|  |  |
| --- | --- |
|  | Автономное учреждение  профессионального образования  Ханты-Мансийского автономного округа – Югры  «Нефтеюганский политехнический колледж» |

**Курсовая работа (проект)**

Создание приложения Гостиница для животных

09.02.07 Информационные системы и программирование

Выполнил(а) работу

обучающийся 3 курса \_\_\_\_\_\_\_\_\_\_\_\_ Степанов С. А.

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_ Степанов В. А.

(подпись)

г. Нефтеюганск, 2024

|  |  |
| --- | --- |
|  | Автономное учреждение  профессионального образования  Ханты-Мансийского автономного округа – Югры  «Нефтеюганский политехнический колледж» |

**Рецензия на курсовую работу (проект)**

**Обучающийся**  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(Фамилия, имя, отчество).

**Группа № \_\_\_\_\_\_\_\_\_\_\_**

**Специальность: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Квалификация: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***ТЕМА:* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**1. Краткое описание курсовой работы (проекта) и принятых решений:**

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

**2. Отрицательные стороны работы(проекта)**

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

**3. Положительные стороны работы (проекта)**

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

**4. Оценка конструкторской разработки и графического оформления**

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

**5. Оценка общеобразовательной подготовки и деловых качеств студента**

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

**Оценка работы: \_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**Руководитель курсовой работы (проекта)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Степанов В.А. )

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_20 г.

|  |  |
| --- | --- |
|  | Автономное учреждение  профессионального образования  Ханты-Мансийского автономного округа – Югры  «Нефтеюганский политехнический колледж» |

**З  А  Д  А  Н  И  Е**

**на курсовую работу (проект)**

Студент группы ИС 1.22 Степанов Сергей Алексеевич

**Содержание:**

- Введение.

Глава 1. Теоретическая часть.

1.1 Описание предметной области.

1.2 Обзор аналогов имеющихся решений проблемы.

1.3 Обоснование необходимости разработки.

1.4 Постановка задачи.

1.5 Состав ПЭВМ.

1.6 Используемое программное обеспечение.

Глава 2. Практическое часть.

2.1 Выбор средства автоматизации.

2.2 Дополнительные программные средства.

2.3 Требования к программе.

2.4 Проектирование программного продукта.

2.5 Текст программы с описанием.

2.6 Тестирование и отладка.

- Заключение.

- Список литературы.

Содержание графической части, практической работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Преподаватель профессионального цикла (руководитель работы) Степанов В. А.

г. Нефтеюганск, 2024 г.

**Содержание**

[ВВЕДЕНИЕ 5](#_Toc184899578)

[ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 8](#_Toc184899579)

[**1.1 Описание предметной области** 8](#_Toc184899580)

[**1.2 Обзор аналогов имеющихся решений проблемы** 11](#_Toc184899581)

[**1.3 Обоснование необходимости разработки.** 12](#_Toc184899582)

[**1.4 Постановка задачи** 13](#_Toc184899583)

[**1.5 Состав ПЭВМ** 16](#_Toc184899584)

[**1.6 Используемое программное обеспечение** 17](#_Toc184899585)

[ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ. 21](#_Toc184899586)

[**2.1 Выбор средства автоматизации** 21](#_Toc184899587)

[**2.2 Дополнительные программные средства** 23](#_Toc184899588)

[**2.3 Требования к программе** 24](#_Toc184899589)

ВВЕДЕНИЕ

Актуальность выбранной темы курсовой работы «Гостиница для животных» обоснована растущим интересом общества к качественному уходу за питомцами, особенно в условиях современного ритма жизни. С увеличением количества домашних животных и изменениями в образе жизни владельцев, востребованность услуг гостиниц для животных значительно возросла. Эти заведения обеспечивают питомцам комфортные условия в отсутствие их владельцев, предлагая профессиональный уход, внимание и безопасность. Кроме того, современное развитие информационных технологий открывает новые возможности для оптимизации процессов управления гостиницами для животных, что делает исследуемую тему еще более актуальной и перспективной.

