачи квартиры в аренду за определенный период, Процент собственника — процент собственника от дохода, оговоренный при заключении договора.

Алгоритм прогнозирования использует методы машинного обучения для прогнозирования доходности на основе исторических данных о бронировании. Основные шаги алгоритма:

1. Подготовка данных, которая подразумевает очистку и преобразование данных для использования в моделях машинного обучения;
2. Тренировка модели, а именно использование исторических данных для обучения модели;
3. Применение модели: использование обученной модели для прогнозирования доходности на будущее.

Архитектура приложения и реализованные алгоритмы представляют собой комплексную систему, предназначенную для оценки доходности аренды жилой недвижимости. Четкая структура приложения обеспечивает удобство использования и надёжность. Реализуемые алгоритмы позволяют выполнять точные расчёты и прогнозы, что помогает руководителям компаний принимать обоснованные решения о сдаче квартир в аренду.

4 Проектирование интеллектуальной системы

4.1 Разработка пути пользователя

Путь пользователя — это последовательность шагов, которые пользователь выполняет, взаимодействуя с веб-сайтом, приложением или любым другим интерфейсом. Он помогает понять, какие трудности или проблемы могут возникнуть у пользователей на каждом этапе и как их можно устранить.

Учитывая это, разработчики и дизайнеры могут оптимизировать интерфейс и улучшить пользовательский опыт, выявить проблемные моменты и барьеры, с которыми пользователи могут столкнуться, и внести соответствующие изменения. Если путь пользователя продуман, то приложение будет более удобным и интуитивно понятным для пользователей, что в свою очередь может привести к увеличению их числа и повышению уровня удовлетворенности продуктом [19].

Путь пользователя можно представить различными способами, используя нотации для визуализации. Нотация Task Flow является одним из таких методов визуализации, которая направлена на простое и понятное представление логики пользовательского взаимодействия. Она сосредоточена на конкретных задачах, которые пользователь выполняет в приложении или системе, что позволяет более детально и точно определить шаги, необходимые для выполнения каждого действия.

При разработке пути пользователя, необходимо обеспечить простую и интуитивно понятную навигацию. Важно, чтобы пользователь мог легко понять, как переходить между страницами и возвращаться к предыдущим. На рисунке 3 показана легенда и основной путь пользователя в нотации Task Flow.

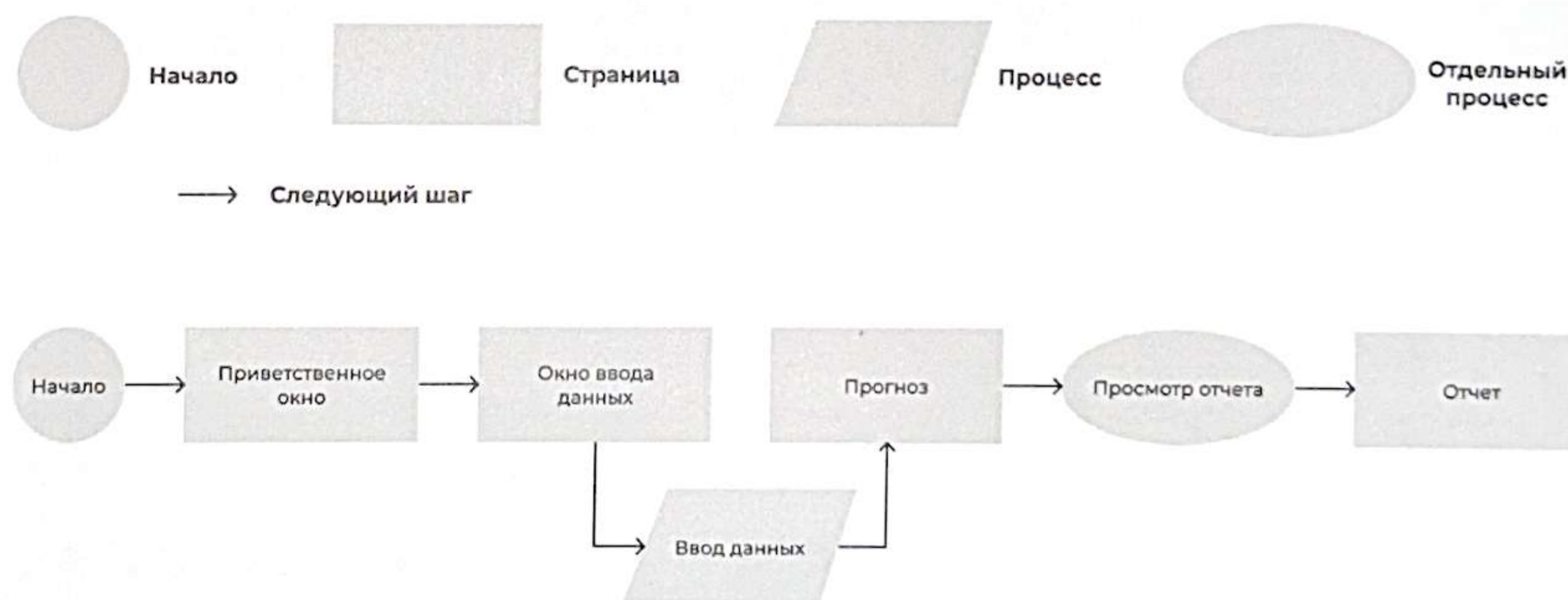


Рисунок 3 – Основной путь пользователя в нотации Task Flow

В приложении будет три основных страницы. Окно ввода данных, на котором соответственно нужно вводить необходимую информацию, а именно: адрес с обязательным указанием улицы, площадь квартиры, состояние квартиры следует выбрать из выпадающего списка и процент прибыли собственника от дохода.

Из окна ввода данных пользователь уже попадает на страницу с прогнозом дохода. На ней будет отображаться график изменения динамики прибыли по месяцам. Помимо этого, выводится основная информация о доходности объекта недвижимости.

Также присутствует страница с возможностью просмотра более открытого отчета со статистикой о доходах и прибыли каждой из сторон и итоговой информации по годовой доходности.

