Изображение выглядит как графическая вставка, Графика, мультфильм

Автоматически созданное описание

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Тема: Разработка программного модуля**

**информационной системы «Продажа авиабилетов»**

**Специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **Выполнил студент группы 311ИС-22** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **П. В. Лиляков** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Н. И. Кручинкина** |

**Москва 2024**

Изображение выглядит как графическая вставка, Графика, мультфильм

Автоматически созданное описание

**УТВЕРЖДАЮ**

**Зам. директора КМПО**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С. Ф. Гасанов**

**«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.**

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Специальность 09.02.07 «Информационные системы и программирование»**

**Студент группы 311ИС-22 Лиляков Павел**

**ТЕМА: «Разработка программного модуля информационной**

**системы «Продажа авиабилетов»**

**Дата выдачи задания «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г.**

**Срок сдачи проекта «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.**

**Москва 2024**

**Перечень вопросов, подлежащих разработке:**

1. Информационное обеспечение задачи
2. Обзор и анализ существующих решений
3. Постановка задачи
4. Разработка требований к проекту
5. Функциональные характеристики
6. Характеристики системы
7. Построение диаграммы использования
8. Разработка сценария проекта
9. Проектирование базы данных
10. Реализация интерфейса приложения на Tkinter и Figma
11. Реализация проекта системы
    1. Описание среды разработки
    2. Разработка программного модуля
    3. Результат работы и тестирования

**Исходные данные:**

**Перечень разрабатываемых материалов для визуализации:** документация, содержащая в том числе:

1. ER-диаграмма базы данных.
2. Архитектура программного модуля
3. Экранные формы интерфейса.

Задание выдал:

Руководитель курсового проекта Н.И. Кручинкина

Задание принял к исполнению В.Е. Третьякова

*Рассмотрено*

на заседании предметно-цикловой комиссии

информационных технологий и системного

администрирования

Протокол № \_\_\_*от «\_\_» \_\_\_\_\_\_\_\_\_* 2024 г.

Председатель ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_

СОДЕРЖАНИЕ

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc185539500)

[1.1 Информационное обеспечение задачи 7](#_Toc185539501)

[1.2 Обзор и анализ существующих программных решений 7](#_Toc185539502)

[1.3 Постановка задачи. Структура входной и выходной информации 8](#_Toc185539503)

[2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ 10](#_Toc185539504)

[2.1 Разработка требований к проекту 10](#_Toc185539505)

[2.1.1 Функциональные характеристики 10](#_Toc185539506)

[2.1.2. Характеристики системы 11](#_Toc185539507)

[2.2 Построение диаграммы использования 12](#_Toc185539508)

[2.3 Разработка сценария проекта 14](#_Toc185539509)

[2.4 Проектирование базы данных 17](#_Toc185539510)

[2.5 Реализация интерфейса приложения на Tkinter и Figma 19](#_Toc185539511)

[3 РЕАЛИЗАЦИЯ ПРОЕКТА СИСТЕМЫ 24](#_Toc185539512)

[3.1 Описание среды разработки 24](#_Toc185539513)

[Основные функции PyCharm: 24](#_Toc185539514)

[3.2 Разработка программного модуля 24](#_Toc185539515)

[3.2.1 Реализация пользовательского интерфейса программы 24](#_Toc185539516)

[3.3 Результат работы и тестирования 26](#_Toc185539517)

[ЗАКЛЮЧЕНИЕ 28](#_Toc185539518)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](#_Toc185539519)

[Стандарты и законодательные материалы 29](#_Toc185539520)

[Учебники и учебные пособия 29](#_Toc185539521)

[Интернет-источники 30](#_Toc185539522)

[Монография 31](#_Toc185539523)

[Приложение 1 32](#_Toc185539524)

[Приложение 2 35](#_Toc185539525)

# 

**ВВЕДЕНИЕ**

Актуальность данной темы курсового проекта обусловлена необходимостью гибкой и легко настраиваемой системы, способной удовлетворить потребности широкого спектра пользователей. Современный мир наблюдает значительный рост авиационной инфраструктуры и туристической отрасли. Несмотря на начальные затраты, бизнес по продаже авиабилетов может стать прибыльным. Это создает высокий спрос на эффективные системы бронирования и продажи авиабилетов.

Целью проекта является разработка программного модуля системы продажи авиабилетов на языке Python с использованием библиотеки Tkinter. Такой модуль позволит эффективно управлять процессами заказа и продажи авиабилетов, автоматизируя работу авиакомпании и предоставляя удобный интерфейс для клиентов.

Для осуществления поставленной цели необходимо решить следующие задачи:

1. Проанализировать информацию из источников в интернете, которые помогут реализовать проект.
2. Проанализировать существующие решения для автоматизации покупки авиабилетов и выявить их преимущества и недостатки.
3. Разработать базу данных, создание схемы базы данных для хранения информации о рейсах, пассажирах, бронированиях и реализация подключения к базе данных.
4. Разработать пользовательский интерфейс, создание основной страницы приложения
5. Провести тестирование приложения и внести необходимые коррективы на основе полученных результатов.

Объектом данного исследования является продажа авиабилетов

Предметом исследования является система бронирования и

продажи билетов.