В последние годы на рынке гостиничных услуг для животных наблюдается рост конкуренции, что требует более гибких подходов к управлению такими заведениями. В связи с этим, использование современных программных решений для автоматизации процессов бронирования, учета и коммуникации с клиентами становится необходимостью. Программное обеспечение, разработанное с использованием Python и библиотеки Flet, представляет собой удобное средство для создания интерфейсов и автоматизации различных задач гостиниц для животных. Это открывает новые перспективы для повышения эффективности работы таких заведений.

Состояние разработанности проблемы гостиничного сервиса для животных в научной литературе представлено рядом исследований в области гостиничного бизнеса, предпринимательства и использования информационных технологий. Однако, в контексте автоматизации процессов гостиниц для животных с использованием современных инструментов программирования, таких как Python и Flet, данный вопрос не получил должного освещения. Анализ литературы показал, что существует потребность в комплексном подходе к исследованию этой проблемы, который сочетает как организационные аспекты работы гостиниц для животных, так и современные технологические решения для их автоматизации.

**Объект исследования:** разработка приложений с графическим интерфейсом (GUI).

**Предмет исследования** — разработка кроссплатформенного приложения для управления гостиницей для животных с использованием библиотеки Flet на языке программирования Python.

**Целью работы** является разработка программного решения для гостиницы для животных, которое поможет автоматизировать основные процессы и улучшить качество обслуживания. Для достижения поставленной цели необходимо решить следующие задачи:

* Изучить существующие подходы и системы автоматизации в гостиничном бизнесе для животных.
* Разработать программное приложение для автоматизации работы гостиницы для животных с использованием Python и Flet.
* Проанализировать возможности использования данной системы для повышения эффективности работы гостиниц для животных.
* Провести тестирование и оценку функциональности разработанного программного решения.

Методами исследования являются анализ существующих решений в сфере гостиничного сервиса для животных, изучение возможностей программирования на Python с библиотекой Flet, а также проведение экспериментов и тестирования разработанного программного приложения.

Новизна работы заключается в применении современных технологий программирования для автоматизации деятельности гостиницы для животных, а также в разработке оригинальной программы с использованием Python и Flet для решения этой задачи.

Практическая значимость исследования заключается в возможности применения результатов для создания или улучшения программного обеспечения для гостиниц для животных, что повысит эффективность их работы и улучшит качество обслуживания клиентов.

Положения, выносимые на защиту:

* Применение технологий программирования на Python с библиотекой Flet может значительно улучшить процессы автоматизации в гостиницах для животных.
* Разработка специализированных программных решений для гостиниц для животных способна повысить эффективность и качество обслуживания.
* Системы автоматизации, созданные с использованием Python и Flet, обладают высокой гибкостью и адаптивностью к различным нуждам бизнеса.

# **ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **1.1 Описание предметной области**

Описание предметной области для курсовой работы по теме "Создание приложения гостиницы для животных " включает в себя следующие аспекты:

**Управление пользователями**:

* Система поддерживает регистрацию новых пользователей и вход в систему.
* Пользователи могут создать учетную запись, указав имя пользователя, пароль, контактные данные (имя, фамилию, email и телефон).
* Каждому пользователю назначается уникальный логин и email для идентификации в системе.

**Управление животными**:

* Животные, размещаемые в гостинице, могут быть добавлены владельцами через интерфейс системы.
* Для каждого животного фиксируются:

1. **Имя животного**.
2. **Тип животного** (например, кошка, собака и т.д.).
3. **Имя владельца** (владелец указывается через имя и фамилию).
4. **Количество дней** — срок пребывания животного в гостинице.
5. **Цена за пребывание** — фиксированная стоимость за день пребывания животного. В данном коде цена фиксирована на уровне 1200 рублей за день.
6. **Общая стоимость** — вычисляется как произведение количества дней и фиксированной стоимости за день.

**Цели и задачи системы**:

* **Регистрация пользователей** позволяет владельцам животных создавать учетные записи для доступа к функционалу системы.
* **Вход в систему** обеспечивает доступ пользователям к функционалу добавления животных в гостиницу и просмотра списка добавленных животных.
* **Добавление животных** дает владельцам возможность зарегистрировать своих питомцев в системе, указав необходимые данные.
* **Отображение списка животных** предоставляет пользователю информацию о текущих животных, находящихся в гостинице, и их стоимости за пребывание.