Разработка пути пользователя является важным этапом, направленным на обеспечение интуитивного и эффективного взаимодействия с системой. Этот процесс должен быть простым и удобным для пользователя, что позволит ему оперативно получать необходимую информацию и принимать правильные решения.

4.2 Атомарный дизайн

После разработки пути пользователя необходимо создать интерфейс будущего приложения. Для упрощения процесса, будем придерживаться методологии «атомарный дизайн». В нём разработка продукта начинается не с макетов страниц, а с более мелких деталей, из которых поэтапно создаются более крупные составляющие. Благодаря этому подходу, разработка сильно ускоряется, а дизайн не засоряется лишними элементами [20].

Перед тем, как начать разработку, стоит определиться с цветовой палитрой. Выбор пал на цвета синих оттенков, т.к. они воспринимаются как одни из самых комфортных цветов [21]. Также есть акцентный ярко-розовый цвет, притягивающий внимание пользователя. Палитра изображена на рисунке 4.

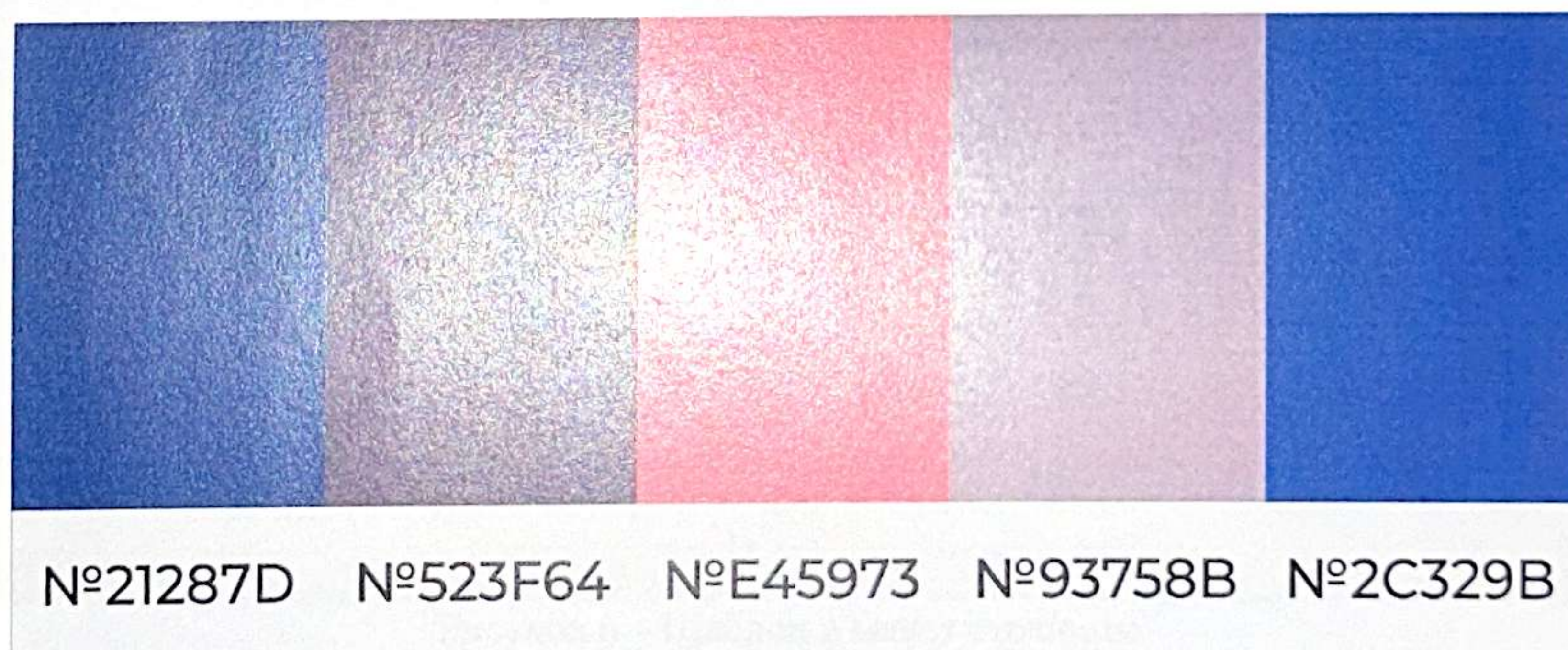


Рисунок 4 – Цветовая палитра

Атомарный дизайн начинается с продумывания атомов, так называемых, простейших элементов. Из них составляются уже более крупные элементы – молекулы. А комбинация атомов и молекул помогает построить организмы – наиболее составные элементы интерфейса. Разработанные части представлены на рисунке 5.