Практическая значимость исследования продажи билетов имеет важное значение для различных отраслей и организаций. Результаты исследования могут быть применены в реальных условиях и помочь оптимизировать процесс продажи билетов.

# 

# **1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

## **Информационное обеспечение задачи**

Вступительное слово

В данном разделе необходимо предоставить краткое, но емкое введение в тему продажи авиабилетов. Это может включать общую информацию о рынке авиасообщения, важности точного планирования и необходимости эффективной системы управления продажами.

Структура базы данных

База данных является ключевой частью информационного обеспечения системы. Она должна содержать следующие основные таблицы:

**Авиабилеты**: информация о маршруте, аэропортах, категории, количестве, цене, наличии.

**Клиенты**: данные о клиентах, включая контактную информацию и историю покупок.

**Сотрудники:** информация о работниках, включая их должности, контактные данные, историю продаж, график продаж.

**Продажи**: данные о совершенных приобретениях, включая дату, время, количество и сумму.

## **Обзор и анализ существующих программных решений**

На современном рынке существует множество онлайн-площадок для продажи авиабилетов. Анализ существующих решений показывает следующее:

1. **Skyscanner** - популярная платформа для поиска и бронирования авиарейсов. Предоставляет широкие возможности фильтрации и сравнения цен.  Компания стала одним из ведущих игроков в сфере онлайн-бронирования путешествий.
2. **Aviasales** - один из ведущих мета поисковых сервисов для поиска авиабилетов в России. Предлагает удобные инструменты поиска: календари низких цен, карты цен, подписки на изменения цены. Работает с более чем 700 авиакомпаниями и 200 агентствами.
3. **TUTU.RU** - крупнейший оператор России, предлагающий комплекс услуг по бронированию билетов и туров, имеющий мобильные приложения для iOS и Android.

Анализ этих решений показывает, что большинство из них предлагают схожие функциональные возможности, но различаются по уровню сервиса, ассортименту и ценовой политике. Учитывая уникальные особенности работы агентства по продаже авиабилетов, обеспечивая более эффективное управление запасами, продажами и клиентским обслуживанием, открывает возможности для создания более специализированного и удобного программного продукта.

## **1.3 Постановка задачи. Структура входной и выходной информации**

На основе анализа работы агентства по продаже авиабилетов, были сформулированы следующие основные задачи разработки системы:

* Учет товаров (авиабилетов): создание и ведение базы данных обо всех доступных рейсах.
* Управление запасами: автоматическое отслеживание количества свободных мест на рейсы.
* Анализ продаж: формирование отчетов о продажах за различные периоды.
* Ведение клиентской базы: учет информации о клиентах и их бронирований.
* Учет сотрудников: ведение данных о работниках агентства и их результатах продаж.

Структура входной информации включает:

* Данные о рейсах (номер рейса, маршрут, дата вылета, количество мест, цена билета).
* Данные о клиентах (имя, контактная информация).
* Данные о сотрудниках (имя, должность, результаты продаж).
* Данные о бронированиях и продажах (дата, сумма, номер клиента, таб. номер продавца).

Выходная информация будет представлять собой отчеты, такие как:

* Отчеты о продажах за определенный период.
* Остатки мест на рейсах.
* Аналитические данные для принятия управленческих решений.
* Отчеты о работе сотрудник

# **2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ**

## **Разработка требований к проекту**

Разработка требований к проекту является ключевым этапом в создании системы учета и реализации товарной базы в авиационном агентстве. На этом этапе необходимо четко определить функциональные и нефункциональные характеристики, которые будут служить основой для проектирования и воплощения системы.

1. **Функциональные характеристики**

Функциональные характеристики описывают, что именно должна делать система. В контексте системы учета и реализации товарной базы в авиационном агентстве можно выделить следующие ключевые функции:

1. **Учет рейсов:** Система должна позволять добавлять новые рейсы в базу данных с указанием наименования, маршрута, класса, стоимости и количества мест. Необходимо обеспечить возможность редактирования информации о рейсах, включая изменение цены и количества доступных мест. Система должна поддерживать удаление рейсов из базы данных, если они больше не доступны для реализации. Должна быть возможность поиска рейсов по различным критериям (наименование, маршрут, класс).
2. **Учет клиентов:** Система должна хранить информацию о клиентах, включая имя, контактные данные (телефон, электронная почта) и историю заказов. Необходимо обеспечить возможность добавления новых клиентов и редактирования существующих данных. Система должна поддерживать возможность удаления клиентов из базы данных.
3. **Учет авиакомпаний:** Система должна содержать информацию о партнерских авиакомпаниях, включая имя, контактные данные и условия сотрудничества. Необходимо обеспечить возможность добавления новых авиакомпаний и редактирования их данных. Система должна поддерживать возможность удаления авиакомпаний из базы данных.
4. **Учет пунктов назначения: Система** должна содержать информацию о пунктах назначения, включая название города, страны и важные географические или туристические характеристики. Необходимо обеспечить возможность добавления новых пунктов назначения и редактирования существующих данных. Система должна поддерживать возможность поиска пунктов назначения по различным критериям (название города, страна).
5. **Учет продаж:** Система должна фиксировать информацию о каждой продаже, включая дату и время, наименование товара (рейса), количество, сумму и способ оплаты. Необходимо обеспечить возможность просмотра истории продаж и фильтрации данных по различным критериям (дата, товар, клиент). Система должна автоматически обновлять количество доступных мест на рейсе после каждой продажи.
6. **Характеристики системы**

