Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |
| Отделение прикладной математики и информатики |
| наименование кафедры |

Допускаю к защите

|  |  |
| --- | --- |
| Руководитель |  |
|  | подпись |
|  | З.А. Бахвалова |
|  | И.О. Фамилия |

|  |
| --- |
| Проектирование и разработка базы данных |
| «Торговая Организация» |
| наименование темы |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по дисциплине

|  |
| --- |
| Информатика |

|  |  |  |
| --- | --- | --- |
| 1.000.00.00 – ПЗ | | |
| обозначение документа | | |
| Выполнил студент |  | ИСТб-20-3 | |  |  | |  | Е.А. Житов |
|  |  | шифр | |  | подпись | |  | И.О. Фамилия |
| Нормоконтроль |  |  | |  |  | |  | З.А. Бахвалова |
|  |  |  | |  | подпись | |  | И.О. Фамилия |
| Курсовая работа защищена с оценкой | | | | |  | | | |

Иркутск 2023 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **По курсу** | Технологии разработки программных комплексов | | | | | | | | | | | | | | | | |
| **Студенту** | Житову Е.А. | | | | | | | | | | | | | | | | |
|  | (фамилия, инициалы] | | | | | | | | | | | | | | | | |
| **Тема работы** | | Онлайн-регистрация авиабилетов | | | | | | | | | | | | | | | |
| **Исходные данные** | | Создать информационную систему для совершения онлайн-регистрации авиабилетов. Проанализировать предметную область и реализовать систему в соответствии с жизненный циклом ПО. | | | | | | | | | | | | | | | |
| **Рекомендуемая литература** | | | | | | |  | | | | | | | | | | |
| 1. СТО 005-2015 СИСТЕМА МЕНЕДЖМЕНТА КАЧЕСТВА. Учебно-методическая деятельность. Оформление курсовых проектов (работ) и выпускных квалификационных работ технических специальностей (http://www.istu.edu/structure/57/2506/) | | | | | | | | | | | | | | | | | |
|  | | |  | | | | | | | | | | | | | | |
| Дата выдачи задания | | | « | 1 | » | мая | | | | | 2023 г. | | | | |
|  | | | | | | | | | | | | | | | | | |
| Задание получил | | | | | | | |  | | | | |  | | Е.А. Житов | | |
|  | | | | | | | | подпись | | | | |  | | И.О. Фамилия | | |
|  | | | | | | | | |  |  | |  | |  | | |  | |
| Дата представления работы руководителю | | | | | | | | | « | 21 | | » | | мая | | | 2023г. |
|  | | | | | | | | | | | | | | | | | | |
| Руководитель курсовой работы | | | | | | | |  | | | | |  | | З.А. Бахвалова | | |
|  | | | | | | | | подпись | | | | |  | | И.О. Фамилия | | |

**Содержание**

[Формирование требований 4](#_Toc134525984)

[1 Описание предметной области 4](#_Toc134525985)

[2 Описание проблемы 4](#_Toc134525986)

[3 Цель 4](#_Toc134525987)

[4 Словари 5](#_Toc134525988)

[5 Постановка задачи 8](#_Toc134525989)

[6 Функциональные требования ПО 9](#_Toc134525990)

[Анализ функциональных требований 10](#_Toc134525991)

[1 Модель предметной области (IDEF0) 10](#_Toc134525992)

[2 Варианты использования UML 12](#_Toc134525993)

[3 DFD - диаграмма потоков данных 13](#_Toc134525994)

[4 Концептуальная модель хранилища данных 14](#_Toc134525995)

[5 Описание вариантов использования 15](#_Toc134525996)

[6 Диаграммы состояний 19](#_Toc134525997)

[7 Описание интерфейса 20](#_Toc134525998)

[Проектирование 22](#_Toc134525999)

[1 Инструменты 22](#_Toc134526000)

[2 Описание алгоритма работы приложения 25](#_Toc134526001)

[3 Интерфейс 26](#_Toc134526002)

[Тестирование 28](#_Toc134526003)

[Исходный код авто тестов 28](#_Toc134526004)

[Документация пользователя 29](#_Toc134526005)

[Заключение 30](#_Toc134526006)

[Список использованных источников 31](#_Toc134526007)

## Формирование требований

## 1 Описание предметной области

Аэропорт, занимающейся обслуживанием авиаперевозок. Пассажир, купивший авиабилет использует пришедший ему номер бронирования для самостоятельной онлайн-регистрации. Для совершения авиаперелета, пассажиру необходимо купить билет, зарегистрироваться и получить посадочный талон. Полученный посадочный талон, пассажир предоставляет сотруднику безопасности. Сотрудник безопасности проверяет данные на талоне, паспорт и самого пассажира. После успешной проверки пассажир допускается в зону посадки.

## 2 Описание проблемы

Компанией заказчиком является аэропорт. Аэропорт заключил новые контракты с авиакомпаниями. Количество рейсов будет увеличено. Проблемой является ограниченное пространство в зоне регистрации для планируемого в ближайшее время увеличения пассажиропотока. Аэропорт не имеет возможности расширить зону регистрации.

## 3 Цель

Сократить время нахождения пассажиров в зоне регистрации, исключив столпотворение людей и очереди путем реализации механизма **онлайн-регистрации** для пассажиров без багажа.

## 4 Словари

Таблица 1 - Словарь предметной области

|  |  |  |
| --- | --- | --- |
| **№** | **Сущность** | **Описание** |
| 1 | Номер бронирования | Уникальная последовательность символов, которая позволяет однозначно определить пассажира и его рейс. |
| 2 | Номер рейса | Уникальная последовательность символов, которая позволяет однозначно определить рейс. |
| 2 | Маршрутный  Лист | Квитанция (Чек), подтверждающий факт прохождения регистрации. Содержит в себе номер бронирования, паспортные данные пассажира, информацию о рейсе. |
| 3 | Посадочный Талон | Посадочный талон – документ, дающий право пассажиру на посадку в самолет. Чтобы получить талон необходимо предоставить соответствующий маршрутный лист |
| 5 | Онлайн-Регистрация | Процесс регистрации пассажира на рейс, не требующий нахождение пассажира в зоне регистрации. |
| 6 | Зона Регистрации | Специально отведенное место в аэропорте для осуществления регистрации пассажиров. |
| 8 | Паспортные данные | Паспортные данные, предъявляемые пассажиром |
| 9 | Пассажир | Лицо, купившее билет для совершения перелета |
| 10 | Структура посадочного талона | Специальный формат, структура, которой должны соответствовать все посадочные талоны будто оформление или предоставляемая информация |
| 11 | Информация о рейсе | Специальный формат, структура, которой должны соответствовать все рейсы будто оформление или предоставляемая информация |
| 12 | Структура номера бронирования | Специальный формат, структура, которой должны соответствовать все номера бронирования будто оформление или предоставляемая информация |
| 13 | Структура номера рейса | Специальный формат, структура, которой должны соответствовать все номера рейсов будто оформление или предоставляемая информация |
| 14 | Место | Место в самолете. Каждое место пронумеровано и имеет свой номер от 0 до 255 включительно |

Таблица 2 - Словарь по Абботу

|  |  |  |
| --- | --- | --- |
| **№** | **Сущность** | **Глагол** |
| 1 | Номер бронирования | Вводить |
| 2 | Маршрутный  Лист | Сгенерировать, Скачать, Распечатать, Отдать |
| 3 | Посадочный Талон | Распечатать, Отдать |
| 5 | Онлайн-Регистрация | Пройти |
| 6 | Зона Регистрации | Избегать |
| 8 | Паспортные данные | Предоставлять, Вводить |
| 9 | Пассажир | Регистрироваться |
| 10 | Структура посадочного талона | Следовать, Соответствовать |
| 11 | Информация о рейсе | Следовать, Соответствовать |
| 12 | Структура номера бронирования | Следовать, Соответствовать |
| 13 | Структура номер рейса | Следовать, Соответствовать |
| 14 | Место | Выбирать |

Таблица 3 - Объектно-ориентированный словарь

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Сущность** | **Данные** | **Методы** |
| 1 | Номер бронирования | Номер бронирования | Вводить |
| 2 | Маршрутный  Лист | Паспортные данные  Информация о рейсе | Скачать, Отдать |
| 3 | Талон | Паспортные данные  Информация о рейсе  Номер места, выхода | Получить, Отдать |
| 5 | Онлайн-Регистрация | Номер бронирования Паспортные данные | Пройти |
| 6 | Зона Регистрации | - | Избегать |
| 7 | Паспортные данные | * Фамилия; * Имя; * Отчество; * Пол; * Дата рождения; * Серия паспорта; * Номер паспорта. | Вводить, Предоставлять |
| 8 | Пассажир | * Паспортные данные; * Паспорт; * Маршрутный лист; * Посадочный талон. | Регистрироваться |
| 9 | Структура посадочного талона | * Паспортные данные * Информация о рейсе | Следовать, Соответствовать |
| 10 | Информация о рейсе | * Номер рейса; * Название авиакомпании; * Город отправления и город прибытия * Аэропорт отправления * Аэропорт прибытия * Дата и время отправления и прибытия | Предоставлять |
| 14 | Место | Номер места | Выбирать |

|  |  |  |  |
| --- | --- | --- | --- |
| 11 | Структура номера бронирования | Последовательность, состоящее из 36 символов цифр и английских букв (от a до f) разделенных между собой через знак тире в соотношении **8-4-4-4-12**.  Пример:**123e4567-e89b-12d3-a456-426655440000** | Следовать, Соответствовать |
| 12 | Структура номера рейса | Последовательность, состоящее из 36 символов цифр и английских букв (от a до f) разделенных между собой через знак тире в соотношении **8-4-4-4-12**.  Пример:**123e4567-e89b-12d3-a456-426655440000** | Следовать, Соответствовать |

## 5 Постановка задачи

Создать механизм онлайн регистрациидля аэропорта.

