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 |

Допускаю к защите

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

|  |
| --- |
| Проектирование и разработка прикладного ПО |
| «Онлайн-регистрация авиабилетов» |
| наименование темы |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по дисциплине

|  |
| --- |
| Информатика |

|  |  |  |
| --- | --- | --- |
| 1.010.00.00 – ПЗ | | |
| обозначение документа | | |
| Выполнил студент |  | ИСТб-20-3 | |  |  | |  | Ю.В. Жалсанов |
|  |  | 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** | Жалсанову Ю. В. | | | | | | | | | | | | | | | | |
|  | (фамилия, инициалы] | | | | | | | | | | | | | | | | |
| **Тема работы** | | Проектирование и разработка прикладного ПО. | | | | | | | | | | | | | | | |
| **Исходные данные** | | 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 | | | « | 16 | » | февраля | | | | | 2023 г. | | | | |
|  | | | | | | | | | | | | | | | | | |
| Задание получил | | | | | | | |  | | | | |  | | Ю.В. Жалсанов | | |
|  | | | | | | | | подпись | | | | |  | | И.О. Фамилия | | |
|  | | | | | | | | |  |  | |  | |  | | |  | |
| Дата представления работы руководителю | | | | | | | | | « | 21 | | » | | мая | | | 2023г. |
|  | | | | | | | | | | | | | | | | | | |
| Руководитель курсовой работы | | | | | | | |  | | | | |  | | З.А. Бахвалова | | |
|  | | | | | | | | подпись | | | | |  | | И.О. Фамилия | | |

**Содержание**

[1 Формирование требований 5](#_Toc134953269)

[1.1 Описание предметной области 5](#_Toc134953270)

[1.2 Описание проблемы 5](#_Toc134953271)

[1.3 Цель 5](#_Toc134953272)

[1.4 Словари 6](#_Toc134953273)

[1.5 Постановка задачи 9](#_Toc134953274)

[1.6 Функциональные требования ПО 10](#_Toc134953275)

[2 Анализ функциональных требований 11](#_Toc134953276)

[2.1 Модель предметной области (IDEF0) 11](#_Toc134953277)

[2.1 Диаграмма вариантов использования Use Case 13](#_Toc134953278)

[2.2 DFD - диаграмма потоков данных 14](#_Toc134953279)

[2.3 Концептуальная модель хранилища данных 15](#_Toc134953280)

[2.4 Описание вариантов использования 16](#_Toc134953281)

[2.5 Диаграммы состояний 20](#_Toc134953282)

[2.6 Описание интерфейса 21](#_Toc134953283)

[3 Проектирование 23](#_Toc134953284)

[3.1 Инструменты 23](#_Toc134953285)

[3.2 Реализация 27](#_Toc134953286)

[3.3 Описание алгоритма 28](#_Toc134953287)

[3.4 Интерфейс 29](#_Toc134953288)

[4 Тестирование 31](#_Toc134953289)

[4.1 Исходный код авто тестов 31](#_Toc134953290)

[5 Документация пользователя 32](#_Toc134953291)

[Заключение 33](#_Toc134953292)

[Список использованных источников 34](#_Toc134953293)

**Введение**

Современный мир становится все более информационно-ориентированным, что приводит к необходимости использования информационных технологий во всех сферах деятельности. Один из наиболее важных аспектов современной транспортной инфраструктуры - это возможность быстрой и удобной регистрации пассажиров на авиарейсы. Онлайн-регистрация авиабилетов - это одно из самых популярных и востребованных решений для пассажиров, которые хотят сэкономить время и избежать очередей на регистрации в аэропорту.

Разработка системы онлайн-регистрации авиабилетов имеет большое значение в современном мире. Такие системы позволяют пассажирам производить регистрацию в любом месте и в любое время, не выходя из дома или офиса. Это не только экономит время, но и существенно повышает удобство и комфорт пассажиров.