Характеристики системы описывают особенности функционирования системы, которые не связаны с конкретными функциями, но важны для ее успешной работы. К ним относятся:

1. **Производительность:** Система должна обеспечивать быстрый обход запросов пользователей, включая добавление, редактирование и удаление данных. Время отклика системы на запросы не должно превышать 2 секунды.
2. **Безопасность:** Доступ к системе должен быть защищен с помощью аутентификации пользователей (логин и пароль). Необходимо реализовать уровни доступа для различных ролей пользователей (администратор, продавец), чтобы ограничить доступ к определенным функциям.
3. **Удобство использования:** Интерфейс системы должен быть интуитивно понятным и удобным для пользователей, чтобы минимизировать время на обучение. Система должна поддерживать работу с различными устройствами (компьютеры, смартфоны планшеты) для обеспечения мобильности пользователей.
4. **Надежность:** Система должна обеспечивать сохранность данных и защиту от потери информации (например, с помощью регулярного резервного копирования). Необходимо реализовать механизмы восстановления системы в случае сбоя или аварии.
5. **Адаптивность:** Система должна быть способна адаптироваться к изменениям нагрузки и увеличиваться пропускную способность без снижения производительности. Необходимо предусмотреть возможность горизонтального расширения для обеспечения роста системы.
6. **Построение диаграммы использования**

Диаграмма взаимодействия пользователей — это эффективный способ визуализации процесса взаимодействия клиентов с сервисом. Она позволяет выявить ключевые возможности системы и роли участников, которые будут использовать ее. В этом разделе рассмотрим диаграмму взаимодействия пользователей для системы бронирования и продажи авиабилетов, а также проанализируем основных акторов и сценарии их действий. Актеры представляют собой пользователей или внешние системы, которые взаимодействуют с системой. В контексте нашей системы можно выделить следующих ключевых актеров:

1. **Администратор:** управляет системой, имеет доступ ко всем функциям. Отвечает за добавление и редактирование информации о билетах, клиентах и рейсах. Генерирует отчеты о продажах и работе сотрудников.
2. **Пассажир:** осуществляет поиски и покупку билетов. Проверяет наличие свободных мест на выбранных рейсах. Обрабатывает оплату и получает подтверждение бронирования.

Задачи, которые система должна выполнять в ответ на действия актеров. Для нашей системы можно выделить следующие функции:

1. **Добавление авиабилета:** Актер: Покупатель. Описание: Покупатель выбирает желаемый рейс и указывает количество билетов для покупки.
2. **Удаление авиабилета:** Актер: Администратор. Описание: Администратор отменяет несостоявшийся или отмененный рейс из системы.
3. **Добавление пассажира:** Актер: Покупатель. Описание: Покупатель регистрирует информацию о себе в системе при покупке билета.
4. **Удаление пассажира:** Актер: Администратор. Описание: Администратор удаляет информацию о пассажире из системы при необходимости.
5. **Осуществление продажи:** Актер: Покупатель. Описание: Покупатель выбирает товар, вводит количество и фиксирует продажу, обновляя информацию о количестве доступных билетов.
6. **Генерация отчетов:** Актер: Администратор. Описание: Администратор выбирает параметры для отчета (период, тип отчета) и генерирует отчет о продажах, остатках товаров или работе сотрудников.

**Осуществление продажи:** Актер: Покупатель. Описание: Покупатель выбирает товар, вводит количество и фиксирует продажу, обновляя информацию о количестве доступных билетов.

**Генерация отчетов:** Актер: Администратор. Описание: Администратор выбирает параметры для отчета (период, тип отчета) и генерирует отчет о продажах, остатках товаров или работе сотрудников.

Изображение выглядит как диаграмма, зарисовка, круг, текст

Автоматически созданное описание

## **Разработка сценария проекта**

Проектирование сценариев проекта — это важный этап, который помогает точно определить, как будущая система будет работать с людьми и какие задачи она будет решать. Эти сценарии позволяют представить себе, как система будет использоваться, и служат основой для дальнейшей разработки.

В данном проекте выделим несколько ключевых сценариев, которые описывают основные действия людей при работе с системой и как система на них реагирует. Каждый из этих сценариев включает в себя информацию об участниках процесса, последовательности действий и ожидаемых результатах.

Этот подход помогает лучше понять, как система будет работать в реальных условиях, и обеспечивает более эффективное планирование дальнейших работ по ее созданию.

**Сценарий 1: Добавление авиабилета**

**Участники:** Покупатель

**Описание:** Покупатель добавляет информацию об авиабилете в систему.

Последовательность действий:

1. Администратор входит в систему управления рейсами.
2. Администратор нажимает кнопку "Добавить новый рейс».

Администратор заполняет следующие поля:

* Маршрут (например, "Москва — Санкт-Петербург")
* Введите аэропорт отправления
* Введите аэропорт прибытия
* Введите цену
* Количество доступных мест
* Введите дату вылета

1. Администратор выбирает авиакомпанию, которая будет выполнять рейс.
2. Администратор добавляет пункт назначения и добавляет все данные о нём в систему.
3. Администратор подтверждает добавление рейса.
4. Система обновляет и сохраняет информацию о новом рейсе в базе данных.