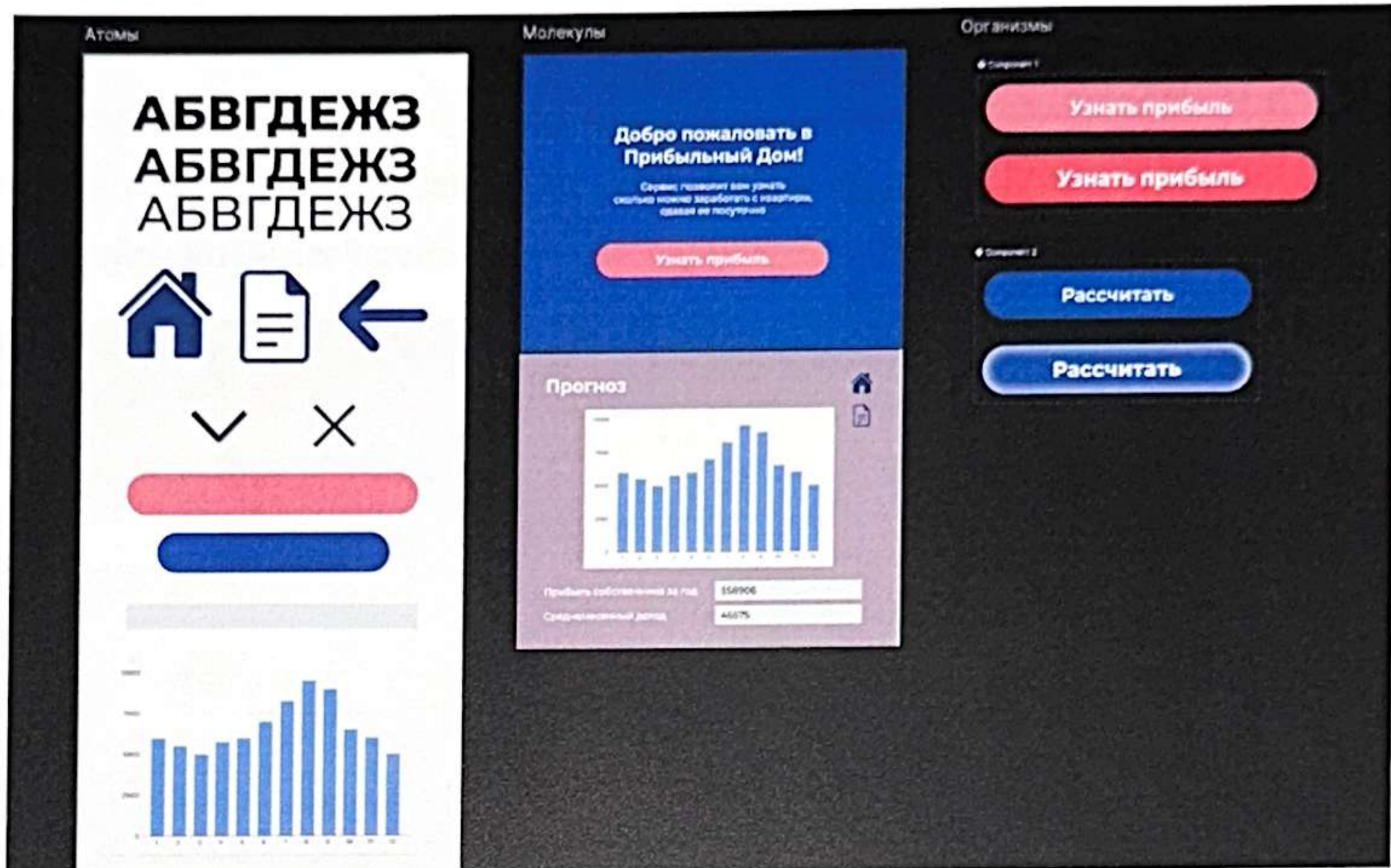


Рисунок 5 – Атомы, молекулы и организмы интерфейса навигатора

Затем начинаются работы над шаблоном и макетом. Именно на шаблоне все элементы собираются в будущий дизайн, а создание макета является заключительной работой над дизайном продукта. На данном этапе необходимо выделить все недочеты и проверить интерфейс на ошибки. Разработанные шаблон и макет страницы прогноза представлены на рисунке 6.

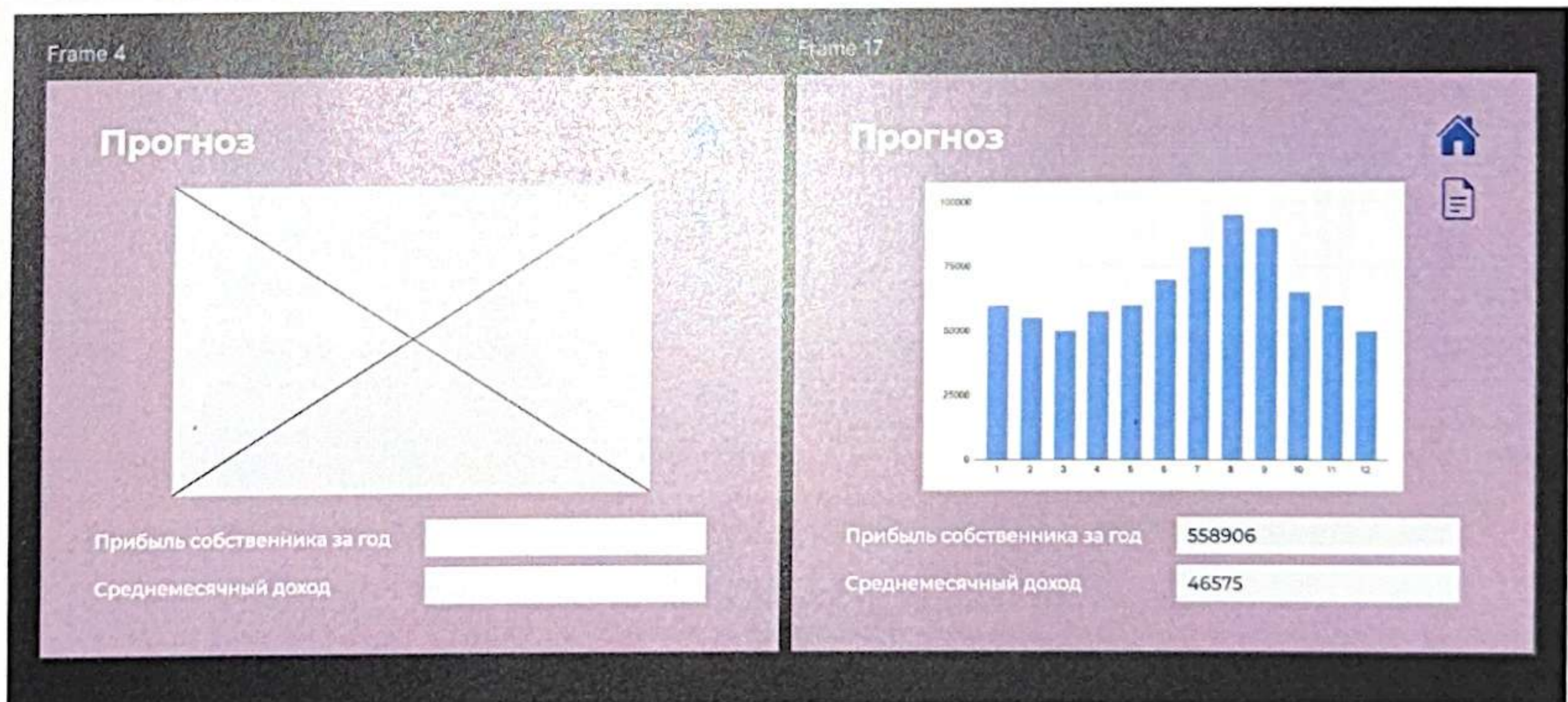


Рисунок 6 – Шаблон и макет страницы

После окончания работ над макетом, можно приступать к разработке прототипа.

