Министерство науки и высшего образования РФ форматирование?

ФГБОУ ВО «Нижневартовский государственный университет»

Факультет информационных технологий и математики

Кафедра информатики и методики преподавания информатики

~~Курсовая работа~~ Курсовой проект по дисциплине

“Основа научно-исследовательской деятельности”

**РАЗРАБОТКА ИНТЕРАКТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ТОВАРИЩЕСТВА СОБСТВЕННИКОВ ЖИЛЬЯ**

**Исполнитель:**

студент группы 3214

**Петров**

**Кирилл Викторович**

**Руководитель:**

кандидат педагогических наук,

доцент кафедры информатики и методики преподавания информатики  
**Манюкова Наталья Владиславовна**

Отметка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Нижневартовск, 2024

Оглавление

Введение

Глава 1. Теоретические аспекты разработки информационной системы товарищества собственников жилья

1.1 Анализ деятельности товарищества собственников жилья

1.2 Особенности автоматизации деятельности ТСЖ

1.3 Обзор существующих аналогов

1.4 Обоснование средств разработки информационной системы

Глава 2. Проектирование и Разработка информационной системы ТСЖ

2.1 Архитектура информационной системы

2.2 Проектирование базы данных

2.3 Проектирование интерфейса

2.4 Реализация разработки

Заключение.

Список использованной литературы

**Введение**

Отформатировать (См. Положение о КП)

**Актуальность исследования.** Значимость данного исследования, посвященного созданию интерактивной информационной системы для товариществ собственников жилья (ТСЖ), обусловлена необходимостью автоматизации деятельности этих организаций. С ростом количества жилых и дачных домов, а также увеличением числа товариществ собственников жилья, управление этими организациями становится все более сложным. Разработка интерактивной информационной системы для товариществ собственников жилья – довольно важная задача, которая не только направлена на повышение уровня жизни владельцев недвижимости, но и способствует устойчивому развитию местных сообществ.

**Объект исследования** – информационная система ТСЖ

**Предметы исследования** – разработка информационной системы ТСЖ

**Цель исследования** – разработать интерактивную информационную систему ТСЖ

Где знаки препинания?

**Задачи исследования:**

1. Анализ деятельности ТСЖ
2. Описание особенностей автоматизации деятельности ТСЖ
3. Обзор существующих аналогов
4. Обоснование средства разработки информационной системы
5. Проектирование информационной системы ТСЖ
6. Описание архитектуры информационной системы
7. Проектирование базы данных
8. Проектирование интерфейса
9. Реализация разработки

**~~Практическая значимость:~~**

**Практическая значимость** **исследования** заключается в возможности организации и поддержке деятельности ТСЖ, позволяющее осуществлять качественное и эффективное информационное сопровождение деятельности ТСЖ

Где знаки препинания?

Глава 1. **Теоретические аспекты разработки информационной системы ТСЖ**

* 1. **Анализ деятельности товарищества собственников жилья** по центру все заголовки

Товарищество собственников жилья (ТСЖ) - это некоммерческая организация, созданная жильцами для управления общим имуществом, создания комфортных условий проживания и обеспечения надлежащего уровня обслуживания. Основной целью ТСЖ является улучшение качества жилищных условий собственников, оптимизация эксплуатационных расходов.

Эффективность работы ТСЖ во многом зависит от уровня информированности его членов, прозрачности финансовых операций и качества предоставляемых услуг. В современном мире, где информационные технологии играют ключевую роль, автоматизация процессов управления становится необходимостью для повышения эффективности и благоустройства ТСЖ.

Убрать лишние отступы между абзацами

**Основные функции ТСЖ:**

* **Управление общим имуществом:** Поддержание в надлежащем состоянии общего имущества (дворы, подъезды, крыши, инжиниринг и т.д.).
* **Сбор взносов:** Обеспечение регулярного сбора взносов на текущие нужды (ремонт, уборка, коммунальные услуги).
* **Контроль расходов:** Мониторинг и контроль за финансовыми расходами, оптимизация бюджетных выделений.
* **Решение споров:** Урегулирование конфликтов между жильцами и представление их интересов перед государственными органами.