**Ожидаемый результат:**

* Новый авиабилет успешно добавлен в систему и отображается в списке доступных билетов.
* Создан новый заказ с информацией о билете.
* Покупатель может просмотреть детали своего заказа и статус билета.

**Сценарий 2: Добавление рейса**

**Участники:** Администратор

**Описание:** Администратор добавляет все данные о рейсе в систему.

**Последовательность действий:**

1. Администратор входит в систему управления рейсами.
2. Администратор нажимает кнопку "Добавить новый рейс».
   * 1. Администратор заполняет следующие поля:

Маршрут (например, "Москва — Санкт-Петербург")

Введите аэропорт отправления

Введите аэропорт прибытия

Введите цену

Количество доступных мест

Введите дату вылета

Администратор выбирает авиакомпанию, которая будет выполнять рейс.

1. Администратор добавляет пункт назначения и добавляет все данные о нём в систему.
2. Администратор подтверждает добавление рейса.
3. Система обновляет и сохраняет информацию о новом рейсе в базе данных.

**Ожидаемый результат:** после успешного сохранения, система выводит подтверждение о добавлении нового рейса и может предложить администратору добавить дополнительные рейсы или вернуться в главное меню.

**Сценарий 3: Генерация отчетов**

**Участник:** Администратор

**Описание:** Администратор создает детальный анализ продаж за указанный период.

**Последовательность действий:**

1. Администратор выбирает функцию "Создание аналитического отчета".
2. Система предлагает выбрать временной диапазон для анализа (дата начала и дата окончания).
3. Администратор вводит дополнительные параметры для уточнения запроса (например, маршруты рейсов, авиакомпании) и подтверждает запрос.
4. Система формирует отчет и отображает его в удобном формате.

**Ожидаемый результат:** Интегрированный аналитический отчет о продажах успешно создан и доступен для детального изучения.

1. **Проектирование базы данных**

Проектирование базы данных для продажи авиабилетов включает в себя создание нескольких таблиц, каждая из которых отвечает за определенные аспекты работы приложения. Ниже представлены таблицы, атрибуты, а также взаимосвязи.



Рисунок 1 – таблица авиакомпаний

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 2 – таблица бронирования

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – таблица мест назначения

Изображение выглядит как текст, программное обеспечение, Шрифт, веб-страница

Автоматически созданное описание

Рисунок 4 – таблица рейсов

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 5 – таблица пассажиров

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 6 – взаимосвязи между таблицами

## **Реализация интерфейса приложения на Tkinter и Figma**

Рассматриваемый процесс касается разработки пользовательского интерфейса приложения для учета и продажи авиабилетов в специализированном десктопном приложении с использованием библиотеки Tkinter и инструментария Figma. Заранее до начала кодирования интерфейса был создан макет в Figma, что позволило визуализировать структуру и элементы интерфейса. Основные шаги включали:

1. Создание каркаса: Определены основные экраны приложения, такие как главное окно, страница добавления рейсов, страница клиентов и т. д.
2. Элементы управления: Добавлены текстовые поля, кнопки, выпадающие списки и другие элементы управления, необходимые для взаимодействия клиентом.
3. Стиль и цветовая палитра: Выбраны цветовая палитра и шрифты, которые будут использоваться в приложении, чтобы обеспечить единый стиль.

После завершения проектирования интерфейса в Figma началось его реализация на Tkinter. Было создано главное окно приложения и реализована функциональность для добавления рейсов и авиабилетов, просмотра каталога и других необходимых операций. Вот код, отвечающий за создание главного окна приложения:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 7 – код создания главного окна приложения

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание**

Рисунок 8 – код подключения через Figma строк приветствия

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание**

Рисунок 9 – код подключения через Figma кнопки пункты назначения

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание**

Рисунок 10 – код подключения через Figma кнопки рейсы и клиенты

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание**

Рисунок 11 – код подключения через Figma кнопок авиакомпании и бронирование

Для интеграграции дизайна из Figma в Tkinter применялись специализированные инструменты, способствующие автоматической генерации кода на основе созданных макетов. Процесс можно описать следующим образом:

1. Экспорт дизайна: Первым этапом было преобразование элементов интерфейса, созданных в Figma, в формат, совместимый с Tkinter.
2. Генерация кода: на втором этапе использовались разработанные инструменты, которые автоматически преобразовывали дизайн-макеты в соответствующий Python-код.

# **РЕАЛИЗАЦИЯ ПРОЕКТА СИСТЕМЫ**

## **Описание среды разработки**

*PyCharm* **—** это мощная интегрированная среда разработки (IDE) для языка программирования Python, разработанная компанией *JetBrains*. Она широко используется разработчиками благодаря своей функциональности, удобству и поддержке современных технологий. *PyCharm* предоставляет полный набор инструментов для разработки, отладки, тестирования и управления проектами.