**Основные компоненты системы**:

* **Пользователи**: Включают владельцев животных, которые могут зарегистрироваться, войти в систему и добавить своих питомцев в базу данных.
* **Животные**: Включают данные о питомцах, такие как имя, тип, владелец и стоимость пребывания.
* **База данных**: Хранит информацию о пользователях и животных в двух таблицах SQLite (users и animals).

**Пример использования**:

* Владельцы животных могут зарегистрироваться в системе, войти в свой аккаунт и добавить своих питомцев для размещения в гостинице.
* После добавления животных, система рассчитывает общую стоимость их пребывания на основе количества дней и фиксированной цены.
* Пользователи могут увидеть список добавленных животных с их общей стоимостью пребывания.

**Основные сущности и их атрибуты:**

1.Пользователь (User):

id: Уникальный идентификатор пользователя.

username: Имя пользователя (логин).

password: Пароль пользователя.

first\_name: Имя пользователя.

last\_name: Фамилия пользователя.

email: Электронная почта пользователя.

phone: Телефон пользователя.

2. Животное (Animal):

id: Уникальный идентификатор животного.

name: Имя животного.

type: Тип животного (например, кошка, собака и т.д.).

owner: Имя владельца животного.

price: Стоимость за день пребывания (фиксированная на 1200 рублей).

days: Количество дней, на которые животное размещается в гостинице.

total\_price: Общая стоимость пребывания животного (вычисляется как price \* days).

**Процесс взаимодействия:**

Регистрация пользователя:

Пользователь регистрируется, предоставляя необходимые данные (логин, пароль, имя, фамилию, email и телефон).

При успешной регистрации создается новая запись в таблице users, а пользователь получает доступ к функционалу системы.

Вход пользователя:

Пользователь вводит свой логин и пароль, которые проверяются в базе данных.

Если данные верны, пользователь получает доступ к добавлению животных и просмотру списка своих питомцев.

Добавление животного:

Владелец вводит данные животного (имя, тип, владелец, количество дней пребывания).

Система рассчитывает стоимость пребывания, исходя из фиксированной цены за день.

Информация о животном сохраняется в таблице animals, и расчетная стоимость добавляется в систему.

Просмотр списка животных:

Пользователь может просматривать список животных, зарегистрированных в гостинице, с информацией о стоимости их пребывания.

**1.2 Обзор аналогов имеющихся решений проблемы**

**Цели обзора:**

1.Изучить основные функции и возможности аналогов.

2.Сравнить их с требованиями к искомому решению.

3.Сформулировать выводы о актуальности выбранной темы работы и поставленных в ней задач.

В последние годы наблюдается рост популярности мобильных приложений, которые предоставляют услуги для домашних животных. Это связано с тем, что все больше людей рассматривают своих питомцев как членов семьи, а соответственно, их благополучие и комфорт становятся важными аспектами повседневной жизни. Одним из решений для обеспечения заботы о питомцах является использование мобильных приложений для гостиниц для животных, которые позволяют владельцам питомцев находить удобные и надежные места для размещения своих животных во время путешествий, командировок или других обстоятельств. В этом обзоре будут рассмотрены существующие решения в данной области, их сильные и слабые стороны, а также технологические аспекты, которые могут быть полезны при разработке нового приложения.

На рынке существует несколько популярных мобильных приложений и платформ для гостиниц для животных, которые обеспечивают широкий спектр услуг, включая размещение животных, выгуливание, груминг и ветеринарное обслуживание.

**1.PetBacker.** PetBacker — это онлайн-платформа и мобильное приложение для поиска услуг для домашних животных, включая гостиницы для животных, сиделок, груминг и выгуливание собак. Она работает в более чем 30 странах мира и предоставляет пользователям возможность выбирать подходящее место для питомца, просматривать отзывы, а также устанавливать контакт с поставщиками услуг. Сильной стороной PetBacker является широкий выбор опций и возможность выбора между частными и профессиональными поставщиками услуг. Однако приложение имеет несколько недостатков, среди которых сложность навигации и отсутствие функции отслеживания в реальном времени.