**Проблемы в деятельности ТСЖ:**

* Низкая активность жильцов: часто жильцы не заинтересованы в участии в жизни ТСЖ, что затрудняет принятие коллективных решений.
* Недостаток информации: Отсутствие доступа к актуальной информации о состоянии финансов и работе ТСЖ может снижать доверие жильцов.
* Проблемы с коммуникацией: Традиционные методы связи (например, бумажные уведомления) могут быть неэффективными.

Между параграфами должен быть один отступ

## 1.2 Особенности автоматизации деятельности ТСЖ

Автоматизация деятельности ТСЖ включает в себя внедрение программных решений, оптимизирующих различные аспекты управления. К ключевым особенностям автоматизации относятся:

* Упрощение учета и отчетности. Использование специализированного программного обеспечения позволяет автоматизировать процессы учета денежных средств, коммунальных платежей, а также формировать отчеты для собственников и контролирующих органов.
* Улучшение коммуникаций. Автоматизированные системы дают возможность создавать интерактивные площадки для общения между членами ТСЖ, что способствует более открытому и прозрачному управлению.
* Управление запросами и апелляциями. Внедрение систем учета обращений собственников, а также контроль за их исполнением, позволяет оперативно реагировать на проблемы и повышать качество обслуживания.
* Оптимизация процессов управления. Системы автоматизации помогают в планировании и организации работ по обслуживанию и ремонту, что способствует более эффективному использованию ресурсов.

Таким образом, автоматизация деятельности ТСЖ не только повышает эффективность управления, но и способствует улучшению качества жизни собственников.

**1.3 Обзор существующих аналогов**

На современном рынке программного обеспечения для ТСЖ представлено множество решений, каждое из которых обладает своими особенностями и функциональностью. К наиболее известным аналогам относятся:

**Программные комплексы для учета и отчетности**. Такие системы, как «1С: Управление ТСЖ», предоставляют возможности для ведения бухгалтерского учета, учета коммунальных платежей и формирования отчетов.

**Платформы для взаимодействия с собственниками.** Решения, такие как «ТСЖ Онлайн», предлагают интерактивные инструменты для общения между членами товарищества, а также возможность подачи заявок на услуги и обращения.

**Мобильные приложения.** Некоторые компании разрабатывают мобильные приложения, которые позволяют собственникам управлять своими платежами, следить за новостями и событиями в ТСЖ, а также получать актуальную информацию.

Однако, несмотря на наличие аналогов, существует необходимость в разработке уникальной системы, способной учесть специфику конкретного ТСЖ и удовлетворить потребности его членов.

Добавьте скрины и более детальный сравнительный анализ по определенным критериям

**1.4 Обоснование средств разработки информационной системы**

Основные критерии выбора средств разработки для информационной системы ТСЖ

При создании информационной системы для товарищества собственников жилья (ТСЖ) необходимо учитывать ряд факторов, влияющих на эффективность работы системы и её удобство для конечного пользователя. Вот основные критерии, которые будут учтены при разработке:

1. Гибкость и масштабируемость. Система должна быть тщательно спроектирована с учетом потенциальных изменений в потребностях пользователей. Важным аспектом является возможность интеграции новых функций и модулей, чтобы система могла адаптироваться к растущим требованиям ТСЖ.
2. Простота и доступность интерфейса имеет первостепенное значение для пользователей, незнакомых с технологиями, особенно для пожилых людей. Четкая навигация и понятная структура интерфейса.
3. Безопасность данных. Защита личной информации членов ТСЖ должна быть приоритетной задачей. Выбор технологий для разработки должен учитывать возможность реализации современных методов шифрования и защиты от несанкционированного доступа.
4. Производительность и стабильность. Система должна обеспечивать быстрое время отклика и стабильную работу под различными нагрузками. Выбранные технологии должны поддерживать эффективное управление ресурсами.
5. Кроссплатформенность. Возможность работы системы на различных устройствах и операционных системах расширяет её аудиторию пользователей и облегчает доступ к функционалу.
6. Интеграция с другими системами. Возможность интеграции с существующими системами и внешними сервисами (например, системами учета платежей, управления домами и т. д.) повысит функциональность информационной системы и упростит рабочие процессы.

С помощью этих критериев, можно выбрать правильные инструменты и технологии для успешного создания интерактивной информационной системы, которая будет эффективно функционировать и обеспечивать высокую степень удовлетворенности пользователей.