# **Основные функции PyCharm:**

* Многоязычная поддержка: *PyCharm* поддерживает *Python*, а также другие языки, такие как *HTML, CSS, JavaScript* и *SQL*. Это делает его универсальным инструментом для разработки веб-приложений и многослойных проектов.
* Интеллектуальное авто дополнение: *PyCharm* предоставляет *IntelliSense*, который предлагает контекстно-зависимое авто дополнение кода, что значительно ускоряет процесс разработки и снижает количество ошибок.
* Встроенный отладчик: *PyCharm* имеет мощный отладчик, позволяющий устанавливать точки останова, просматривать значения переменных и выполнять пошаговую отладку кода.
* Интеграция с *Git* и другими системами контроля версий: *PyCharm* поддерживает работу с *Git, GitHub, Mercurial* и другими системами контроля версий, что упрощает управление версиями кода.

1. **Разработка программного модуля**

Разработка программного модуля курсовой работы на тему «Продажа авиабилетов» включает в себя несколько этапов и компонентов, которые обеспечивают функциональность для автоматизации процессов поиска, бронирования и управления авиабилетами.

1. **Реализация пользовательского интерфейса программы**

Для обеспечения продуктивного взаимодействия пользователя с модулем системы «Продажа авиабилетов» был разработан интуитивно понятный и удобный интерфейс. Структура интерфейса включает следующие элементы:

1. **Окно авторизации для пользователей**.
2. **Главное окно администратора**:

* Окно добавления рейсов.
* Окно просмотра информации обо всех пользователях системы.
* Окно просмотра заказов.
* Окно бронирования билета.
* Окно просмотра авиакомпаний.
* Окно просмотра пунктов назначения.

Проектирование пользовательского интерфейса включает не только правильное размещение элементов на экране, но и эффективное представление информации, а также учет действий пользователя. Это, в свою очередь, помогает улучшить пользовательский опыт и сделать его более комфортным.

Разработанный интерфейс пользователя, реализованный на языке программирования Python с использованием библиотеки Tkinter, показан ниже.

* Приведены поля для ввода логина и пароля, которые нужно заполнить, чтобы войти в систему.
* Под полями расположена кнопка «Войти», которая реализует вход в систему.
* После кнопки «Войти» идет кнопка «Зарегистрироваться», которая перенаправляет пользователя на окно регистрации.
* Окно авторизации и входа в систему (см. Приложение 2, Рисунок 12):
* Главное окно администратора (см. Приложение 2, Рисунок 13):
* Окно добавления рейса (см. Приложение 2, Рисунок 14).
* Окно удаления рейса (см. Приложение 2, Рисунок 15).
* Окно просмотра информации обо всех пользователях системы (см. Приложение 2, Рисунок 16).
* Окно управления пунктами назначения (см. Приложение 2, Рисунок 17).
* Окно управления Авиакомпаниями (см. Приложение 2, Рисунок 18).
* Окно управления бронями (см. Приложение 2, Рисунок 19).
* Окно выхода из приложения (см. Приложение 2, Рисунок 20).

## **Результат работы и тестирования**

В данном разделе будут описаны тестовые сценарии, с которыми можно будет подробнее ознакомиться в приложении 1.

Кейс "Авторизация пользователя": в данном кейсе тестируется функционал входа пользователя в систему. После того как пользователь ввел данные для входа и нажал на кнопку "Войти", если данные верны, пользователя перенаправляет на основное меню модуля, в противном случае отображается ошибка.

Кейс "Добавление маршрута рейса": в кейсе тестируется введение корректных данных и проверка уникальности идентификатора рейса.

Кейс "Удаление маршрута рейса": в кейсе тестируется введение корректных данных и проверяется наличие нужного рейса в доступных.

Кейс "Просмотр информации о рейсе": в кейсе тестируется корректное отображение информации о рейсе.

Кейс "Добавление авиабилета": в кейсе тестируется введение корректных данных и проверка соответствия требуемым критериям.

Кейс "Создание заказа": в кейсе тестируется корректное заполнение всех необходимых полей.

Кейс "Добавление билета в рейс": в кейсе тестируется введение корректных данных и проверка наличия нужного товара в ассортименте.

Таким образом, реализация проекта системы управления продажей авиабилетов продемонстрировала важность комплексного подхода к разработке программного обеспечения в сфере туристической индустрии. Успешная автоматизация процессов не только повышает эффективность работы авиакомпаний и турагентств, но и улучшает качество обслуживания клиентов. Применение современных технологий и методов разработки позволяет создавать удобные и функциональные решения, которые соответствуют требованиям времени и повышают конкурентоспособность на рынке.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсового проекта по теме «Разработка программного модуля системы «Продажа авиабилетов» были достигнуты поставленные цели и решены все задачи, связанные с созданием функционального и удобного программного решения. Проект был реализован с использованием языка программирования *Python* и библиотеки *Tkinter*, что позволило создать десктопное приложение с интуитивно понятным интерфейсом. Проектирование модуля включало разработку функциональных и нефункциональных требований, построение диаграммы использования и создание сценариев работы системы. Были разработаны сценарии для основных функций, таких как добавление рейсов, бронирование билетов и генерация отчетов. Интерфейс приложения был спроектирован в *Figma* и реализован с использованием *Tkinter*, что обеспечило удобство и наглядность взаимодействия с системой. На этапе отладки и тестирования был составлен план тестирования, включающий как позитивные, так и негативные сценарии. Проведенные тесты подтвердили корректность работы модуля, его устойчивость к ошибкам и соответствие требованиям пользователей. В результате была создана система, которая позволяет эффективно управлять продажами авиабилетов, автоматизировать процессы бронирования и предоставлять аналитические данные для принятия управленческих решений.