4.3 Разработка кликабельного прототипа

Разработка прототипа необходима для визуализации и оценки концепции и дизайна приложения, а также для того, чтобы провести предварительный анализ его привлекательности и usability.

На первом этапе прототипирования необходимо разработать wireframe для всех будущих страниц приложения. Он служит для того, чтобы передать структуру, разметку и навигацию объектов. В основном создается из обычных простых фигур, без картинок. Примеры разработанных wireframes приведены на рисунке 7.

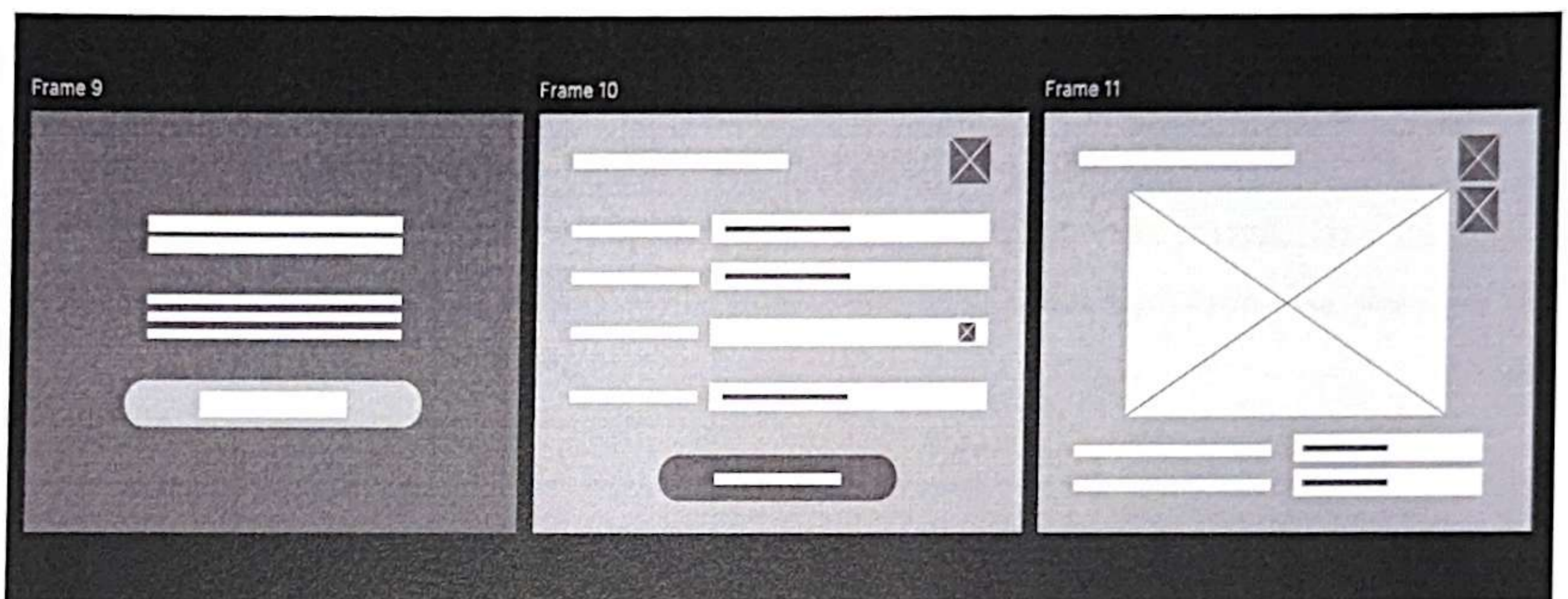


Рисунок 7 – Часть разработанных wireframes

После разработки wireframes, можно начинать разрабатывать mockups. Они уже включают в себя цвет, изображения и типографику. Примеры разработанных mockups приведены на рисунке 8.