Для реализации данной задачи, мною выбраны следующие языки программирования: Python, Javascript, CSS, HTML

**Back-end (Серверная сторона)**

Python — это универсальный язык программирования, который выбран для разработки серверной части нашей интерактивной информационной системы. Он представляет собой язык высокого уровня, на котором возможно разрабатывать приложения в различных парадигмах программирования: императивное, объектно-ориентированное программирование (ООП), функциональное, асинхронное. Также его выбор обоснован следующими преимуществами:

* Простота и читабельность кода. Python известен своим легким синтаксисом, что позволяет быстро и эффективно разрабатывать, и поддерживать код.
* Широкая экосистема библиотек и фреймворков. Python предлагает множество библиотек, которые могут существенно ускорить разработку.
* Поддержка асинхронного программирования. Возможности Python по работе с асинхронными процессами (например, с использованием библиотеки asyncio), значительно улучшают производительность приложений.
* Широкое сообщество и поддержка. Python имеет огромное сообщество, что обеспечивает доступ к множеству ресурсов для обучения и решения проблем.

Фреймворк — это набор правил, шаблонов и инструментов, которые используются для построения продуктов или процессов как в программировании, так и в других областях. Для данного языка существуют различные фреймворки, которые задают структуру, определяют правила и предоставляют необходимый набор инструментов для создания web-приложений. Одними из наиболее актуальных и используемых фреймворков для Python являются: Django, FastAPI и Flask. В данном случае наиболее подходящим является FastAPI, по ряду причин:

* Производительность. Судя по названию, FastAPI делает упор на скорость. Фреймворк работает быстрее по сравнению с другими платформами Python. По ряду бенчмарков он показал лучшие результаты, чем Django или Flask. При этом FastAPI выигрывает не только в количестве обрабатываемых запросов в секунду, но и в скорости и простоте разработки.
* Гибкость. Нет какой-либо утвержденной четкой архитектуры, что дает возможность выбирать различные подходы разработки, по сравнению с Django (использует MVT – Model View Template).
* Асинхронность. FastAPI по умолчанию использует ASGI-серверы, в то время как в Django требуется время на конфигурацию приложения с WSGI на ASGI, а Flask не поддерживает асинхронный режим и функционирует только в среде WSGI.
* Встроенная проверка данных и документация. FastAPI предлагает удобную систему автоматического документирования. Он предоставляет пользовательский интерфейс на основе браузера, который в интерактивном режиме документирует API на основе графического пользовательского интерфейса Swagger UI. Также, фреймворк строит свою логику на основе Pydantic. Pydantic — это библиотека для Python, предназначенная для валидации и трансформации данных. Она помогает разработчикам гарантировать, что входные данные соответствуют установленным правилам и типам, а также обеспечивает их автоматическое преобразование в нужные форматы.

Также будет использован REST API – архитектурный подход, который устанавливает правила для API (Application programming interface - промежуточный интерфейс между двумя программами), который включает в себя 6 требований (1 опциональное) по проектированию:

* Модель клиент – сервер.
* Отсутствие состояния.
* Кэширование.
* Единый интерфейс.
* Многоуровневая система.
* Код по требованию. (опциональное)  
  Отформатировать

**Frontend (Клиентская часть)**

На клиентской части будет задействован JavaScript, TailWindCSS, HTML и CSS, но HTML код будет генерироваться на сервере (SSR – Server Side Rendering) с помощью Jinja2.

JavaScript — это основной язык для разработки клиентской части интерактивных веб-приложений. Leaflet, в свою очередь, является мощной библиотекой для работы с картами. Их выбор обоснован следующими преимуществами:

* Интерактивность. JavaScript позволяет создавать богатый и отзывчивый пользовательский интерфейс. В сочетании с библиотекой Leaflet, разработка интерактивных карт становится простой и эффективной.
* Поддержка большинства браузеров. JavaScript и Leaflet совместимы с широким спектром браузеров и устройств, что важно для доступности системы.

Jinja2 — это шаблонизатор для Python. Использование Jinja2 в проекте связано с следующими причинами:

* Гибкость шаблонов. Jinja2 позволяет отделить логику приложения от представления, что способствует лучшей структурированности кода.
* Интеграция с FastAPI. Jinja2 легко интегрируется с FastAPI, что делает его удобным инструментом для генерации HTML-страниц на основе данных, поступающих с сервера.
* Производительность. Быстрота и эффективность обработки шаблонов играют ключевую роль в создании динамического контента.