Таким образом, разработанный программный модуль является практически значимым решением, которое может быть внедрено в реальную работу авиационного агентства. Он обеспечивает высокий уровень автоматизации, удобство использования и надежность, что делает его конкурентоспособным на рынке программных решений для продажи авиабилетов.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

## Стандарты и законодательные материалы

1. Федеральный закон от 27 июля 2006 года № 152-ФЗ

Закон регулирует обработку и защиту персональных данных граждан. При разработке модуля туристические путевки необходимо учитывать требования этого закона для обеспечения конфиденциальности и безопасности данных пользователей, таких как имя и контактная информация.

1. ГОСТ Р ИСО/МЭК 27001–2012

Стандарт определяет требования к системам управления информационной безопасностью, включая оценку рисков и управление ими.

1. Официальная документация Tkinter

URL: <https://docs.python.org/3/library/tkinter.html>

Описание: Документация по библиотеке Tkinter для создания графического интерфейса в Python.

1. Figma — инструмент для проектирования интерфейсов

URL: <https://www.figma.com/>

Описание: Офлайн-документация и руководства по использованию Figma для проектирования интерфейсов.

1. Aviasales — мета поисковая система для авиабилетов

URL: <https://www.aviasales.ru/>

Описание: Исследование функциональности Aviasales для понимания современных подходов к продаже авиабилетов.

## Учебники и учебные пособия

1. Книга: "Изучаем Python. " (Эрик Мэтиз)

Издательство: Питер

Год: 2024

Описание: Подробное руководство по программированию на Python.

## Интернет-источники

1. Статья: "Создание баз данных с помощью SQLite в Python"

Источник: Хабр (Habr)

URL: <https://habr.com/ru/articles/488438/>

Описание: Пошаговое руководство по работе с SQLite в Python.

1. **Курс: "Tkinter. Курс по созданию графических интерфейсов"**

Источник: Young Linux

URL: <https://younglinux.info/tkinter/course>

1. Описание: Подробный курс по созданию графических интерфейсов с использованием библиотеки Tkinter.
2. **Metanit — учебный портал по программированию**

URL: <https://metanit.com/>

Описание: Подробные руководства по Python, Tkinter, базе данных SQLite и другим технологиям.

1. Статья: "Тестирование программного обеспечения"

Источник: GeekBrains

URL: <https://geekbrains.ru/>

Описание: Основы тестирования программного обеспечения, включая модульное и интеграционное тестирование.

1. Статья: "Создание макетов в Figma для разработчиков"

Источник: Medium

URL: <https://medium.com/>

Описание: Пошаговое руководство по созданию макетов интерфейсов в Figma.

1. **Видео: "Создание графического интерфейса с Tkinter"**

Источник: Яндекс. Видео

URL: <https://yandex.ru/video/preview/14249406639630984250>

Описание: Видеоурок по созданию графического интерфейса с использованием библиотеки Tkinter.

1. Статья: "Изучаем Python: Вводный курс"

Источник: Питер (издательство)

URL: [https://www.piter.com](https://www.piter.com/)

Описание: Подробное руководство по программированию на Python.

1. Статья: "Программирование на Python 3. Подробное руководство"

Источник: Символ-Плюс (издательство)

URL: [https://www.symbol-plus.ru](https://www.symbol-plus.ru/)

Описание: Подробное руководство по языку Python 3, включая примеры и практические задания.

1. Статья: "Программирование на Python для начинающих"

Источник: Питер (издательство)

URL: [https://www.piter.com](https://www.piter.com/)

Описание: Основы программирования на Python, включая создание графических приложений.