Реализовать возможность пассажиру зарегистрироваться на рейс по его номеру бронирования, используя паспортные данные. Предусмотреть возможность выбора места в зависимости от типа авиабилета (эконом, бизнес). Некоторые авиабилеты могут не предоставлять возможность выбора места или предоставлять, но за дополнительную плату.

Реализовать возможность зарегистрированному пассажиру самостоятельно получить маршрутный лист после прохождения онлайн регистрации.

Пассажир находит по номеру бронирования свой рейс, указывает свои паспортные данные, выбирает место в самолете и подтверждает регистрацию. Если пассажир ввел неверный номер бронирования, то он не сможет найти свой рейс и продолжить регистрацию. Если пассажир на этапе ввода паспортных данных, указал неверные данные, то на 3-ем этапе паспортного контроля сотрудник безопасности не пропустит его, так как он не будет совпадать паспортные данные. После ввода паспортных данных, пассажиру будет предоставлена возможность выбора места (Если авиабилет предусматривает выбор места). Если пассажир не выбрал место, то место после подтверждения регистрации назначается автоматически. Последний этап регистрации – это её подтверждение. После ввода номера бронирования, паспортные данных, места (если возможно), пассажир обязан подтвердить регистрацию путем нажатия на соответствующий элемент интерфейса. В противном случае регистрация не будет действительной и маршрутный лист будет недоступен.

После регистрации пассажир скачивает маршрутный лист, который пассажир подносит к специальному терминалу в аэропорту для получения посадочного талона.

## 6 Функциональные требования ПО

**Онлайн регистрация**

Модуль должен предоставить пассажиру возможность произвести онлайн регистрацию. Пассажиру необходимо выполнить следующий порядок действий:

1. Ввести номер бронирования и получить информацию о своем рейсе;
2. Ввести паспортные данные и пройти проверку;
3. Выбрать место (Если пассажира не устроило место по умолчанию)
4. Завершить регистрацию;

По завершении регистрации, пассажиру необходимо предоставить возможность скачать маршрутный лист.

Предусмотреть исключительные ситуация, которые могут возникнуть в процесса онлайн-регистрации.

# Анализ функциональных требований

## 1 Модель предметной области (IDEF0)

Данная модель IDEF0 описывает процесс «Онлайн-регистрации пассажира».



Рисунок 1 – Модель IDEF0. Уровень А0. (Онлайн-регистрация)

Пассажир вводит номер бронирования, который пассажир получает при покупке авиабилета и также свои паспортные данные для прохождения онлайн-регистрации.

База данных рейсов хранит информацию о рейсах и зарегистрированных пассажиров. Пассажир является лицом, который проходит данную онлайн-регистрацию. Терминал – устройство, находящееся в аэропорту, которое распечатывает посадочный талон.

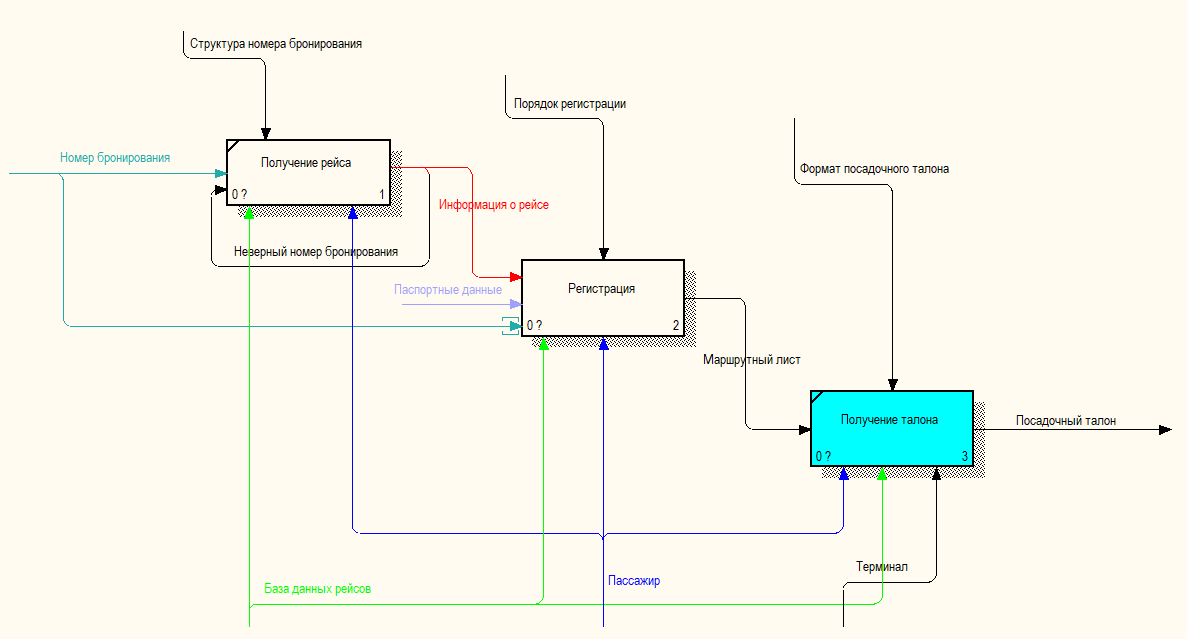


Рисунок 2 – Модель IDEF0. Уровень А1. (Онлайн-регистрация)

Процесс А1 «Онлайн-регистрация» имеет детализацию, представленную на рисунке 2.

В начале процесса регистрации, пассажир вводит номер бронирования, чтобы получить информацию о рейсе, проверить свой рейс (в случае необходимости). Далее, пассажир инициирует процесс регистрации, подавая на вход информацию о рейсе и паспортные данные, которые он ввел. По завершении регистрации, пассажир получает маршрутный лист, который необходимо предоставить терминалу в аэропорту для распечатывания посадочного талона. Получение талона выходит за границы нашей предметной области, поэтому данный элемент диаграммы помечен синим цветом.

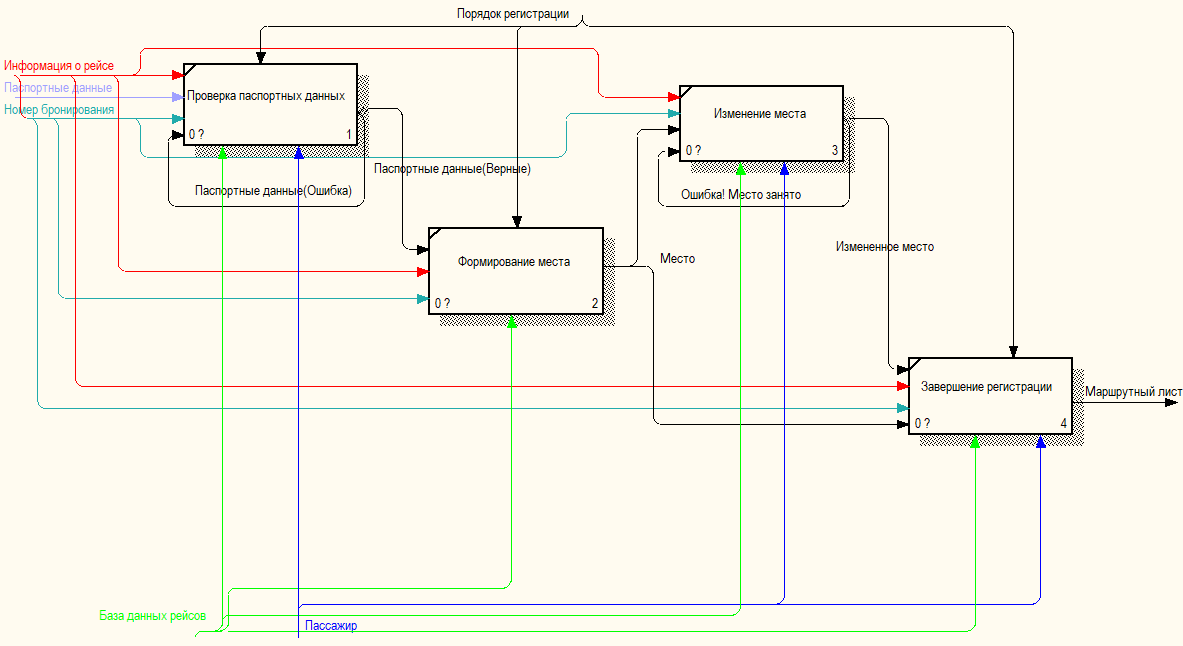


Рисунок 3 – Модель IDEF0. Уровень А2. (Регистрация)

Процесс А2 «Регистрация» имеет детализацию, представленную на рисунке 3.

Непосредственный процесс регистрации пассажира является ключевым и самым важным процессом во всей системе. Порядок регистрации пассажира следующий:

Пользователь вводит паспортные данные, необходимые при регистрации. Далее система проверяет введенные паспортные данные на то, что они принадлежат паспортным данным, которые были указаны при покупке билета. Если паспортные данные не совпали, то система не пропустит пассажира дальше пока он не введет корректные данные. Если паспортные данные введены верно, пассажир переходит на этап формирования места. На данном этапе пассажиру предлагается свободное место, однако пассажир может его изменить в случае необходимости. Пассажир может выбрать только незанятые места. Чтобы завершить регистрацию, пассажиру необходимо выполнить подтверждение. Это может быть интерактивный элемент интерфейса. Лишь только после завершения регистрации пассажир считается зарегистрированным и ему предоставляется возможность скачать маршрутный лист.