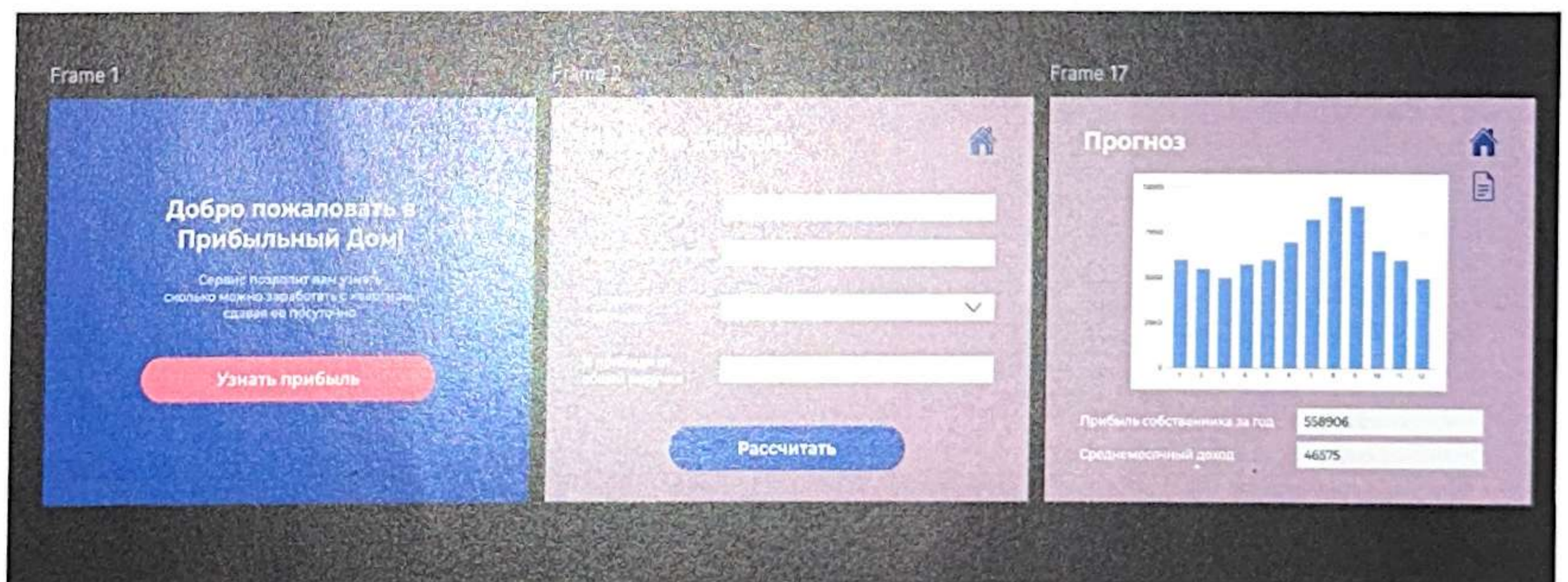


Рисунок 8 – Часть разработанных mockups

На рисунках 7 и 8 приведены wireframes и mockups следующих страниц:

- страница с приветствием;

Из активных элементов на этой странице присутствует только кнопка внизу страницы, при нажатии которой открывается окно для ввода данных.

- страница ввода данных;

Информация будет вводиться в специально отведенные под это окошки. Помимо этого, на странице есть выпадающий список и кнопка, при нажатии на которую пользователь возвращается на главное окно. И кнопка «Рассчитать».

- страница прогноза;

На странице выводится график со статистикой, данные по прибыли, кнопка, возвращающая на главное окно и еще кнопка, открывающая полный отчет с более подробной информацией.

Дизайн готовых wireframes всех страниц интеллектуальной системы приведён на рисунке А.1. в приложении А. И также дизайн готовых mockups всех страниц интеллектуальной системы приведён на рисунке А.2. в приложении А.

После разработки mockups, можно переходить к созданию прототипа. При помощи инструментов сервиса Figma, можно достаточно просто создать прототип. На рисунке 9 показан пример создания одного перехода.

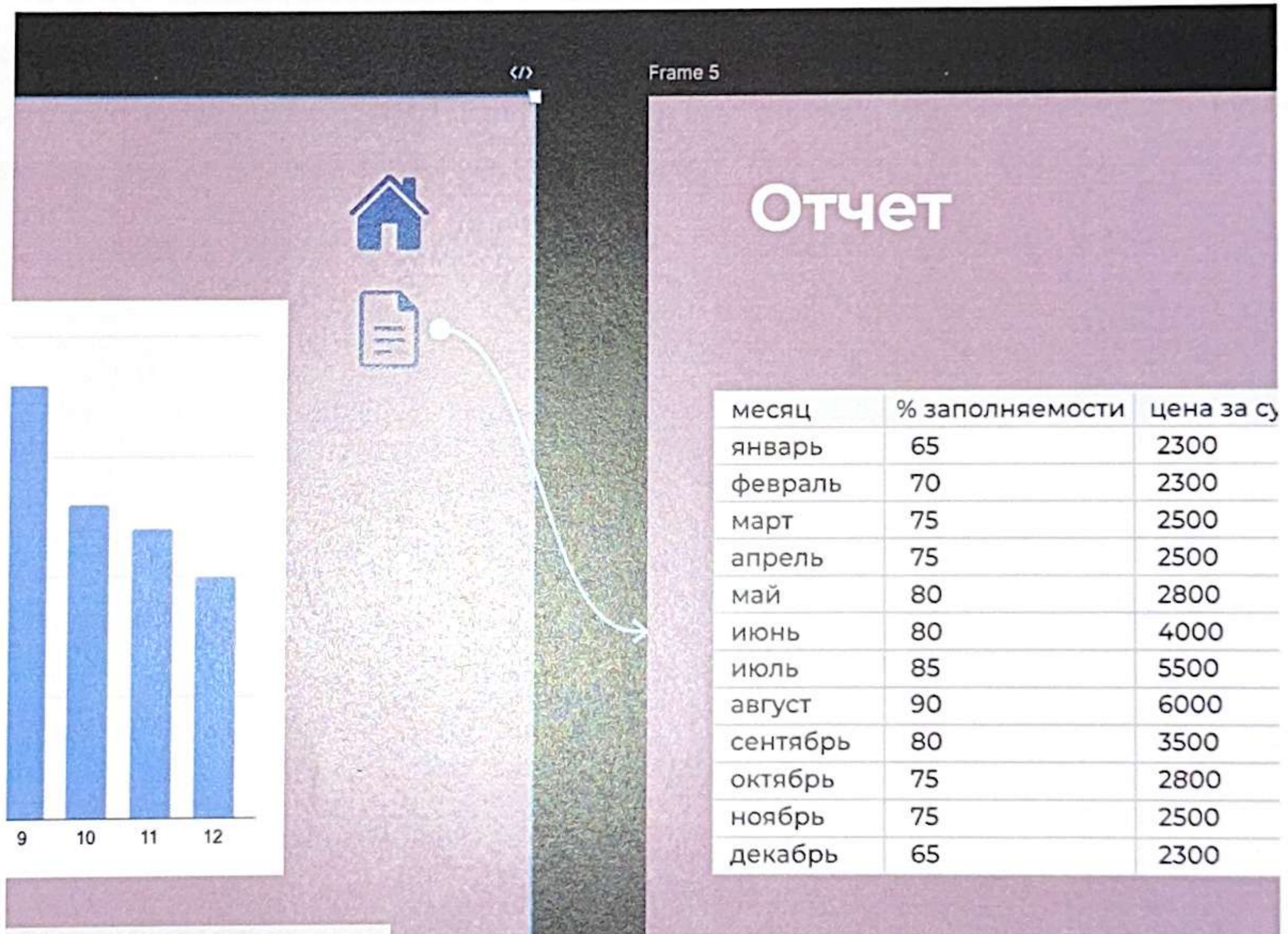


Рисунок 9 – Пример создания прототипа

При нажатии на кнопку «Документ» на одной странице, пользователь переходит на другую, где уже предоставлена полная статистика. Такие связи помогают воссоздать кликабельный прототип в Figma. Готовый результат представлен на рисунке 10.