**2.PetHotel.** PetHotel — это услуга от известного сервиса бронирования гостиниц, предлагающая размещение для питомцев. Это часть платформы Booking.com, которая позволяет пользователям забронировать отель для питомцев на заданный период.

**Сильные стороны**:

Поддержка большого количества отелей и гостиниц по всему миру.

Интеграция с известной платформой бронирования отелей.

**Слабые стороны**:

Услуга ограничена лишь несколькими странами.

Нет функционала для управления бронированиями внутри приложения для владельцев.

**1.3 Обоснование необходимости разработки.**

Обоснование необходимости разработки приложения гостиница для животных:

1. Удобство для пользователей

Приложение позволит владельцам домашних животных быстро находить подходящие гостиницы для своих питомцев, бронировать места онлайн, отслеживать состояние своего питомца через уведомления и фото/видео отчеты от персонала гостиницы. Это существенно экономит время и силы, особенно для занятых людей, которым нужно оставить животное на короткий срок.

2. Повышение уровня сервиса

Современные технологии позволяют улучшить взаимодействие между владельцами животных и персоналом гостиницы. Приложение может включать такие функции, как отслеживание состояния здоровья животного, отправка уведомлений о прогулках, кормлении и других процедурах. Владельцы смогут получать актуальную информацию о своем питомце в режиме реального времени, что повысит их доверие к услугам гостиницы.

3. Увеличение конкурентоспособности

В условиях растущего спроса на услуги по временному содержанию животных, наличие удобного мобильного приложения может стать важным конкурентным преимуществом для гостиниц. Пользователи будут выбирать те заведения, которые предлагают современные и удобные сервисы.

4. Расширение рынка

Развитие цифровых технологий открывает новые возможности для бизнеса. Создание приложения позволит привлечь новых клиентов, включая тех, кто ранее не пользовался услугами гостиниц для животных из-за отсутствия информации или неудобства процесса бронирования.

5. Социальная значимость

Питомцы становятся все более важной частью жизни многих людей, и забота о них выходит на первый план. Разработка приложения, которое помогает обеспечить комфорт и безопасность животным во время отсутствия хозяев, способствует улучшению качества жизни как владельцев, так и их питомцев.

Таким образом, разработка приложения «Гостиница для животных» представляет собой перспективный проект, который может принести пользу как пользователям, так и бизнесу, повышая уровень обслуживания и удовлетворенность клиентов.

## **1.4 Постановка задачи**

**Математическая или логическая модель объекта**

В данном проекте разрабатывается программное обеспечение для автоматизации управления данными пользователей и животных в гостинице для животных. Модель включает две основные сущности: пользователей и животных. Каждый пользователь имеет уникальный идентификатор, а также связанный с ним набор данных, таких как имя, пароль, email и телефон. Каждое животное также имеет уникальный идентификатор и включает информацию о его владельце, типе животного, количестве дней пребывания и стоимости.

Основная логика расчета заключается в определении общей стоимости пребывания животного. Стоимость рассчитывается по формуле:

Total\_price=Price\_per\_day×Days

где Price\_per\_day — это фиксированная цена за день пребывания, а Days — количество дней, на которое животное размещается в гостинице.

**Функциональное назначение**

Разрабатываемое программное обеспечение должно выполнять следующие функции:

1. **Регистрация пользователей** — создание новых записей с уникальными данными о пользователях (имя, пароль, email, телефон и т.д.).
2. **Авторизация пользователей** — вход в систему по имени пользователя и паролю для дальнейшего взаимодействия с системой.
3. **Управление данными животных** — добавление, изменение и удаление записей о животных (имя животного, тип, владелец, количество дней пребывания).
4. **Расчет стоимости пребывания** — автоматическое вычисление общей стоимости на основе фиксированной цены за день и числа дней.
5. **Отчеты и вывод информации** — создание отчетов о зарегистрированных пользователях и животных, а также информирование пользователя о статусе операций.
6. **Обратная связь с пользователем** — вывод сообщений об ошибках (например, если введены неверные данные) или успешных действиях (например, успешная регистрация пользователя или добавление животного).