## 2 Варианты использования UML

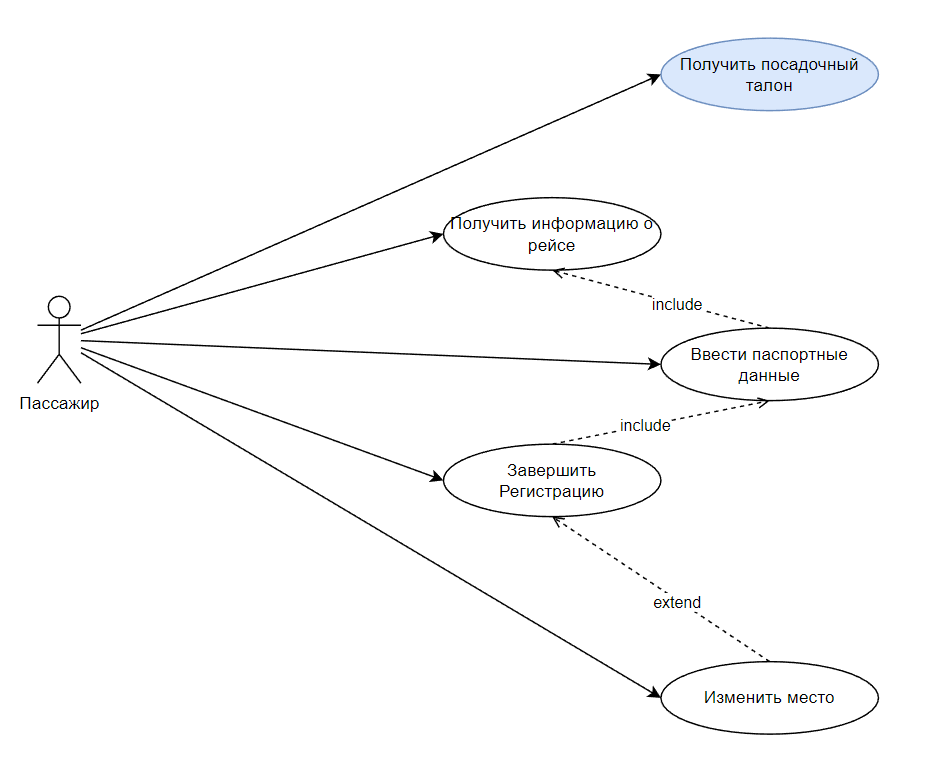


Рисунок 4 – Диаграмма вариантов использования.

Пассажир может выполнять функции, которые отображены на рисунке 4.

На диаграмме вариант использования *«Получить посадочный талон»* выходит за границупредметной области*.* Вариант использования «Получить посадочный талон», помеченный синим цветом, выходит за границы предметной области и не подлежит рассмотрению.

# 3 DFD - диаграмма потоков данных

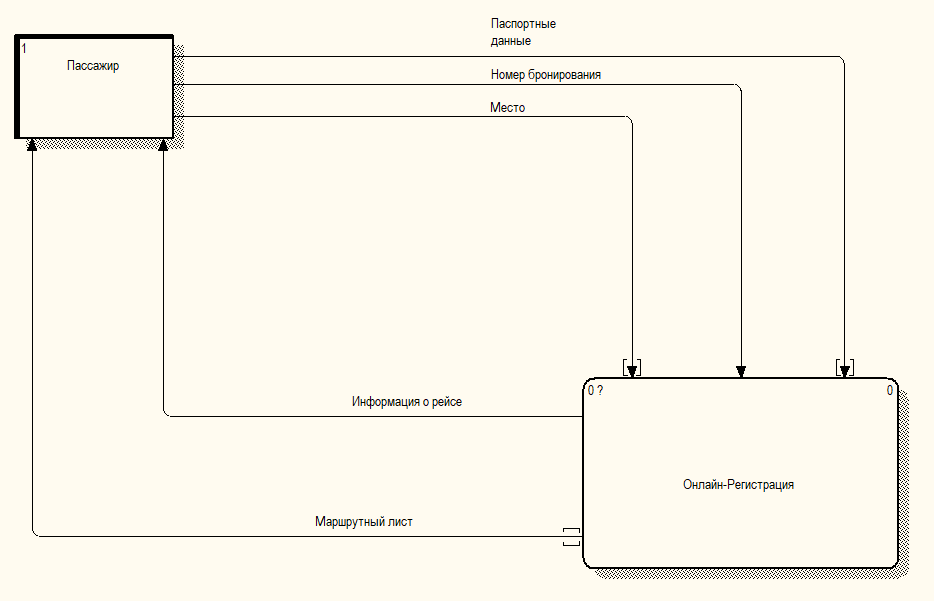


Рисунок 5 – Диаграмма потоков данных. A0

В диаграмме потоков данных была выделена внешняя сущность – пассажир, которая инициализирует процесс *«Онлайн-Регистрация»*

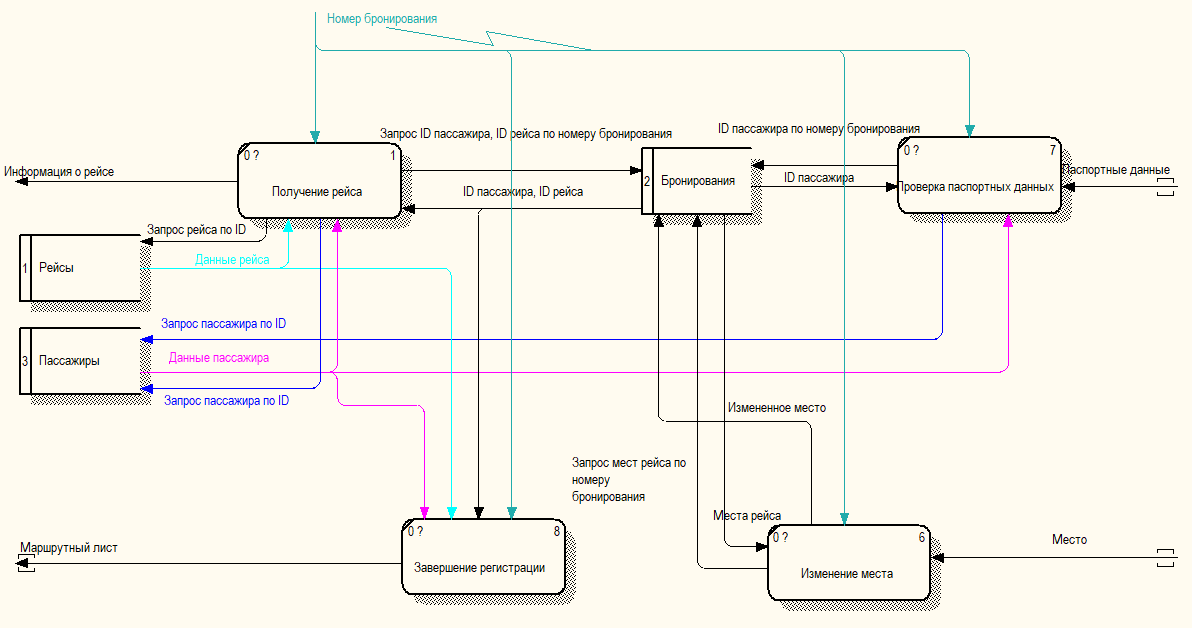


Рисунок 6 – Диаграмма поток данных. Уровень А1.

На рисунке 6 отображена детализация процесса *«Онлайн-Регистрация»*

## 4 Концептуальная модель хранилища данных

Основываясь на DFD диаграмму была разработана Концептуальная модель хранилища данных

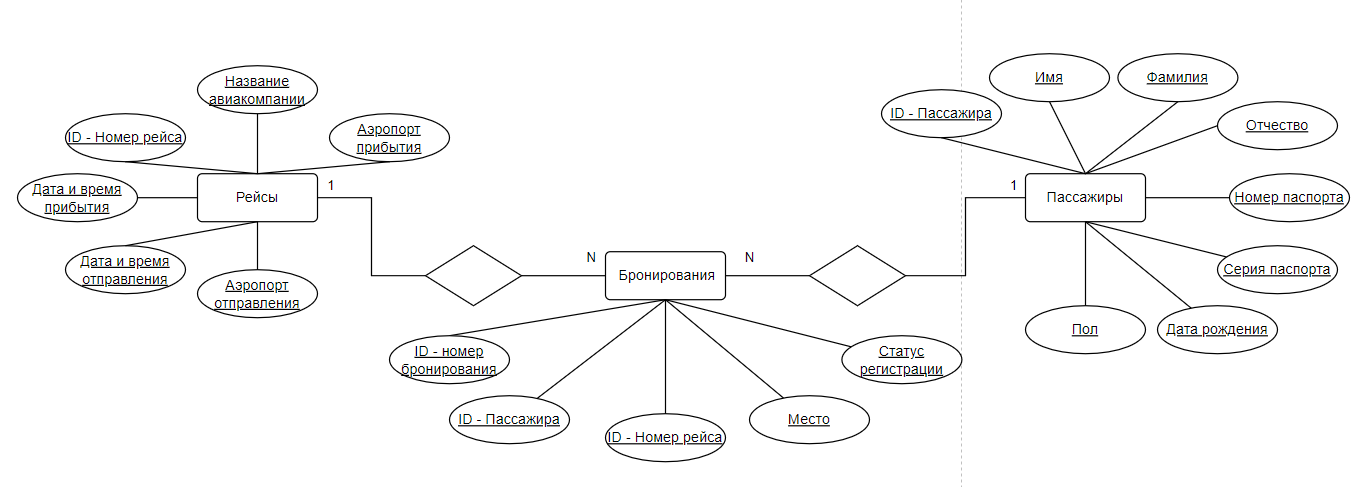


Рисунок 7 – ER-модель.

На рисунке 7 описаны главные сущности системы: Рейсы, Пассажиры, Бронирования. У каждой сущности есть свои атрибуты.

Сущность *Рейсы* и сущность *Пассажиры* имеют отношение **N:N**, пассажир может быть зарегистрирован одновременно на несколько рейсов и на один рейс может быть зарегистрировано несколько пассажиров.