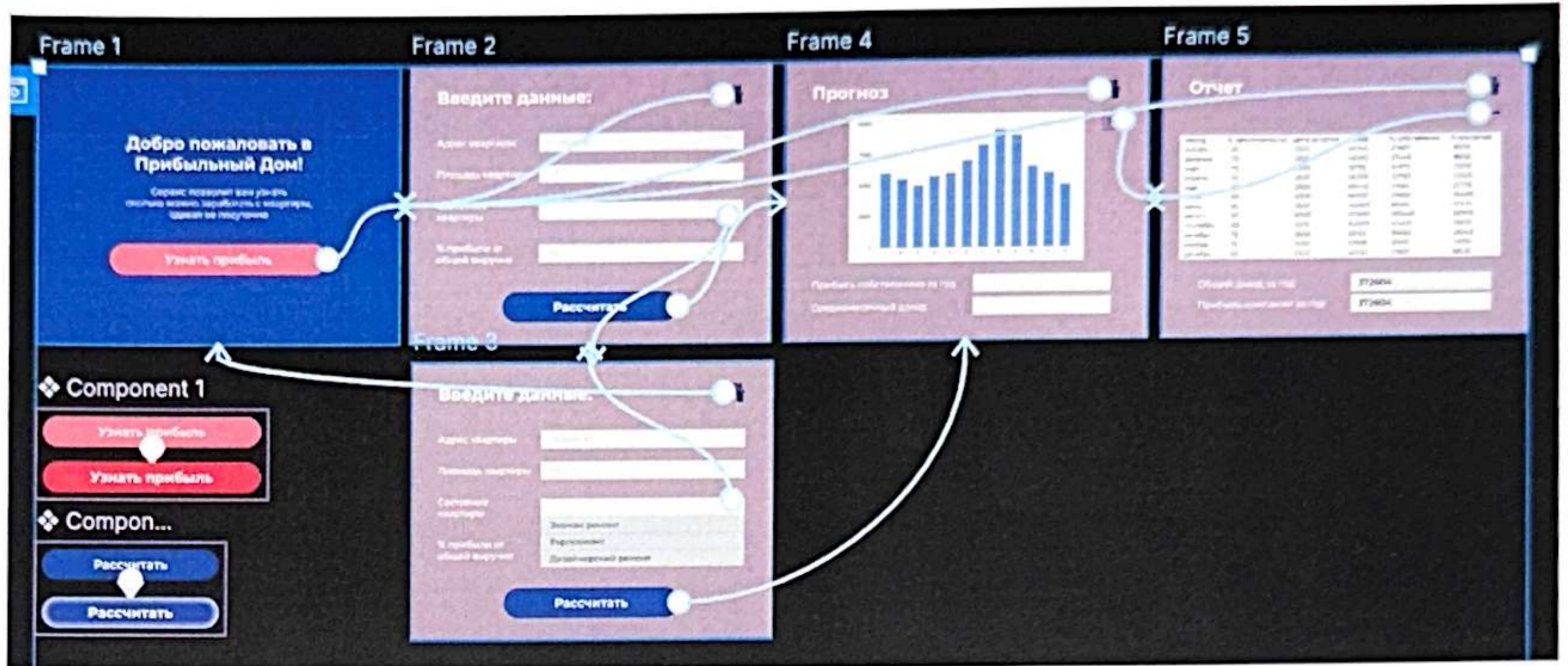


Рисунок 10 – Разработанный кликабельный прототип

Данный прототип может быть использован для получения обратной связи от заказчика, т.е. он может оценить, соответствует ли прототип его ожиданиям и требованиям. Кроме того, прототип может быть протестирован пользователями на предмет удобства использования и выявления потенциальных проблем и улучшений. Этот этап помогает уменьшить риски и повысить шансы на успешное завершение проекта.

Заключение

В процессе курсового проектирования были закреплены знания, навыки и умения в области проектирования информационной системы для предприятий, проанализированы технические средства, разработано техническое задание, описаны структуры данных и разработан интерфейс с последующим созданием прототипа.

Разработанная система успешно сочетает в себе современные технологии анализа данных, машинного обучения и прогнозирования, что позволяет предприятиям принимать обоснованные решения и увеличивать доходность от аренды. Это улучшает качество управления, минимизирует риски и способствует повышению эффективности бизнеса.

Дальнейшая работа над проектом будет направлена на проектирование архитектуры приложения, и на разработку программной части.