#### **Эксплуатационное назначение**

Программное обеспечение будет использоваться круглосуточно для регистрации новых пользователей и добавления данных о животных. Для пользователей, которые хотят зарегистрировать своих питомцев, система будет использоваться периодически (например, при добавлении новых питомцев). Для администраторов и сотрудников системы программное обеспечение будет использоваться ежедневно для мониторинга и управления данными. Результаты работы программы будут поступать к пользователям и администраторам для выполнения их задач, таких как регистрация животных, расчет стоимости, создание отчетов и предоставление информации о статусе животных.

#### **Информационная модель задачи**

* **Входные данные:**
  1. Данные о пользователях: имя пользователя, пароль, email, телефон.
  2. Данные о животных: имя животного, тип животного, владелец, количество дней пребывания.
* **Выходные данные:**
  1. Сообщения о статусе операций (успешно зарегистрирован, ошибка ввода данных).
  2. Отчеты о животных: список зарегистрированных питомцев с их владельцами и стоимостью пребывания.
  3. Ошибки авторизации или регистрации.
* **Идентификаторы:**
  1. Уникальный идентификатор пользователя — используется для идентификации и доступа к данным пользователя.
  2. Уникальный идентификатор животного — используется для отслеживания каждого питомца в системе.
  3. Идентификаторы транзакций или операций для отслеживания состояния и истории данных.

**1.5 Состав ПЭВМ**

Для создания доски объявлений на языке программирования Python с использованием графического интерфейса PyQt6 использовалась вычислительная техника, включающая в себя следующие компоненты:

**Персональный компьютер (ПК)**:

* Центральный процессор (CPU): Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz 3.00 GHz
* Оперативная память (RAM): минимум 8 ГБ RAM. Этого объема достаточно для эффективной работы если в процессе разработки будет использоваться несколько приложений и браузеров одновременно.
* Жесткий **(HDD/SSD)**: SSD объемом не менее 256 ГБ для ускоренной загрузки операционной системы, среды разработки и инструментов. SSD значительно ускоряет работу при компиляции и запуске программ.
* Графическая карта (GPU): встроенная графика процессора или отдельная видеокарта, не требуется мощная графическая карта, так как задачи, связанные с графическим интерфейсом PyQt6, не нагружают GPU.

ПК должен быть достаточно мощным для комфортной разработки программного обеспечения и запуска приложений на основе PyQt6.

**Монитор**: разрешение экрана: Full HD (1920x1080) или выше, c диагональю 24 дюйма или больше для удобства работы с кодом и интерфейсами.

**Клавиатура и мышь**: стандартные устройства ввода, используются для управления приложением.

**Сетевое оборудование**: для скачивания необходимых библиотек и инструментов, а также для обновления операционной системы и среды разработки.

Таким образом, перечисленное оборудование и программное обеспечение обеспечивает комфортные условия для разработки и запуска доски объявлений на языке программирования Python с использованием PyQt6.

## **1.6 Используемое программное обеспечение**

Понятие программного обеспечения

Программное обеспечение (ПО) — это совокупность программ, которые позволяют выполнять вычислительные задачи на вычислительных устройствах. ПО можно разделить на два основных типа:

1. **Системное программное обеспечение** — это набор программ, который управляет аппаратными средствами и создает основу для работы прикладных программ. Оно обеспечивает взаимодействие между пользователем и компьютером, а также выполняет низкоуровневые задачи, такие как управление памятью, файловой системой и периферийными устройствами.
2. **Прикладное программное обеспечение** — это программы, предназначенные для решения конкретных задач пользователей, например, базы данных, графические редакторы, системы управления проектами и другие.

ПО играет критически важную роль в любом вычислительном процессе, так как без программных средств невозможно использовать оборудование для решения задач.