Сущность *Бронирования* являются промежуточной сущностью, которая декомпозирует отношение **N:N** между *Рейсы* и Пассажиры. Сущность *Бронирования* обязательна, для выполнения требования ***однозначное определение по номеру бронирования рейс и пассажира в данном рейсе.***

## 5 Описание вариантов использования

**1 Вариант использования «Получить маршрутный лист».**

Спецификация варианта использования «Получить маршрутный лист».

**Цель:** Получить маршрутный лист.

**Активные субъекты:** Пассажир.

**Краткое описание:** Активный субъект имеет возможность скачать маршрутный лист.

**Предусловия:** необходимо *«Завершить регистрацию»*.

**Постусловия:** нет.

**Основной поток событий:**

1. Пассажир запрашивает на скачивания маршрутного листа
2. Приложение запрашивает данные у БД для создания маршрутного листа.
3. Приложение создает файл маршрутного листа и отдает пассажиру.

**Альтернативные потоки событий:** нет.

**Сценарий обработки ошибок:** нет.

**Специальные требования:** нет.

**Дополнительные замечания:** нет.

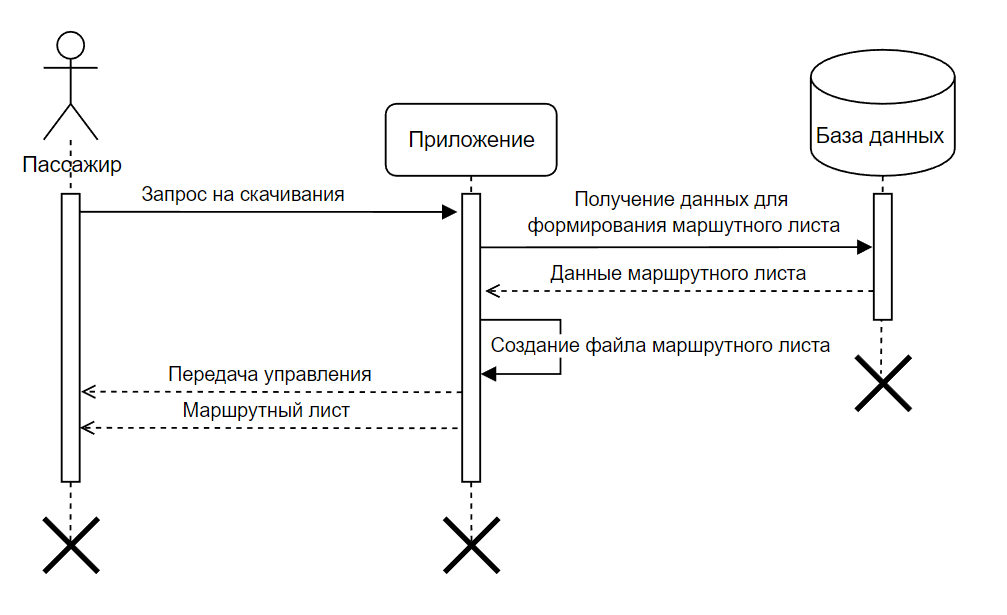


Рисунок 8 – Диаграмма последовательностей варианта использований

*«Получить маршрутный лист»*.

**2 Вариант использования «Завершить регистрацию».**

Спецификация варианта использования «Завершить регистрацию».

**Цель:** получить отсортированный список товаров.

**Активные субъекты:** Пассажир.

**Краткое описание:** Активный субъект имеет возможность завершить регистрацию.

**Предусловия:** необходимо *«Ввести паспортные данные»*.

**Постусловия:** нет.

**Основной поток событий:**

1. Пассажир завершает регистрацию

**Альтернативные потоки событий:**

1. Пассажир может изменить по своему желанию место в самолете.
2. Вывод «Место занято» если пассажир решил выбрать занятое место.

**Сценарий обработки ошибок:**

**Специальные требования:** нет.

**Дополнительные замечания:** нет.

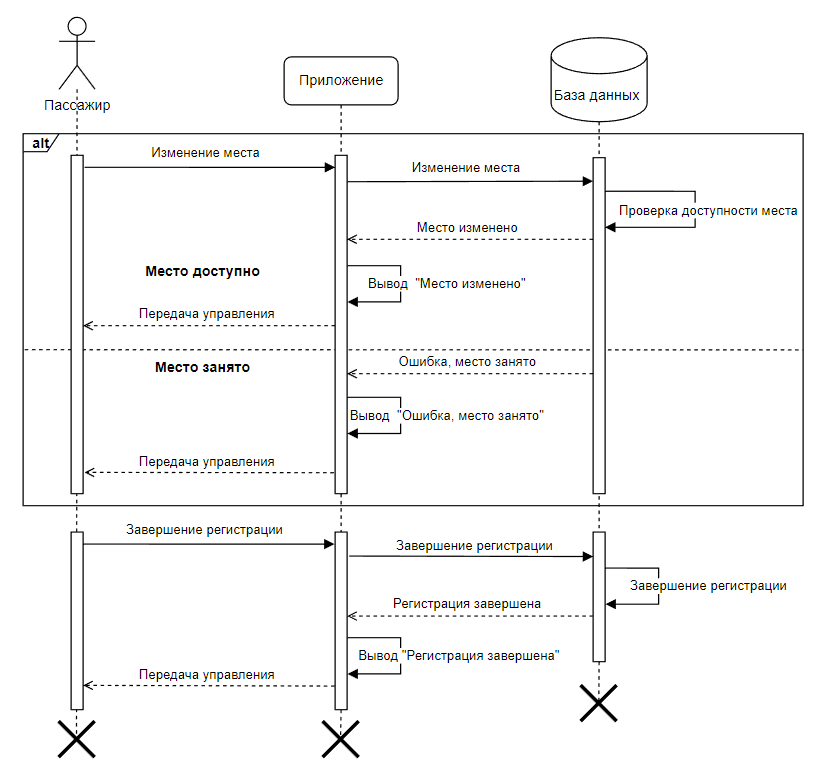


Рисунок 9 – Диаграмма последовательностей варианта использований *«Завершить регистрацию»*.

**3 Вариант использования «Ввести паспортные данные».**

Спецификация варианта использования «Ввести паспортные данные».

**Цель:** Пройти проверку паспортных данных.

**Активные субъекты:** Пассажир.

**Краткое описание:** Активный субъект обязан пройти проверку паспортных данных в процессе онлайн-регистрации.

**Предусловия:** необходимо *«Получить информацию о рейсе»*.

**Постусловия:** нет.

**Основной поток событий:**

1. Пассажир вводит паспортные данные

**Альтернативные потоки событий:**

2. Ошибка, пассажир не прошел проверку паспортных данных.

**Сценарий обработки ошибок:**

**Специальные требования:** нет.

**Дополнительные замечания:** нет.

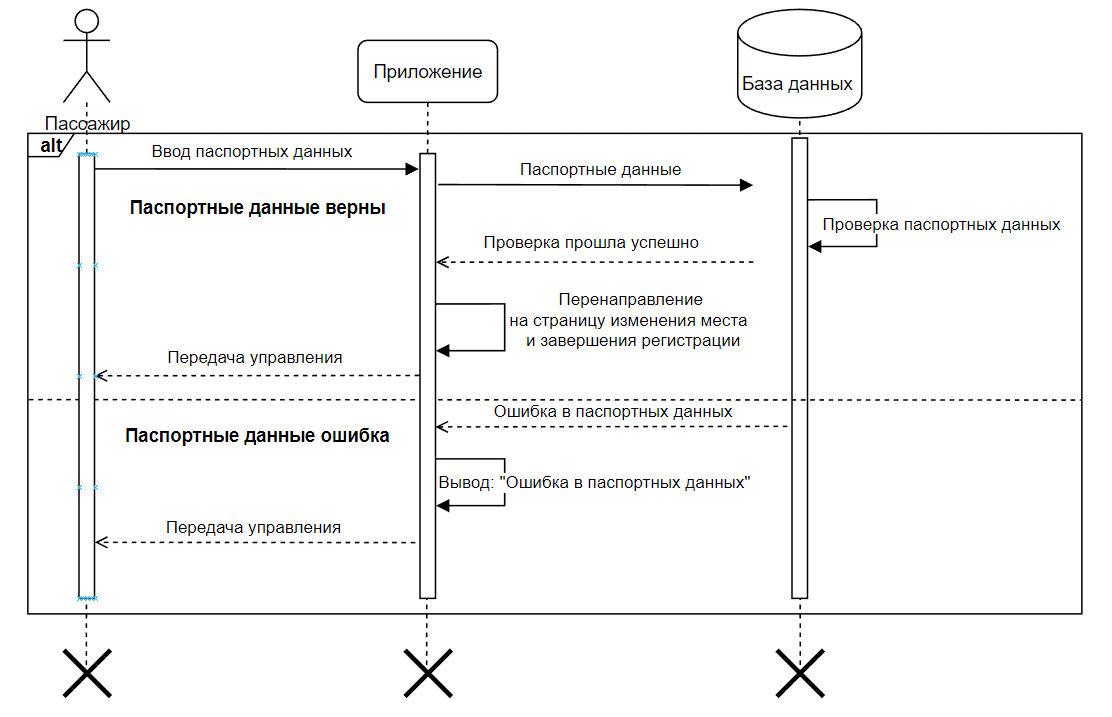


Рисунок 10 – Диаграмма последовательностей варианта использований

*«Ввести паспортные данные»*.

**4 Вариант использования «Получить информацию о рейсе».**

Спецификация варианта использования «Получить информацию о рейсе».

**Цель:** Получить рейс, для прохождения онлайн-регистрации.

**Активные субъекты:** Пассажир.

**Краткое описание:** Активный субъект обязан получить рейс, чтобы продолжить онлайн-регистрацию

**Предусловия:** нет.

**Постусловия:** должен быть выполнен вариант использования

«Ввести паспортные данные».