Монография

1. https://github.com/Pavel-43/build.git

**Приложение 1**

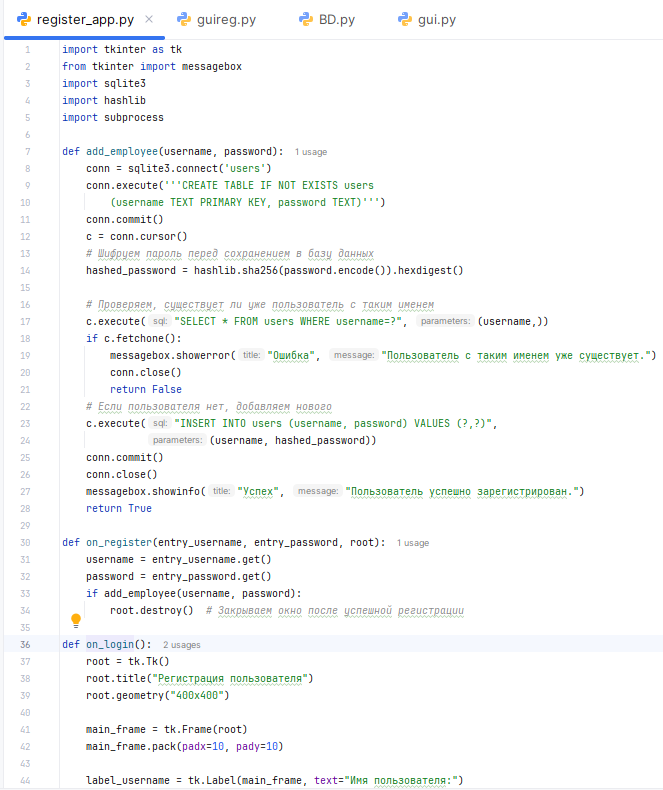
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID теста** | **Описание теста (тип)** | **Предусловия** | **Шаги для воспроизведения** | **Ожидаемый результат** | **Фактический результат** |
| **1** | Авторизация в системе (позитивный) | 1.Пользователь находится на странице входа в личный кабинет  2. Пользователь был ранее зарегистрирован в системе | 1. Ввести данные пользователя в поля «Логин» и «Пароль»  2. Нажать кнопку «Войти» | Пользователь перенаправляется на страницу личного кабинета | Пользователь перенаправляется на страницу личного кабинета |
| **2** | Регистрация (негативный) | 1. Пользователь находится на странице регистрации  2. Пользователь был ранее зарегистрирован в системе | 1. Ввести данные в поля «Имя», «Логин», «Пароль», «Подтвердите пароль»  2. Выбирает тип пользователя – «Покупатель» или «Администратор»  2. Нажать кнопку «Зарегистрироваться» | Пользователь на странице регистрации получает сообщение «Регистрация прошла успешно» | Пользователь на странице регистрации получает сообщение «Покупателя с таким именем не существует» или «Администратора с таким логином не существует» |
| **3** | Редактирование профиля (Негативный) | Пользователь находится на странице личного кабинета | 1. Нажать на кнопку «Профиль»  2. Получить страницу редактирования профиля  3. Нажать на кнопку «Изменить»  4. Внести данные в поля «Логин», «Пароль» и «Подтвердите пароль»  5. Нажать на кнопку «Сохранить» | Пользователю отображается сообщение «Данные сохранены» | Пользователю отображается сообщение «Пароли не совпадают» |
| **4** | Просмотр информации о рейсе (Позитивный) | Покупатель находится на странице рейсы «маршруты» | Покупатель нажимает на поисковую строку и вводит название рейса | Покупателю отображается строка с информацией о рейсе и появляется кнопка «Забронировать» | Покупателю отображается строка с информацией о рейсе и появляется кнопка «Забронировать» |
| **5** | Просмотр информации о маршрутах в категории (Позитивный) | Покупатель находится на странице «Показать рейсы» | 1.Нажать на кнопку «Аэропорт»  2. Нажать на кнопку с нужным аэропортом | 1. Покупатель перенаправляется на страницу «маршруты»  2. На странице отображается список книг, соответствующих выбранному аэропорту | 1. Покупатель перенаправляется на страницу «маршруты»  2. На странице отображается список книг, соответствующих выбранному аэропорту |
| **6** | Просмотр информации о продаже авиабилета (Негативный) | Покупатель находится на странице магазина «Компании» | Пользователь вводит дату вылета и направление | Покупателю отображается строка с информацией отображаются доступные рейсы с ценами «Забронировать» | Покупателю отображается строка «Рейс не найден» |
| **7** | Добавление авиабилета в забронированные (Позитивный) | 1. Покупатель находится на странице «Покупка билетов»  2. На данной странице находится поисковая строка | 1. Написать в поисковой строке название рейса  2. Страница обновляется  3. Нажать на кнопку «Забронировать» | 1. Страница «Рейсы» обновляется  2. Пользователю отображается сообщение «Билет забронирован» | 1. Страница «Рейсы» обновляется  2. Пользователю отображается сообщение «Билет забронирован» |
| **8** | Создание заказа (Негативный) | 1.Администратор находится на главной странице | 1.Администратор нажимает на кнопку «Создать заказ»  2.Появляется список товаров, добавленных в забронированные билеты покупателем  3.Администратор нажимает на кнопку «Оформить заказ»  4. Администратор вводит номер заказа  5. Администратор нажимает на кнопку «Создать» | 1.Появляется надпись «Заказ создан»  2. Данные заказа отправляются в базу данных | Появляется надпись «Заказ с таким номером уже нельзя создать!» |
| **9** | Добавление билетов в рейс (Позитивный) | 1.Администратор находится на главной странице | 1.Администратор нажимает на кнопку «Добавить билеты»  2. Ввести данные в поля «Маршрут», «Аэропорт отправления», «Аэропорт прибытия»  «Цена»  «Количество»  3.Нажать на кнопку «Добавить» | 1.Страница обновляется  2.Появляется надпись «Билеты добавлены в рейс»  3.При поиске покупателем билета с данным названием рейса появляется строка с информацией по билетам | 1.Страница обновляется  2.Появляется надпись «Билеты добавлены в рейс»  3.При поиске покупателем билета с данным названием рейса появляется строка с информацией по билетам |

Таблица 1– План-тест системы «Продажа авиабилетов»

# **Приложение 2**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание



Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 12 – Окно авторизации и входа в систему