Список использованных источников

1. Тенденции мирового ИТ-рынка // Tadviser – URL: https://www.tadviser.ru/index.php/Статья:Тенденции_мирового_ИТ-рынка (дата обращения: 11.11.2024).
2. Python, JavaScript или C++? // Хабр – URL: <https://habr.com/ru/companies/selectel/articles/801765/> (дата обращения: 25.11.2024).
3. Язык программирования Python // Хекслет – URL: <https://ru.hexlet.io/blog/posts/programmirovanie-na-python-osobennosti-obucheniya-perspektivy-situatsiya-na-rynke-truda> (дата обращения: 25.11.2024).
4. C++ в высокопроизводительном и системном программировании // itProger – URL: <https://itproger.com/news/c-v-visokoproizvoditelnom-i-sistemnom-programmirovanii> (дата обращения: 25.11.2024).
5. Язык программирования Java: что нужно знать новичку // Хекслет – URL: <https://ru.hexlet.io/blog/posts/yazyk-programmirovaniya-java-osobennosti-populyarnost-situatsiya-na-rynke-truda> (дата обращения: 25.11.2024).
6. Топ-10 лучших GUI библиотек Python 2023: обзор и сравнение // Дзен – URL: <https://dzen.ru/a/ZQ5zzw6v1Csnubt-> (дата обращения: 26.11.2024).
7. PyQt6 — полное руководство для новичков // Хабр – URL: <https://habr.com/ru/companies/skillfactory/articles/599599/> (дата обращения: 26.11.2024).
8. Kivy // Skillfactory Media – URL: <https://blog.skillfactory.ru/glossary/kivy/> (дата обращения: 27.11.2024).
9. Система управления базами данных // Wikipedia – URL: https://ru.wikipedia.org/wiki/Система_управления_базами_данных (дата обращения: 28.11.2024).
10. Выбор СУБД: шпаргалка, чтобы не запутаться // Хабр. - URL: <https://habr.com/ru/articles/775574/> (дата обращения: 01.12.2025).
11. MySQL: что это за сервер базы данных, пример // Skillfactory media – Честные истории о карьере в IT из реальной среды. - URL: <https://blog.skillfactory.ru/glossary/mysql/> (дата обращения: 01.12.2024).
12. SQLite - что это: обзор на реляционную базу данных с примерами // Skillfactory media – Честные истории о карьере в IT из реальной среды. - URL: <https://blog.skillfactory.ru/glossary/sqlite/> (дата обращения: 02.12.2024).

13. SQLite — самая простая база данных, которая работает везде // Журнал «Код» программирование без снобизма. - URL: <https://thecode.media/sqlite/> (дата обращения: 03.12.2024).
14. SQLite — самая простая база данных, которая работает везде // Журнал «Код» программирование без снобизма. - URL: <https://thecode.media/sqlite/> (дата обращения: 03.12.2024).
15. PostgreSQL: что это за СУБД и как с ней работать // Skillbox Media — журнал для профессионалов. Актуальные статьи про бизнес, дизайн, образование, разработку игр и программирование. - URL: <https://skillbox.ru/media/code/postgresql-vsye-что-nuzhno-znat-dlya-bystrogo-starta/> (дата обращения: 04.12.2024).
16. PostgreSQL: что это, для чего нужна, основы и преимущества - установка и настройка СУБД Postgres, операции // Онлайн-курсы Яндекс Практикум — сервис онлайн-образования в сфере IT, обучение профессиям digital. - URL: <https://practicum.yandex.ru/blog/что-такое-sубд-postgresql> (дата обращения: 04.12.2024).
17. Инструменты и библиотеки для машинного обучения // Skupro – URL: <https://sky.pro/wiki/python/instrumenty-i-biblioteki-dlya-mashinnogo-obucheniya/> (дата обращения: 05.12.2024).
18. Обзор методов классификации в машинном обучении с помощью Scikit-Learn // Tproger – URL: <https://tproger.ru/translations/scikit-learn-in-python> (дата обращения: 06.12.2024).
19. User flow: что это такое и почему без него нельзя делать приложения и сайты // Skillbox Media — журнал для профессионалов. Актуальные статьи про бизнес, дизайн, образование, разработку игр и программирование. - URL: <https://skillbox.ru/media/design/user-flow-basics/> (дата обращения: 15.12.2024).
20. Атомарная дизайн-система: суть, принципы и примеры // Создание и продвижение сайтов в Туле: Разработка и раскрутка от digital-студии Веб-Эталон. - URL: <https://web-etalon.ru/blog/atomarnyy-dizayn/> (дата обращения: 20.12.2024).
21. Психология цвета в дизайне, маркетинге и продажах // Bonnie&Slide – URL: <https://bonnieandslide.com/blog/psihologija-cveta-v-dizajne-marketinge-i-prodazhah> (дата обращения 21.12.2024).