**Основной поток событий:**

1. Пассажир вводит номер бронирования
2. Пассажир получает информацию о рейсе

**Альтернативные потоки событий:**

1. Ошибка, по номеру бронирования не найден рейс.

**Сценарий обработки ошибок:**

**Специальные требования:** нет.

**Дополнительные замечания:** нет.

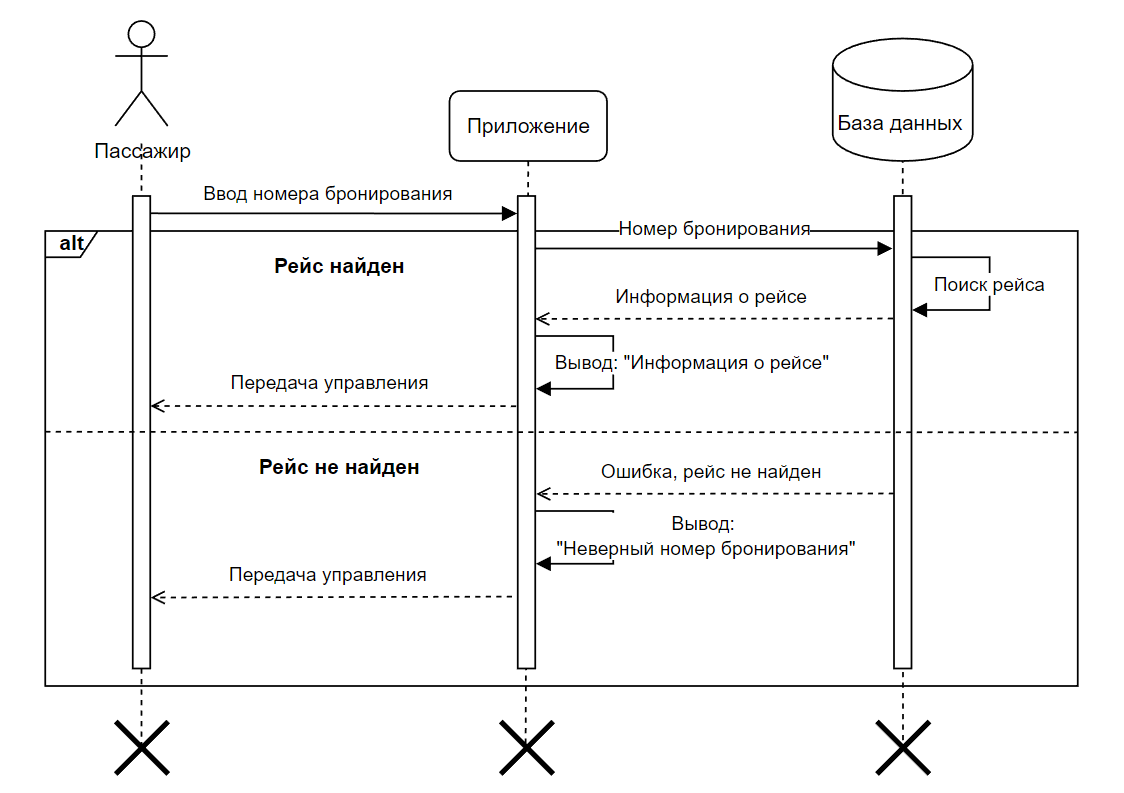


Рисунок 11 – Диаграмма последовательностей варианта использований

*«Получить информацию о рейсе»*.

## 6 Диаграммы состояний

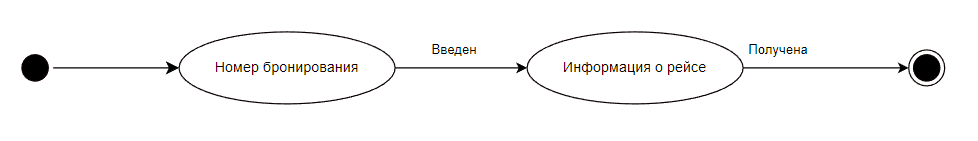


Рисунок 12 – Диаграмма состояний. Получения информации о рейсе

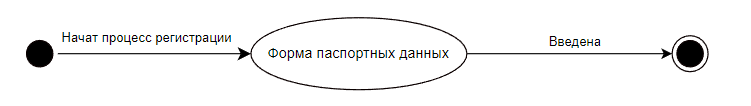


Рисунок 13 – Диаграмма состояний. Ввода паспортных данных

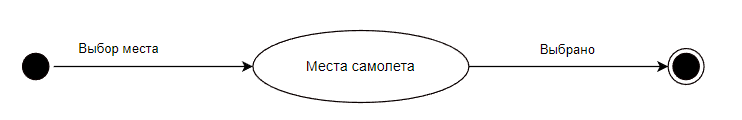


Рисунок 14 – Диаграмма состояний Изменения места

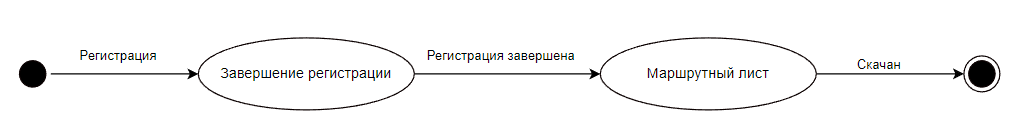


Рисунок 15 – Диаграмма состояний Завершение регистрации.

## 7 Описание интерфейса

Таблица 4 – Описание элементов интерфесов

|  |  |  |  |
| --- | --- | --- | --- |
| **Страница** | **Функция** | **Элемент интерфейса** | **Описание элемента** |
| Начальная страница | Ввести номер бронирования | Поле ввода | Поле, в которое пассажир вводит номер бронировния |
| Информация о рейсе | Посмотреть информацию о рейсе | Таблица с данными рейса | -- |
| Зарегистрироваться | Кнопка | - |
| Проверка паспортных данных | Проверка паспортных данных | Форма |  |
| Регистрация | Изменить место | Салон самолета, с кнопками выбора места | По умолчанию случайно свободное место уже выбрано |
| Подтверждение | Кнопка | - |
| Маршрутный лист | Скачать маршрутный лист | Кнопка | - |

Элементы интерфейса были сформированы на основе диаграммы вариантов использования с учетом выполнения всех разработанных функциональных требований.

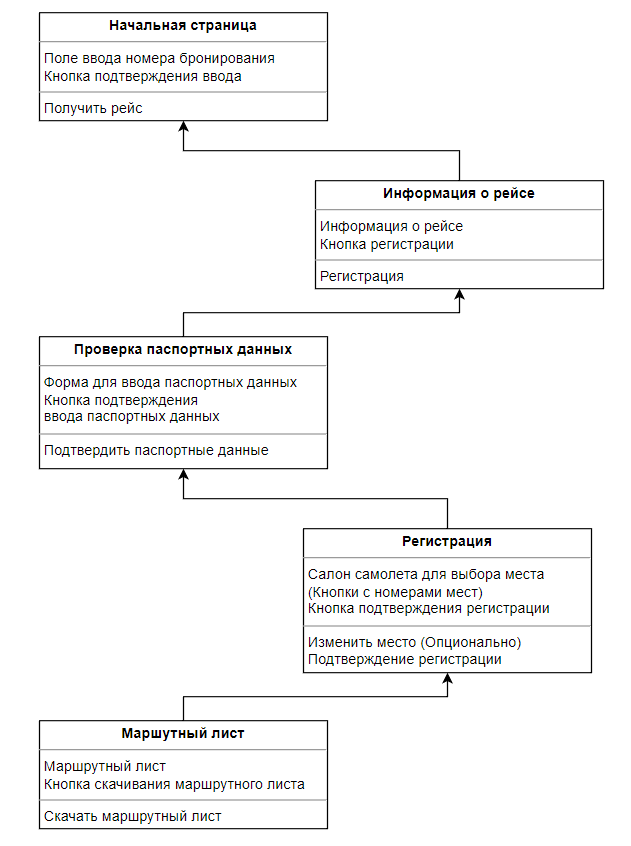


Рисунок 16 – Диаграмма интерфейсных классов

На рисунке 16 изображена диаграмма интерфейсных классов, где отображены главные составляющие интерфейса. Есть 5 основных страницы: Начальная страница, Информация о рейсе, Проверка паспортных данных, регистрация, маршрутный лист.

## Проектирование

## 1 Инструменты

ПО является веб приложением, которое можно будет открыть в браузере для прохождения онлайн регистрации. Будет использован инструментарий программных средств:

1. **Visual Studio Code** – удобный легкий текстовый редактор кода, благодаря расширениям позволяет легко и просто настроить редактор под данный проект.
2. **Python** – язык программирования. Является самым быстрым языком при прототипировании ПО. Позволяет в кратчайшие сроки реализовать MVP для заказчика.
3. **Flask** – фреймворк для написания API приложения. Требует очень низкий порог вхождения для быстрого написания и поддержки API.
4. **PyTest** – фреймворк для написания авто тестов. Является одним из самых популярных и часто используемых.
5. **PostgreSQL** – реляционная СУБД. Бесплатная с открытым исходным кодом. Необходима для хранения и управления данными.
6. **Selenium** – библиотека, позволяющая провести тестирование интерфейса. Бесплатная, открытый исходный код.