Изображение выглядит как текст, снимок экрана, рыба, графический дизайн

Автоматически созданное описание



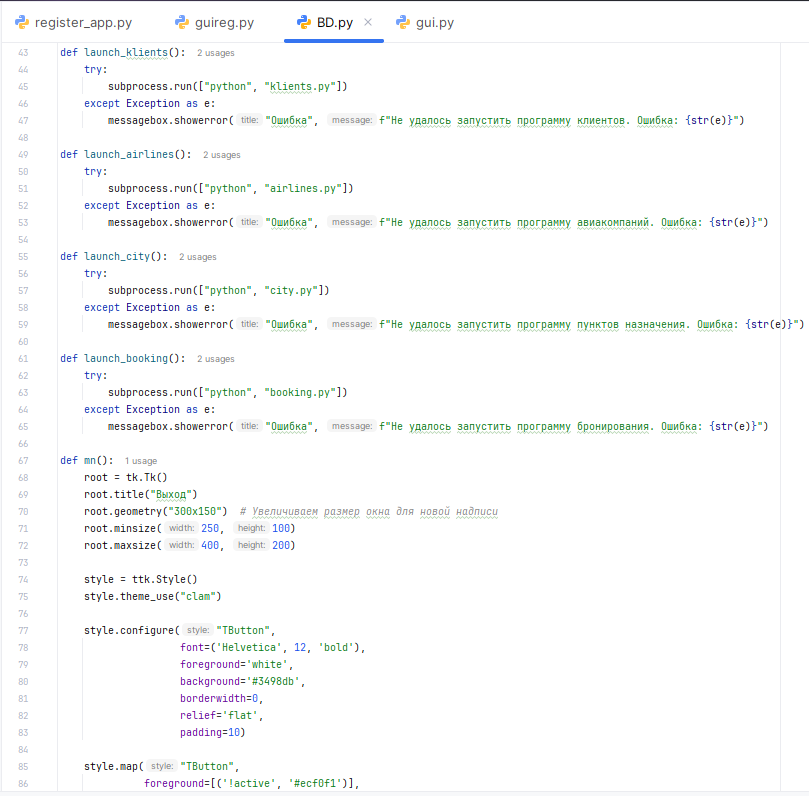


Рисунок 13 - Главное окно администратора

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 14 - Окно добавления рейса

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, логотип

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 15 - Окно удаления рейса из доступных

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

Рисунок 16 - Окно просмотра информации обо всех пользователях системы

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание

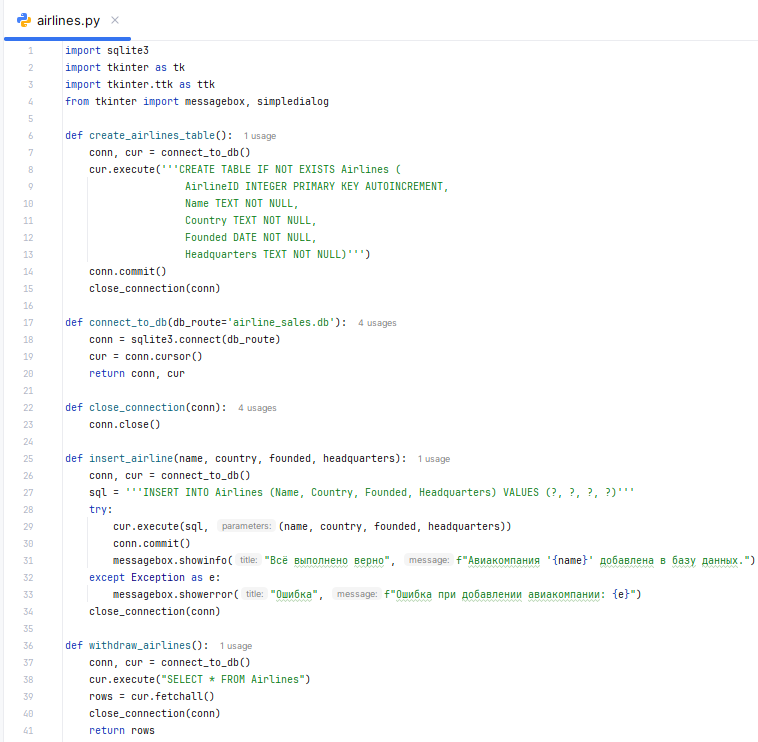
Изображение выглядит как текст, снимок экрана, документ

Автоматически созданное описание

Рисунок 17 – Окно управления пунктами назначения

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

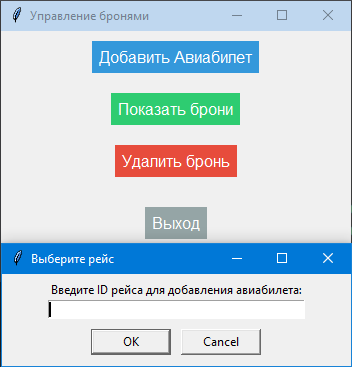
Автоматически созданное описание



Изображение выглядит как текст, снимок экрана, документ

Автоматически созданное описание

Рисунок 18 – окно управления авиакомпаний



Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, документ

Автоматически созданное описание

Рисунок 19 – Окно управления бронями

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

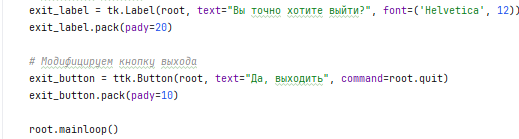


Рисунок 20 – Окно выхода из приложения