Приложение А

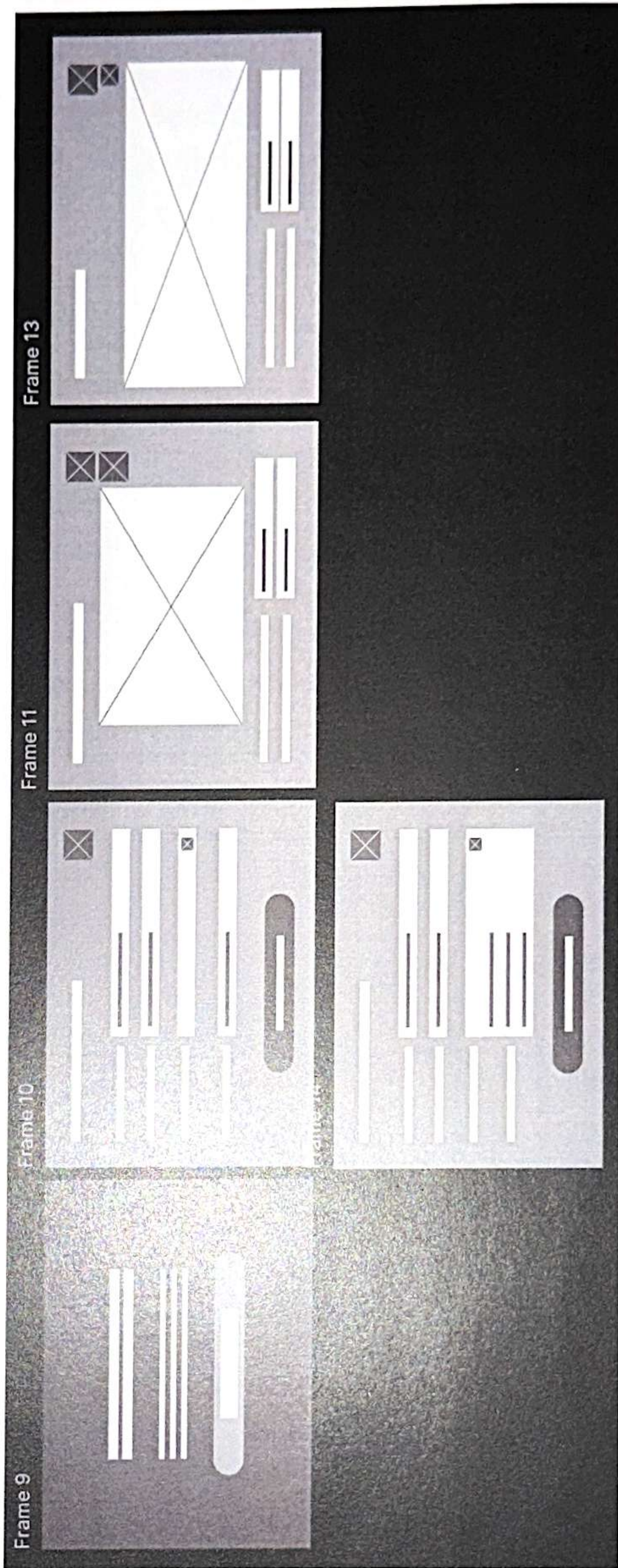


Рисунок А.1 – Разработанный wireframe для интеллектуальной системы

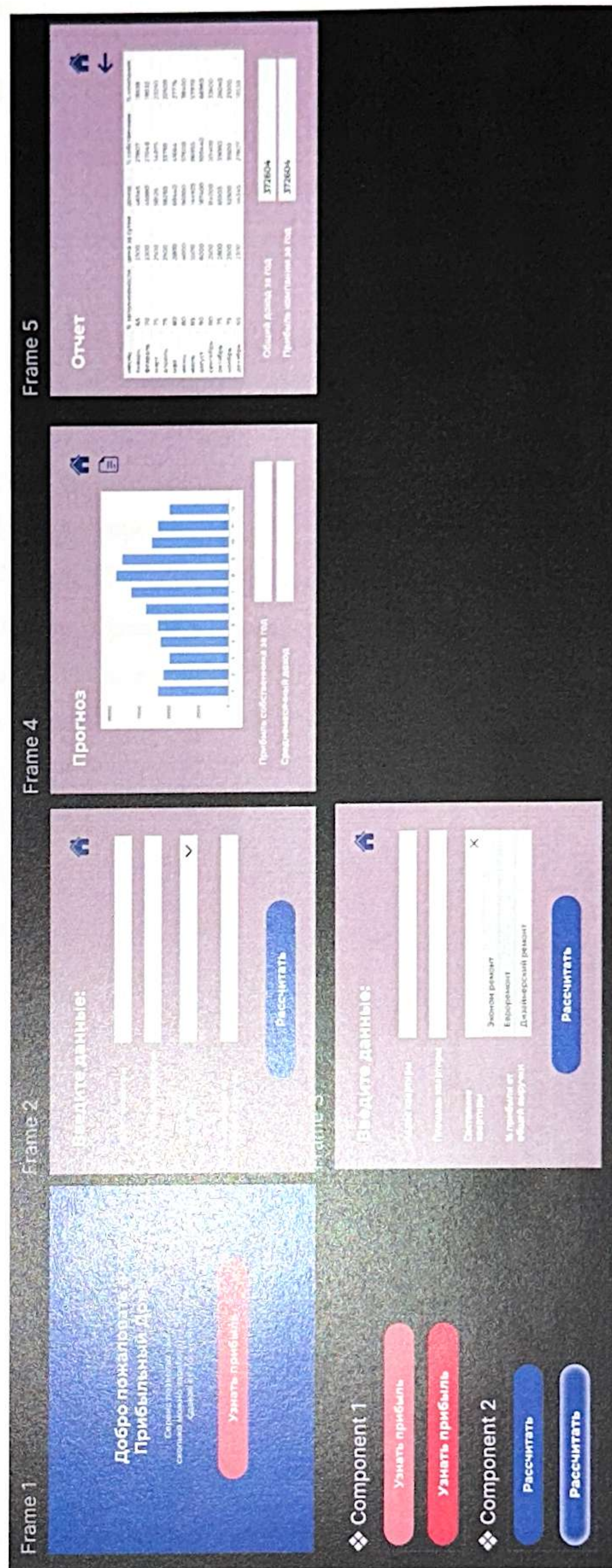


Рисунок А.2 – Разработанный москур для интеллектуальной системы

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВВГУ»)

Институт информационных технологий и анализа данных
Кафедра информационных технологий и систем

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ
на курсовую работу по дисциплине
«Курсовое проектирование»**

Студенту гр. БИН-21-1 Григорьевой Полине Иосифовне

- Задание 1.** Проанализировать и выбрать технические средства.
- Задание 2.** Создать ТЗ для разработки приложения.
- Задание 3.** Описать структуру данных проектируемой системы.
- Задание 4.** Разработать интерфейс приложения и прототип.

Структура курсовой работы:

- Постановка задачи;
- Выбор технологий;
- Разработка ТЗ;
- Описание структуры данных;
- Разработка интерфейса;
- Разработка прототипа.

Срок сдачи работы: 14.01.2025 г.

Руководитель,
Руководитель,
профессор кафедры ИТС,
доцент, д.т.н.



В.М. Гриняк

Задание получил:



П.И. Григорьева

КАЛЕНДАРНЫЙ ПЛАН-ГРАФИК

выполнения курсовой работы студента группы БИН-21-1

Студент: Григорьева Полина Иосифовна

Тема курсовой работы: «Проектирование интеллектуальной системы оценки доходности аренды жилой недвижимости»

Содержание выполняемых работ по программе	Сроки выполнения		Подпись руководителя
	Начало	Окончание	
Выбор технологий	11.11.2024	21.11.2024	<i>ММ</i>
Описание структуры данных	21.11.2024	14.12.2024	<i>ММ</i>
Разработка интерфейса	14.12.2024	03.01.2025	<i>ММ</i>
Разработка прототипа	03.01.2025	10.01.2025	<i>ММ</i>

Согласовано:

Студент

_____ (Дата) *П.И. Григорьева* (Подпись) (Григорьева Полина Иосифовна)

Руководитель курсовой работы
профессор кафедры ИТС,
доцент, д.т.н.

_____ (Дата) *В.М. Гриняк* (Подпись) (Гриняк Виктор Михайлович)