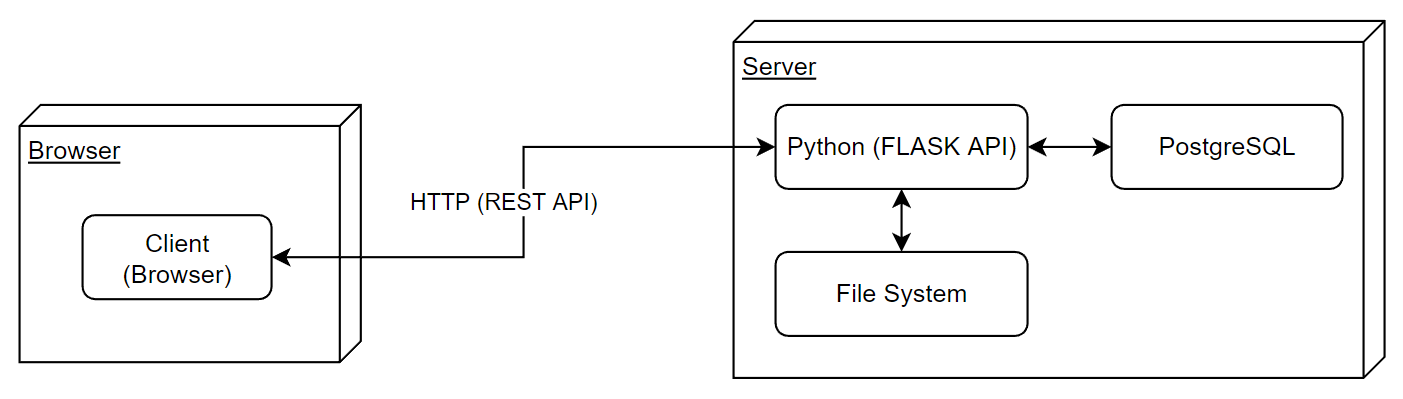


Рисунок 17 – диаграмма развертывания приложения

На диаграмме развертывания представлена архитектура ПО. Клиент (браузер) используя протокол http получает доступ к приложению, загружая файлы для отображения интерфейса. Клиент также может посылать запросы к API, с помощью которого происходит манипуляция с данными и работой с СУБД. FLASK API – сервер принимающий запросы от клиента. Его основная задача выдача генерация файла и его выдача и манипуляция с данными БД.