Структура и состав ПО

1. **Системное программное обеспечение**:
   * **Операционная система (ОС)** — это основа всего ПО, которая управляет вычислительными ресурсами и обеспечивают выполнение прикладных программ. В нашем проекте используется **Ubuntu Server** для серверной части. Эта ОС оптимизирована для работы с большими объемами данных и серверными приложениями, предоставляет множество инструментов для администрирования, управления безопасностью и сетевыми функциями. Для рабочих станций используется **Windows 10/11**, что позволяет легко работать с пользовательским интерфейсом и запускать разнообразные прикладные программы.
   * **Средства безопасности** — включают в себя **фаерволы** (например, **ufw** на Ubuntu), **антивирусные программы**, **системы резервного копирования** и средства шифрования данных. Эти компоненты играют важную роль в защите информации от несанкционированного доступа, предотвращении атак и обеспечении безопасности данных пользователей.
2. **Прикладное программное обеспечение**:
   * **Язык программирования Python** — используется для разработки основной логики приложения. Python был выбран благодаря своей простоте, широкому набору библиотек и возможностям для быстрого прототипирования. Он является основным инструментом для реализации функционала, работы с базой данных, обработки изображений и генерации отчетов.
   * **Библиотека Flet** — служит для создания интерактивных графических интерфейсов пользователя. Это современный инструмент, который позволяет создавать веб-приложения с минимальными усилиями, поддерживая взаимодействие с пользователем через элементы управления, такие как текстовые поля, кнопки, списки и другие.
   * **SQLite** — используется как база данных для хранения информации о пользователях, их данных, а также информации о животных, их стоимости и заказах. SQLite является встроенной базой данных, что позволяет легко использовать ее без необходимости настройки отдельного сервера базы данных, а также позволяет использовать ее как на сервере, так и на клиентских машинах.
   * **Pillow (PIL)** — это библиотека для работы с изображениями в Python. Она используется для обработки изображений животных (например, для их загрузки, отображения и изменения размера). Благодаря этой библиотеке, пользователи могут загружать изображения и видеть их на экране, а также производить операции с изображениями, такие как изменение формата или качество.

Роль операционной системы

Операционная система является важнейшей составляющей системного программного обеспечения и выполняет несколько критически важных функций:

1. **Управление аппаратными средствами**: ОС взаимодействует с процессором, оперативной памятью, накопителями данных и другими устройствами, позволяя прикладным программам использовать эти ресурсы. Она управляет памятью, процессами, устройствами ввода-вывода, файловыми системами и другими аппаратными компонентами.
2. **Многозадачность**: ОС позволяет одновременно выполнять несколько программных процессов. Например, в рамках проекта можно одновременно обрабатывать запросы от нескольких пользователей, выполнять операции с базой данных, а также обновлять интерфейс.
3. **Управление безопасностью**: ОС обеспечивает защиту данных и предотвращение несанкционированного доступа. Она использует различные механизмы безопасности, такие как шифрование, контроль прав доступа, а также поддерживает работу фаерволов и антивирусных программ. Например, ОС Ubuntu Server предоставляет ряд встроенных средств безопасности, таких как SELinux (Security-Enhanced Linux), которые помогают защитить сервер от внешних угроз.
4. **Обработка ошибок и стабильность**: ОС отслеживает выполнение процессов, проверяет их на наличие ошибок и восстанавливает их состояние в случае сбоев. Она также управляет системой журналирования (логирования) событий, что позволяет отслеживать проблемы и выполнять диагностику.
5. **Управление сетью**: ОС предоставляет сетевые службы, которые позволяют взаимодействовать с другими компьютерами и пользователями через интернет или локальные сети. Например, с помощью протоколов TCP/IP, HTTP и других ОС обеспечивает обмен данными между сервером и клиентами, а также помогает в настройке сетевой безопасности.

Без операционной системы невозможно эффективно работать с аппаратными ресурсами и прикладными программами. ОС является связующим звеном между пользователем и вычислительной техникой, и без нее работа любого приложения будет невозможна.

**ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ.**

**2.1 Выбор средства автоматизации**

Средство автоматизации — это инструментальная система, которая объединяет программные и технические средства для разработки, тестирования, развертывания и эксплуатации программных продуктов. Эти средства помогают ускорить процесс создания приложений, обеспечивают качество кода и его поддержку, а также позволяют адаптировать готовые решения под потребности конечного пользователя.