## 1 Формирование требований

## 1.1 Описание предметной области

Сеть магазинов занимается продажей офисной мебели. Учет мебели на всех предприятиях ведется с помощью электронной системы учета. Офисная мебель характеризуется атрибутами: модель, материал, изготовитель, габариты, цвет, цена, количество в наличии. Каждый предмет мебели имеет свой артикул, который является идентификатором товара. Офисная мебель подразделяется на данные категории: шкаф, полка, стул, кресло. Габариты подразумевают следующие свойства мебели: высота, широта, глубина.

## 1.2 Описание проблемы

У магазина мебели есть проблема, когда клиент хочет купить мебель, а на складе магазина нет данной мебели или её нехватка, но в других отделениях магазина эта мебель есть, продавцу приходится звонить в магазин и оповещать другого продавца, чтобы он забронировал требуемую мебель, информацию о брони запоминают, либо записывают на листочек и из за этого есть шанс забыть эту бронь или потерять. Из-за отсутствия возможности бронирования в системе, происходит потеря клиентов.

## 1.3 Цель

Уменьшение потери клиентов магазина за счет добавления модуля бронирования товаров.

## 1.4 Словари

Таблица 1 - Словарь предметной области

|  |  |  |
| --- | --- | --- |
| **№** | **Сущность** | **Описание** |
| 1 | Предмет мебели/товар | Предмет офисной обстановки, который продает компания-заказчик. Имеет свойства: название, модель, артикул, изготовитель, цена, цвет, габариты, материал, кол-во, место расположения. |
| 2 | Запись брони (бронь) | Запись о мебели, уведомляющая о том, что данная мебель забронирована покупателем и не принадлежит продаже другим клиентам. |
| 3 | Модель | Полное имя товара, принятое производителем и его характеристики. |
| 4 | Категория товара | Группа товаров, которые отвечают за один и тот же функционал. Категории могут быть: стол, стул, диван, полка, шкаф, кресло. |
| 5 | Материал | Вещество, из которого изготовлена мебель. |
| 6 | Название | Имя товара, которое является уникальным. Например: диван «Атлант», где «Атлант» является названием. |
| 7 | Изготовитель | Организация, которая является производителем предмета мебели. Изготовитель имеет атрибуты: наименование, страна, адрес. |
| 8 | Габариты | Размер мебели: высота, ширина, длинна. Указывается в мм. |
| 9 | Артикул | Идентификатор товара в системе магазина. Для одного товара разных цветов, материалов будут разные артикулы. |
| 10 | Список товаров | Перечень товаров, который виден пользователю при просмотре категории или является результатом поиска. |
| 11 | Магазин | Предприятие оптово-розничной торговли, размещённое в стационарном здании по закрепленному адресу, оборудованном для продаж товаров. |
| 12 | Список броней | Перечень забронированного товара |

Таблица 2 - Словарь по Абботу

|  |  |  |
| --- | --- | --- |
| **№** | **Сущность** | **Глагол** |
| 1 | Предмет мебели/товар | Выбирать |
| 2 | Запись брони (бронь) | Создавать, удалять, редактировать, выбирать, |
| 3 | Модель | Выбирать |
| 4 | Категория товара | Выбирать |
| 5 | Материал | Выбирать |
| 6 | Название | Выбирать |
| 7 | Изготовитель | Выбирать |
| 8 | Габариты | Выбирать |
| 9 | Артикул | Выбирать |
| 10 | Список товаров | формировать,  сортировать,  отображать |
| 11 | Список броней | формировать,  сортировать,  отображать |

Таблица 3 - Объектно-ориентированный словарь

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Сущность** | **Данные** | **Методы** |
| 1 | Предмет мебели/товар | Название, модель, картинка, артикул, цена, цвет, кол-во, место расположения | выбирать |
| 2 | Бронь | Номер записи, название, модель, цвет, артикул, кол-во которое нужно забронировать, место брони, дата окончания брони. | Создавать, удалять, редактировать, выбирать |
| 3 | Модель | Категория товара, название, материал, производитель, размер | выбирать |
| 4 | Категория товара | Может быть:   1. Шкаф, 2. полка, 3. стул, 4. диван, 5. стол, 6. кресло | Выбирать |
| 5 | Материал | Название вещества, из которого сделана мебель | Выбирать |
| 6 | Название | Имя товара | Выбирать |
| 7 | Изготовитель | Наименование, страна, адрес производства. | Выбирать |
| 8 | Габариты | Высота, ширина, длинна | Выбирать |
| 9 | Артикул | Набор цифр | Выбирать |
| 10 | Список товаров | Модель, цвет, цена, кол-во на складе | Формировать,  Сортировать, отображать |
| 11 | Список броней | Номер брони, название мебели, модель, цвет, кол-во, адрес бронирования, срок бронирования. | формировать,  сортировать, отображать |