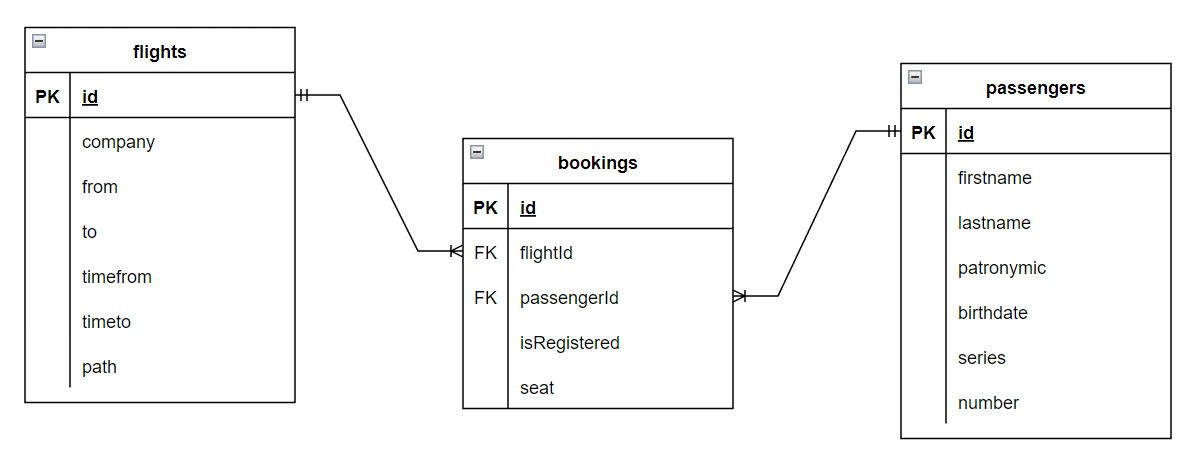


Рисунок 18 – логическая модель базы данных

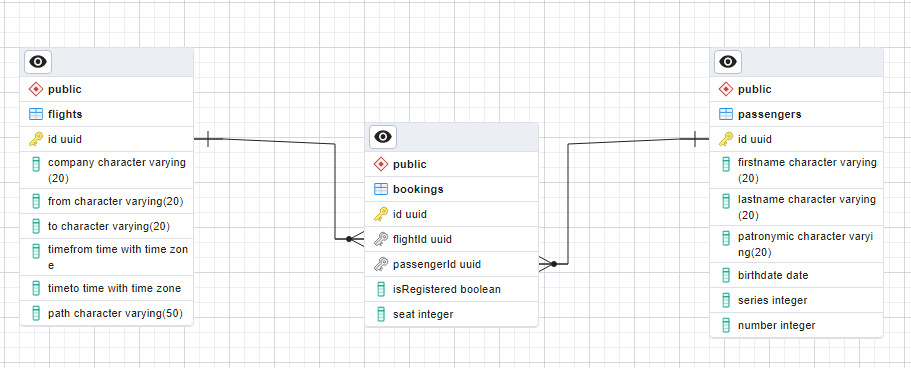


Рисунок 19 – физическая модель базы данных

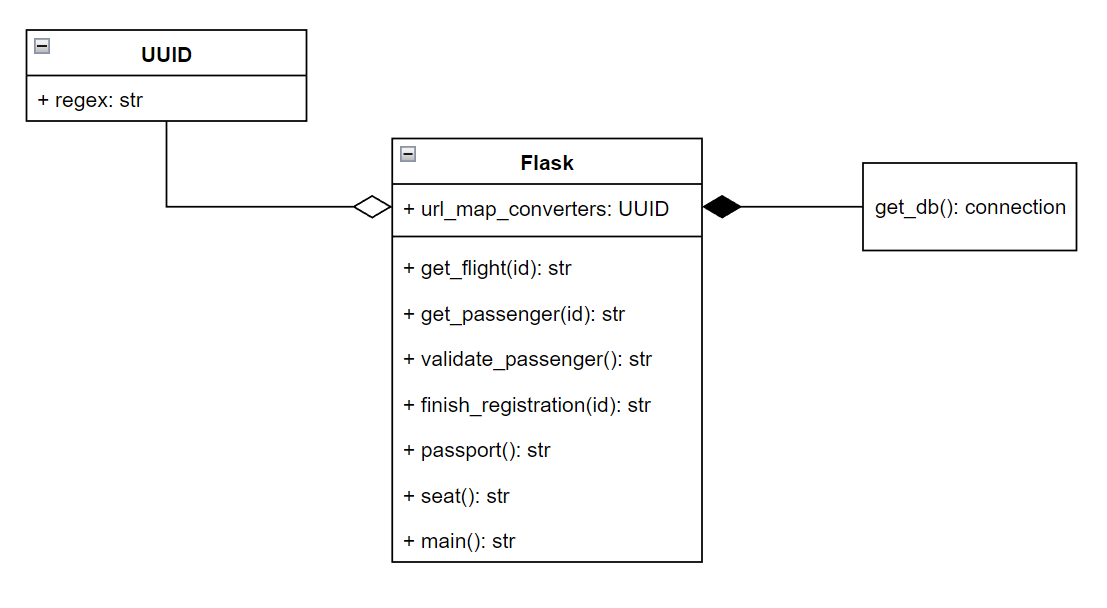


Рисунок 20 – диаграмма классов приложения

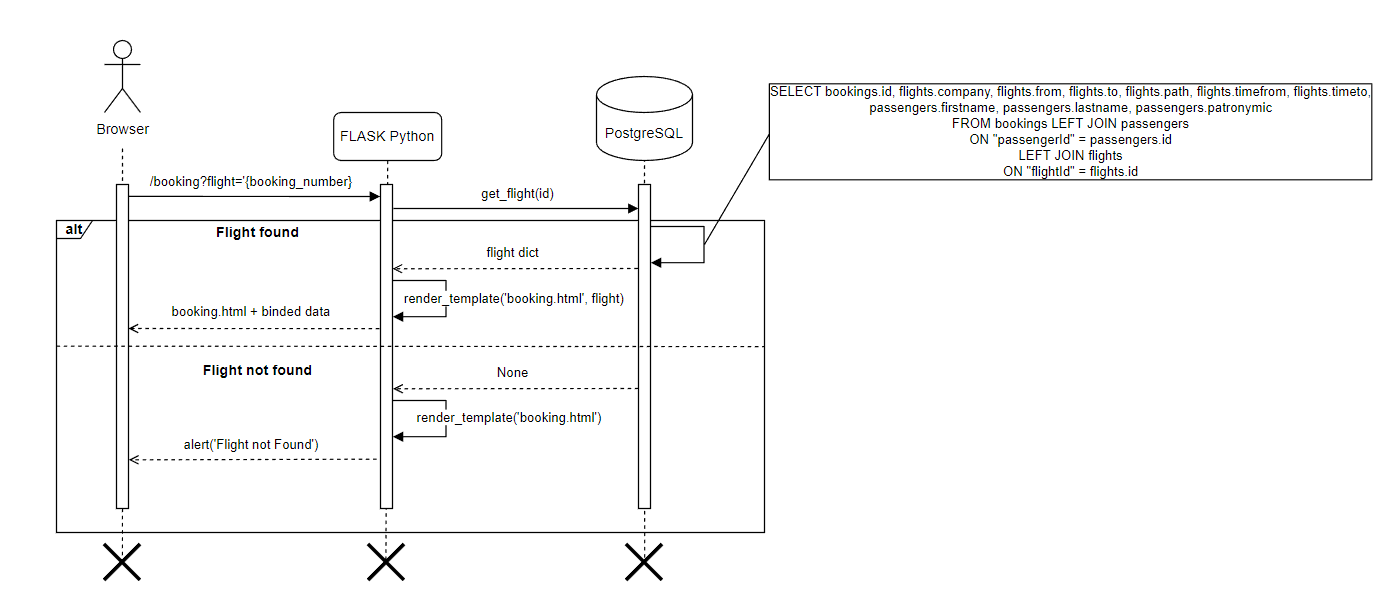


Рисунок 21 – спроектированный Use Case ввод номера бронирования

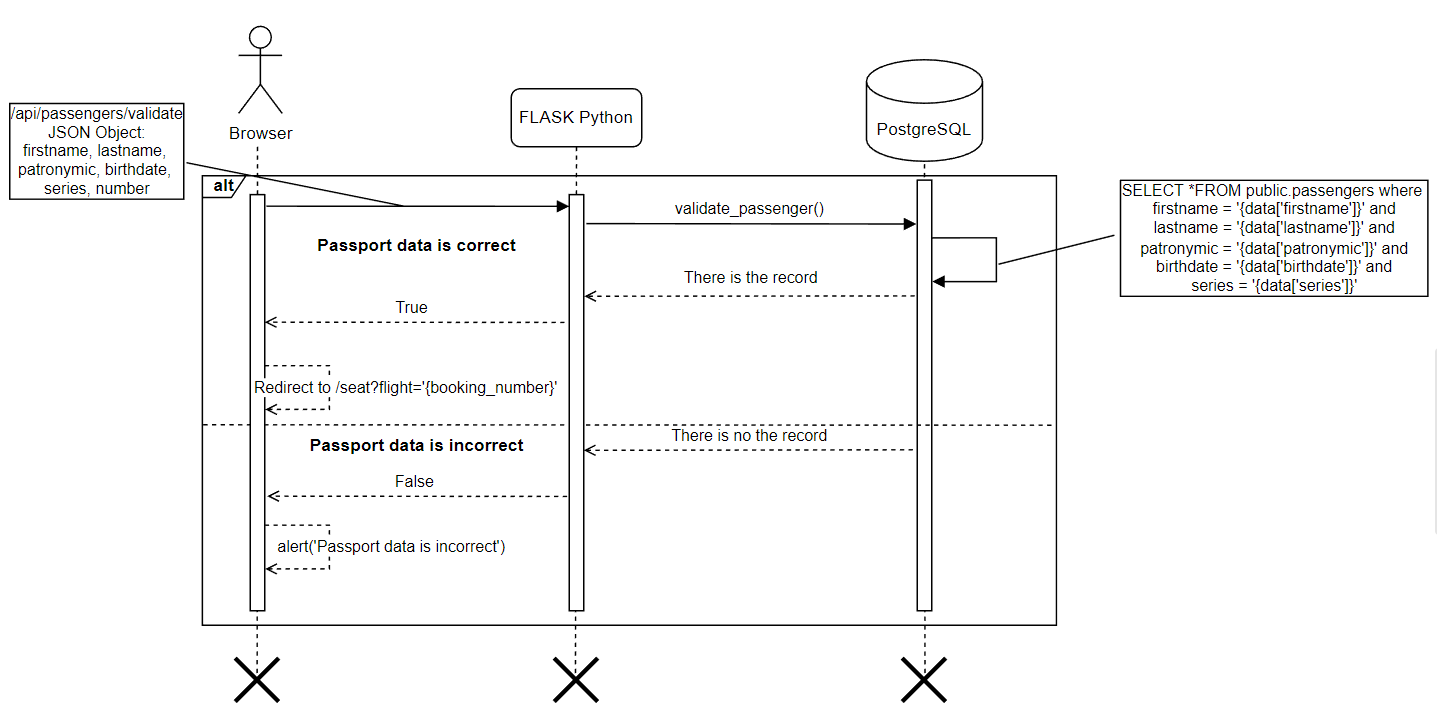


Рисунок 22 – спроектированный Use Case проверка паспортных данных

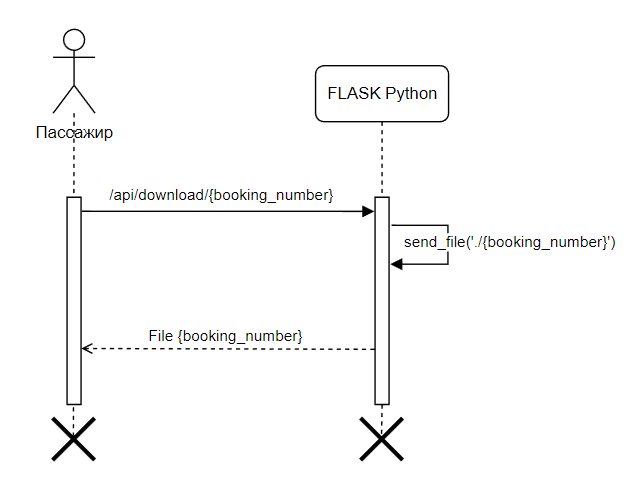


Рисунок 23 – спроектированный Use Case получение маршрутного листа

## 2 Описание алгоритма работы приложения

Приложение состоит 2 двух частей frontend и backend. Frontend это html файлы (интерфейс) с информацией, который отдается пользователю. Backend – это API функции-обработчики для манипуляции над данными из БД. Каждая функция обработчик однотипна и очень похожа по сравнению с другими. Каждая функция обработчик должна вернуть содержимое html файла либо иную строку, если функции обработчики имеют в аннотации, маршрут которых начинающейся с **/api**

Функции, начинающиеся с **/api** обрабатывают данные

Остальные функции-обработчики возвращают интерфейс приложения

Протокол между frontend и backend – HTTP

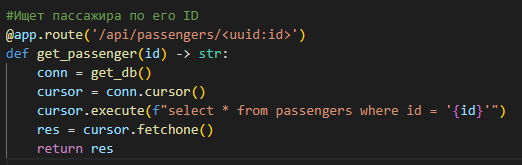


Рисунок 24 – функция обработчик манипулирующая с данными из БД

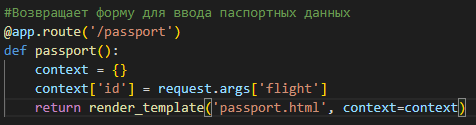


Рисунок X – функция обработчик, возвращает интерфейс (страницу)

## 3 Интерфейс

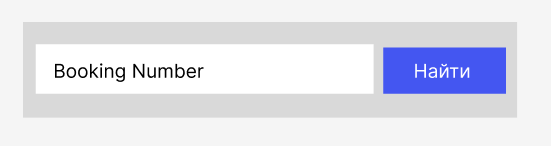


Рисунок 23 – Начальная страница (Ввод номера бронирования)

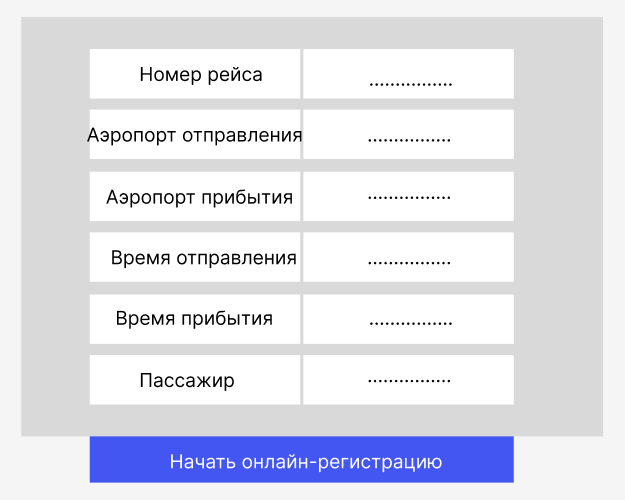


Рисунок 25 – макет страницы информации о рейсе

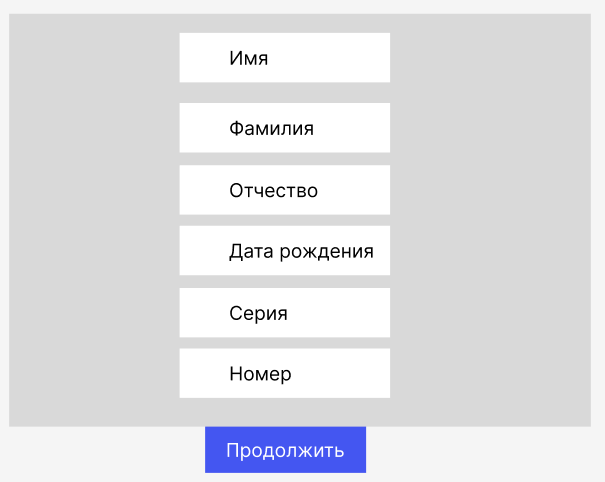


Рисунок 26 – макет страницы ввода паспортных данных



Рисунок 27 – макет страницы выбора места

## Тестирование

## Исходный код авто тестов

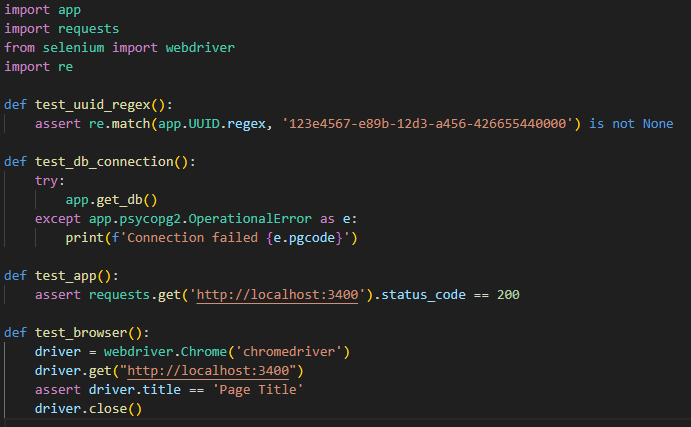
****

Рисунок 28 – Авто тесты ПО

Таблица 5 – Описание авто тестов

|  |  |  |
| --- | --- | --- |
| № | Тип авто теста | Описание |
| 1 | Unit тест | Проверяет корректность написанного регулярного выражения |
| 2 | Интеграционный тест | Проверяет доступ к базе данных |
| 3 | Интеграционный тест | Проверят доступ к приложению |
| 4 | End2End тест | Проверяет доступ к приложению, используя браузер Chrome. |