#### 1. Обзор средств автоматизации

Для разработки приложений с интерфейсом и взаимодействием с базой данных существует несколько типов инструментальных систем, каждый из которых решает определенные задачи:

* **Средства разработки**:
  + **IDE (Интегрированные среды разработки)**, такие как **PyCharm**, **Visual Studio Code**, и **Eclipse**, предоставляют мощные возможности для написания, отладки и тестирования кода. Эти среды поддерживают различные языки программирования и предоставляют встроенные инструменты для работы с базами данных, системами контроля версий и другими инструментами разработки.
* **Системы управления версиями (VCS)**:
  + **Git** является наиболее популярной системой управления версиями, используемой для отслеживания изменений в коде. Для работы с Git используются такие платформы, как **GitHub**, **GitLab**, или **Bitbucket**, которые обеспечивают удобную интеграцию с CI/CD пайплайнами и позволяют вести совместную разработку.
* **Средства работы с базами данных**:
  + Для работы с базами данных можно использовать такие системы, как **SQLite**, **MySQL**, **PostgreSQL**, и **MongoDB**. В данном проекте используется **SQLite**, так как она является легковесной и не требует установки отдельного серверного программного обеспечения, что идеально подходит для небольших приложений, работающих с ограниченными ресурсами.
  + 2. Средства, использованные в проекте

В данном проекте для создания программы с графическим интерфейсом, работы с базой данных и автоматизации процессов разработки были выбраны следующие инструменты:

1. **Python и Flet** — для разработки приложения.
   * **Python** выбран за его универсальность и большое количество библиотек для работы с базами данных, тестирования и создания пользовательских интерфейсов. Это один из самых популярных языков программирования, что делает его удобным для разработки и поддержки.
   * **Flet** используется для создания графического интерфейса приложения. Это современный фреймворк для разработки кросс-платформенных приложений с минимальными усилиями, который поддерживает как локальные, так и веб-приложения.
2. **SQLite** — для работы с базой данных.
   * **SQLite** выбрана для хранения данных о пользователях и животных, так как это легковесная база данных, которая не требует настройки серверного компонента. Это удобное решение для небольших проектов и приложений, которые не требуют сложной структуры и высокой производительности.
3. **Git** — для контроля версий.
   * Для управления версиями кода используется **Git**. Это обеспечит возможность отслеживать изменения в проекте и координировать совместную работу. Платформы, такие как **GitHub** или **GitLab**, могут быть использованы для хранения репозитория и автоматической проверки кода через CI/CD пайплайны.3. Обоснование выбора средств автоматизации

Для этого проекта был выбран следующий стек инструментов, который обеспечит эффективную разработку, тестирование и развертывание приложения:

* **Python и Flet** предоставляют необходимую гибкость для создания как серверной, так и клиентской части приложения с минимальными усилиями. Flet позволяет быстро и удобно разрабатывать графический интерфейс, а Python — легко управлять логикой программы и работать с базами данных.
* **SQLite** идеально подходит для этого проекта, так как приложение требует работы с небольшой базой данных без необходимости развертывания сложной инфраструктуры. Это решение ускоряет процесс разработки и не требует дополнительных затрат на серверное оборудование.
* **Git** выбран для управления версиями, что позволяет обеспечить совместную работу над проектом и гарантировать, что все изменения в коде отслеживаются. Это критически важно при работе в команде и позволяет легко откатить изменения, если возникнут проблемы.

Таким образом, выбранный набор инструментов предоставляет все необходимые функции для разработки и поддержания приложения, позволяя эффективно работать с базой данных, управлять версиями кода, тестировать приложение и автоматизировать процессы разработки и развертывания.

## **2.2 Дополнительные программные средства**

В процессе разработки программного продукта были использованы следующие дополнительные программные средства:

1. **Генератор отчета**:
   * Для формирования отчетов о животных, владельцах и стоимости услуг использовались инструменты для работы с базами данных и генерации отчетов. Эти отчеты создаются с использованием SQL-запросов к базе данных, что позволяет динамически формировать данные и выводить их в нужном формате.
2. **Генератор меню**:
   * Меню интерфейса было создано с использованием компонентов фреймворка Flet, который предоставляет инструменты для создания динамического интерфейса с элементами управления, такими как кнопки для регистрации, входа и навигации между различными экранами программы.
3. **Генератор экрана**:
   * Для генерации экранов программы использовались компоненты, такие как Container, Column, TextField, ElevatedButton и другие, предоставляемые Flet. Эти компоненты позволяют динамически создавать экраны для регистрации, входа в систему и добавления животных, а также управлять их отображением в зависимости от состояния программы.
4. **Формирование исполняемого файла (или дистрибутива)**:
   * Для формирования исполняемых файлов использовались инструменты сборки, такие как **PyInstaller** или **cx\_Freeze**. Эти средства позволяют преобразовать исходный код программы на Python в независимые исполняемые файлы (например, .exe для Windows), которые не требуют установки Python или дополнительных зависимостей на машине пользователя.
5. **Средства SQL**:
   * В качестве системы управления базами данных использована **SQLite**, которая является легковесной и встроенной СУБД. Для взаимодействия с базой данных использовались стандартные SQL-запросы для создания таблиц, выборки данных, вставки и обновления записей