TailwindCSS — это утилитарный CSS-фреймворк, который позволяет быстро создавать адаптивные интерфейсы. Его выбор обусловлен следующими аспектами:

* Гибкость и кастомизация. TailwindCSS предлагает возможность настраивать стили на уровне утилит, что ускоряет процесс разработки и позволяет избежать избыточности кода.
* Адаптивность. Простота в создании адаптивного дизайна позволяет обеспечить хорошее отображение системы на различных устройствах.

**Среда разработки, окружение и контейнеризация:**

1. Visual Studio Code (VS Code)

Visual Studio Code представляет собой мощный редактор кода, обладающий множеством расширений и поддерживающий множество языков программирования. Выбор VS Code обусловлен следующими причинами:

* Поддержка разработки на Python и JavaScript. VS Code предоставляет отличные возможности для работы с этими языками, включая подсветку синтаксиса, автодополнение и интеграцию с терминалом.
* Расширяемость. Большое количество доступных расширений позволяет адаптировать среду под конкретные нужды проекта.

2. Poetry

Poetry — это инструмент для управления зависимостями Python и упаковки проектов. Его использование в проекте объясняется следующими аспектами:

* Упрощение работы с зависимостями. Poetry обеспечивает простую систему управления и разрешения зависимостей, что снижает вероятность конфликтов.
* Управление виртуальными окружениями. Автоматическое создание и использование виртуальных окружений упрощает работу с проектом в команде.
* Поддержка публикации. Poetry облегчает процесс упаковки приложения и его публикации на PyPI.

3. Docker является инструментом для контейнеризации приложений. Его использование в проекте оправдано следующими факторами:

* Портативность. Возможность создавать образы приложения и развертывать их на любой платформе без необходимости заботиться о зависимости среды.
* Изоляция окружения. Каждое приложение работает в своем контейнере, что минимизирует конфликты между зависимостями.
* Упрощенное развертывание. Docker позволяет автоматизировать развертывание приложений, что повышает надежность и удобство.

**База данных и инструменты**

PostgreSQL — это объектно-реляционная система управления базами данных, которая выбрана для использования в данной информационной системе. Предпочтение отдано PostgreSQL по следующим причинам:

* Надежность и производительность. PostgreSQL считается одной из самых надежных баз данных, обеспечивая высокую производительность даже на больших объемах данных. Это достаточно важно для интерактивных систем, которые требуют быстрой обработки запросов.
* Расширяемость. PostgreSQL позволяет создавать собственные типы данных, функции и операторы, что делает ее особенно гибкой для сложных приложений, требующих специфической логики обработки данных.

SQLAlchemy — это библиотека для работы с базами данных в Python, которая обеспечивает высокий уровень абстракции и гибкости при взаимодействии с реляционными базами данных. Выбор обоснован следующими критериями:

* ORM (Object Relational Mapping). SQLAlchemy позволяет разработчикам работать с базой данных в терминах объектов Python, что ускоряет процесс разработки и уменьшает количество ошибок связанных с прямыми SQL-запросами.
* Поддержка различных баз данных. SQLAlchemy поддерживает множество различных типов баз данных, включая PostgreSQL, что позволяет выбирать наиболее подходящее решение в зависимости от требований проекта.
* Гибкость в построении запросов. Возможности SQLAlchemy позволяют создавать сложные запросы без необходимости писать чистый SQL-код, что делает его удобным для разработчиков с разным уровнем подготовки.

Alembic является инструментом миграций для SQLAlchemy, который выбирается для управления изменениями в структуре баз данных. Он позволяет: Управление версиями схемы базы данных (отслеживание изменения и применение их в нужном порядке), генерация миграций, поддержка сложных миграций (изменение структуры, добавление индексов и т. д.)

**~~Итог~~**

Таким образом, совокупность этих инструментов и технологий позволит эффективно разработать интерактивную информационную систему для товарищества собственников жилья, удовлетворяющую требованиям пользователей и обеспечивающую высокую производительность

Глава 2

Заключение

Список использованной литературы