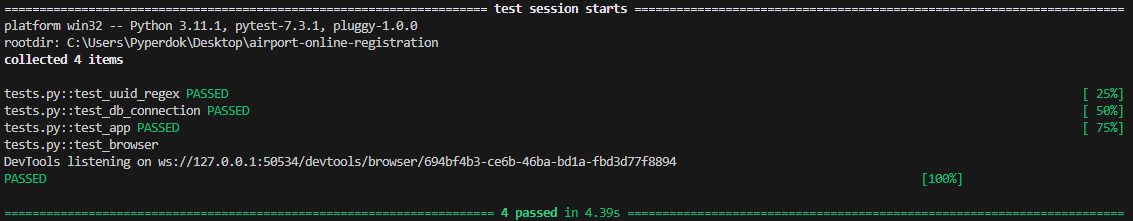


Рисунок 29 – Результат тестирования

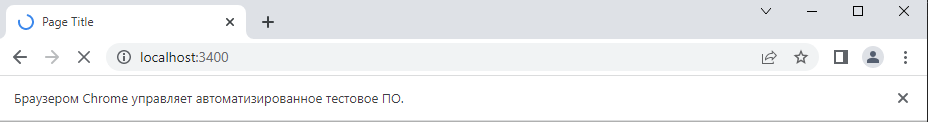
****

Рисунок 30 – Результат End2End теста используя Selenium

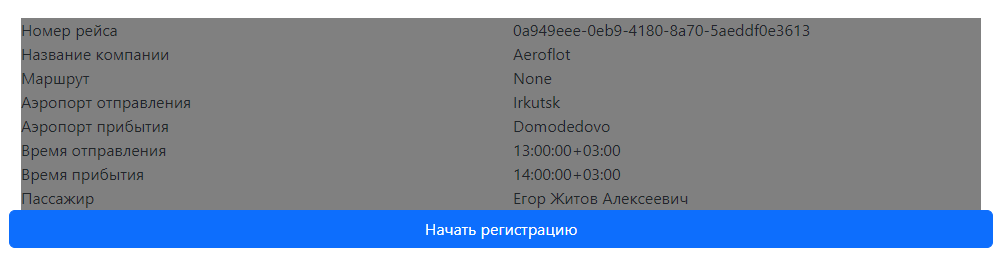
(Проверка доступности содержимого главной страницы)

## Документация пользователя

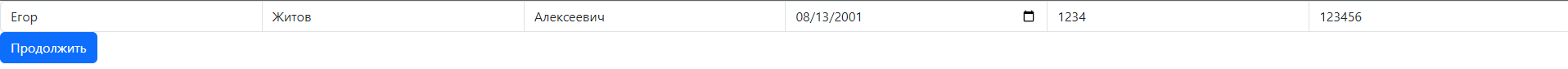
Чтобы начать процесс регистрации, вам необходимо ввести номер бронирования, который вы получили при покупке авиабилета. После того, как вы введете номер, нажмите кнопку "Найти".



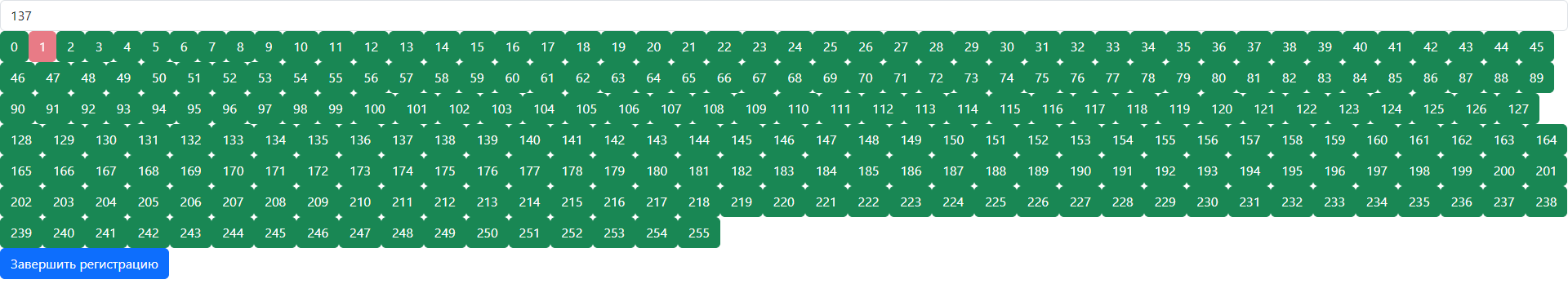
На следующей странице вы увидите информацию о рейсе и вашем билете. Если информация верна, нажмите кнопку "Начать регистрацию".

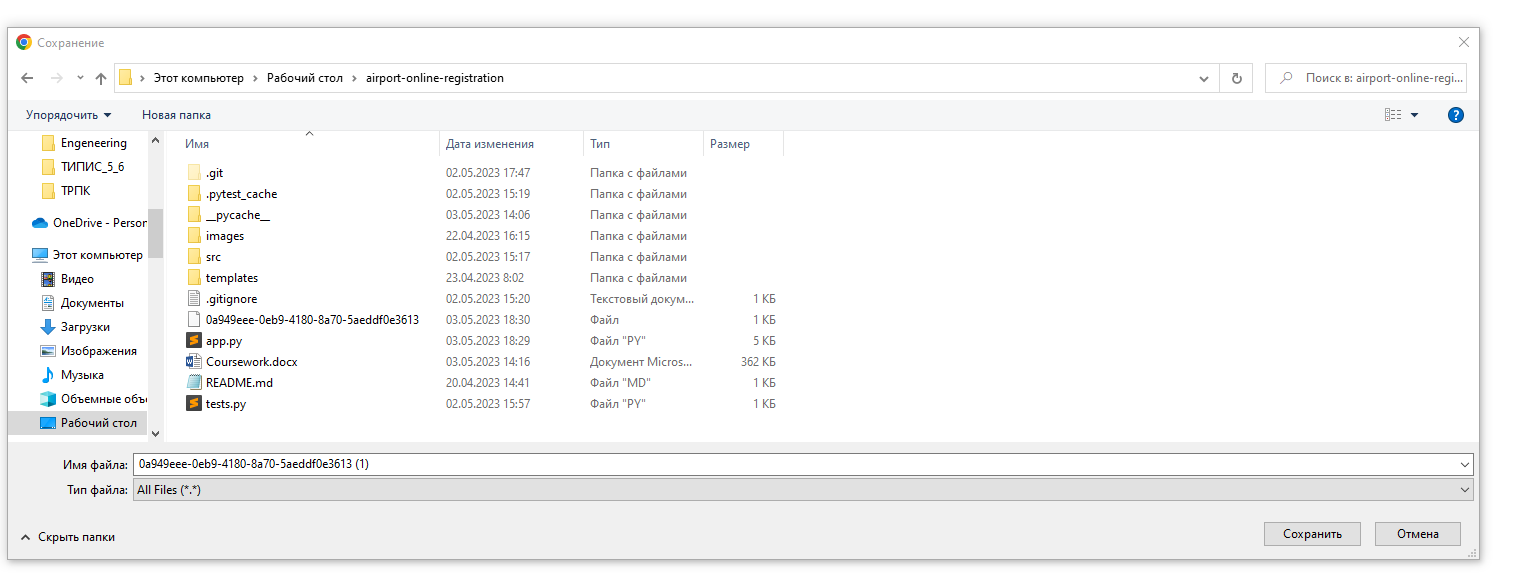


Далее вам будет предложено ввести ваши паспортные данные, такие как Имя, Фамилия, Отчество, Дата рождения, Серия и Номер. После того, как вы введете эту информацию, нажмите кнопку "Продолжить".



На следующей странице вы увидите места и сможете выбрать место, которое вам больше всего нравится. Если вам подходит место, которое автоматически было назначено системой, вы можете его оставить без изменений. Если вы хотите выбрать другое место, выберите любое другое свободное место. После того, как вы выберете место, нажмите кнопку "Завершить регистрацию".



На последней странице вы сможете скачать маршрутный лист, который вам необходимо будет предъявить на регистрации в аэропорту. 

## Заключение

В результате разработки информационной системы для онлайн-регистрации авиабилетов были достигнуты следующие результаты:

1. Улучшены знания языка программирования Python, в том числе его возможностей при создании веб-приложений;
2. Была изучена клиент-серверная архитектура, которая была использована при создании информационной системы;
3. Получены навыки автоматизированного тестирования с использованием библиотеки Selenium, что позволило увеличить скорость разработки и повысить качество решения;
4. Были изучены принципы жизненного цикла ПО, которые были использованы при разработке системы, что обеспечило ее эффективность и надежность;
5. В результате разработки информационной системы была достигнута автоматизация процесса онлайн-регистрации авиабилетов, что позволило повысить качество обслуживания пассажиров и ускорить процесс регистрации.

Таким образом, разработка информационной системы для онлайн-регистрации авиабилетов позволила значительно расширить знания и навыки в области программирования и разработки ПО, а также сделала важный вклад в улучшение качества обслуживания пассажиров.

GitHub репозиторий проекта:

(<https://github.com/Pyperdok/airport-online-registration>)

## Список использованных источников

1. GPT 3.5 // Chat GPT URL: <https://chat.openai.com/> (дата обращения: 08.05.2023).
2. Pytest Docs // Pytest URL: [https://docs.pytest.org/en/7.3.x](https://docs.pytest.org/en/7.3.x%20) (дата обращения: 08.05.2023).
3. Selenium Web Driver // Selenuim URL: <https://www.selenium.dev/documentation/webdriver/> (дата обращения: 08.05.2023).
4. Flask Docs // Flask URL: <https://flask.palletsprojects.com> (дата обращения: 08.05.2023).
5. Questions // StackOverflow URL: <https://stackoverflow.com> (дата обращения: 08.05.2023).