## **2.3 Требования к программе**

#### **Требования к функциональным характеристикам**

Программа представляет собой приложение для работы с базой данных пользователей и животных, с возможностью регистрации, входа, добавления и просмотра информации о животных. Программа должна поддерживать следующие функциональные характеристики:

1. **Функции**:
   * Регистрация нового пользователя с проверкой уникальности имени пользователя и электронной почты.
   * Вход в систему с проверкой имени пользователя и пароля.
   * Добавление животного в базу данных с расчетом общей стоимости за пребывание.
   * Просмотр списка добавленных животных с отображением их характеристик и стоимости.
   * Загрузка изображений для визуального оформления интерфейса.
   * Управление данными пользователей и животных через базу данных (SQLite).
2. **Состав и характеристики данных**:
   * Входные данные:
     + Имя пользователя, пароль, электронная почта, телефон и другие данные для регистрации.
     + Информация о животных: имя, тип, имя владельца, количество дней пребывания, стоимость за день.
   * Выходные данные:
     + Уведомления о статусе регистрации, входа или добавления животного.
     + Список всех зарегистрированных животных с их данными и общей стоимостью.
3. **Критерии эффективности**:
   * Максимально допустимое время ответа системы на запросы (например, время отклика при регистрации или добавлении животного) не должно превышать 5 секунд.
   * Программа должна работать с объемом базы данных до 1000 записей животных и пользователей без значительных задержек.
   * Программа должна использовать не более 200 МБ оперативной памяти.

#### Требования к надежности

Для обеспечения надежности работы программы предусмотрены следующие меры:

1. Контроль входных данных (например, проверка на пустые поля при регистрации и добавлении животного).
2. Проверка уникальности данных (например, уникальность имени пользователя и электронной почты).
3. Резервное копирование промежуточных данных (например, перед обновлением или удалением данных).
4. Обработка исключений при работе с базой данных и внешними ресурсами (например, при загрузке изображения или при взаимодействии с базой данных).
5. Регистрация ошибок и логирование для отслеживания и устранения неполадок.

#### **Требования к составу и параметрам технических средств**

Программа должна работать на следующих технических средствах:

1. **Минимальная конфигурация**:
   * Процессор: 1 ГГц.
   * Оперативная память: 2 ГБ.
   * Свободное место на диске: 50 МБ для базы данных и программы.
   * Операционная система: Windows, Linux, macOS (с установленным Python и SQLite).
2. **Рекомендуемая конфигурация**:
   * Процессор: 2 ГГц и выше.
   * Оперативная память: 4 ГБ и выше.
   * Свободное место на диске: 100 МБ и более для работы приложения.
   * Операционная система: Windows 10 и выше, Linux (Ubuntu или аналогичные дистрибутивы), macOS (с установленным Python 3.x и SQLite).

#### **Требования к информационной и программной совместимости**

Программа должна быть совместима с рядом операционных систем и программных средств:

1. **Операционная система**: Windows, Linux, macOS.
2. **Язык программирования**: Python 3.x.
3. **Используемая среда**: Flet для пользовательского интерфейса.
4. **База данных**: SQLite (библиотека sqlite3).
5. **Дополнительные компоненты**:
   * Для работы с изображениями требуется наличие библиотеки для работы с графикой (например, библиотеки для работы с изображениями в Python, такие как Pillow).

Программа должна корректно взаимодействовать с базой данных SQLite и поддерживать работу на всех перечисленных операционных системах без дополнительных настроек или модификаций.