## 1.5 Постановка задачи

Необходимо реализовать модуль бронирования, для магазина мебели, с помощью которой можно будет бронировать товар в других отделениях магазина.

Данной системой будет пользоваться продавец, далее расписаны возможности взаимодействия пользователя с системой:

Продавец:

* Имеет возможность просмотреть мебель, для предоставления клиенту информации о мебели;
* имеет возможность бронировать мебель в других отделениях, для дальнейшего самовывоза клиентом;
* имеет возможность удалять или редактировать запись бронирования;
* имеет возможность просмотреть записи бронирования.

При создании записи брони продавцу требуется ввести такие данные как категория, артикул, кол-во которое нужно забронировать и адрес магазина в котором бронируют, а также дату бронирования. После ввода этих данных появляется запись в таблице броней.

Если клиент решил не покупать мебель или купил в другом магазина, то есть возможность удалить запись бронирования. Запись бронирования удаляется при вводе id брони.

У продавца так же есть возможность редактировать бронь, если клиент решил изменить кол-во мебели, либо ему требуется увеличить срок бронирования, либо изменить адрес брони.

## 1.6 Функциональные требования ПО

**Формирование брони мебели**

Система должна предоставлять возможность бронирования мебели в отделениях магазина и на складе. Для бронирования, требуется указать категорию, артикул, кол-во, расположение и дату бронирования. В результате в список бронирования добавится новая запись.

**Формирование списка записей бронирования**

Система должна давать возможность пользователю вывести список записей брони, список можно сортировать по разным критериям.

**Формирование списка мебели**

Система должна давать возможность сформировать список мебели. Мебель разделяется на 4 категории(стул, кресло, шкаф, полка), поэтому список формируется при выборе одной из категорий. Каждый элемент в списке имеет следующие поля: модель, артикул, изготовитель, материал, цвет, габариты, цена, расположение и кол-во.

**Удаление брони**

Система должна давать возможность удалять запись бронирования. Бронь удаляется при вводе id записи.

**Редактирование брони**

Система должна давать возможность редактировать запись бронирования. При выборе id записи появляются поля этой записи, которые можно изменять.

# 2 Анализ функциональных требований

## 2.1 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.

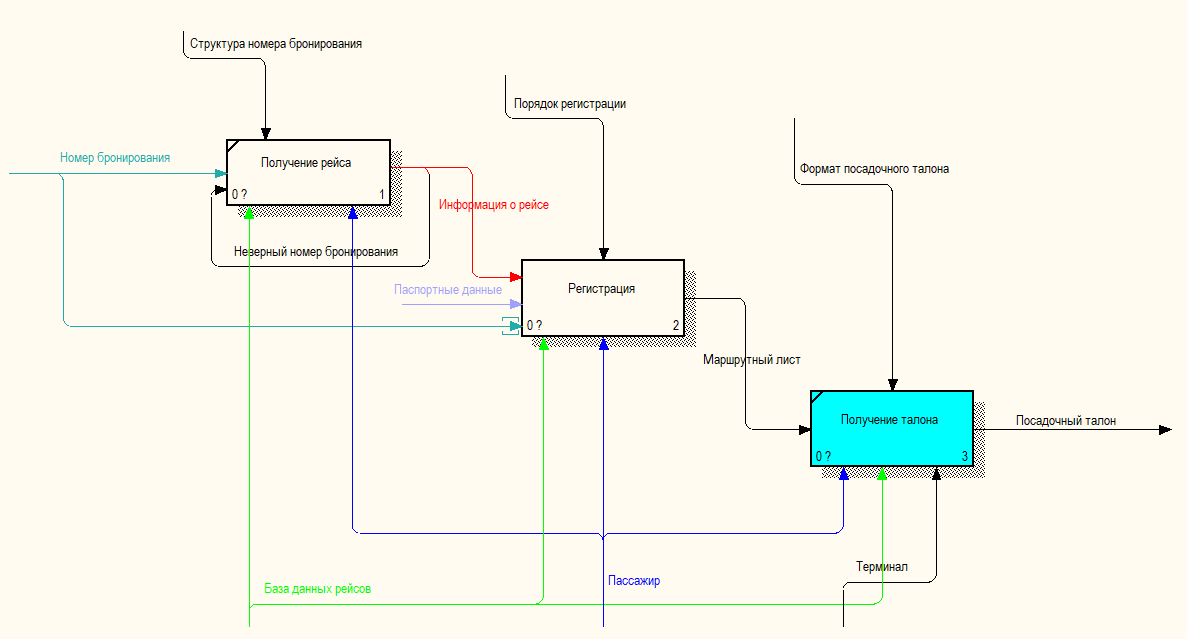


Рисунок 2 – Модель IDEF0. Уровень А1. (Онлайн-регистрация)

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.

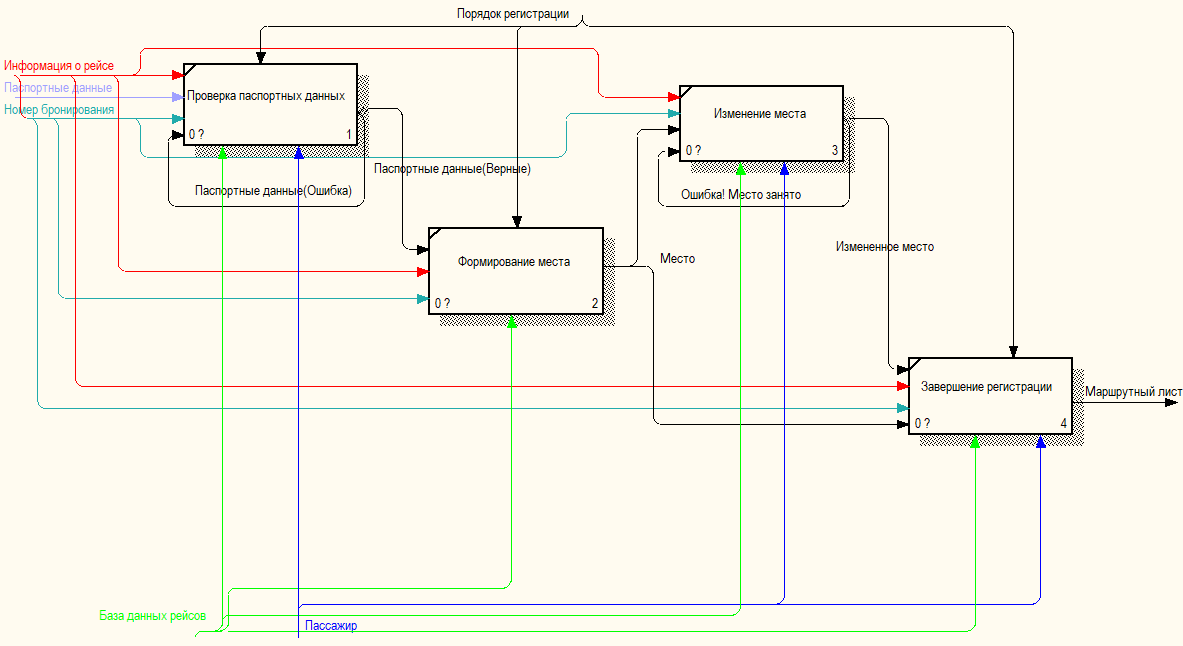


Рисунок 3 – Модель IDEF0. Уровень А2. (Регистрация)

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.

## 2.1 Диаграмма вариантов использования Use Case

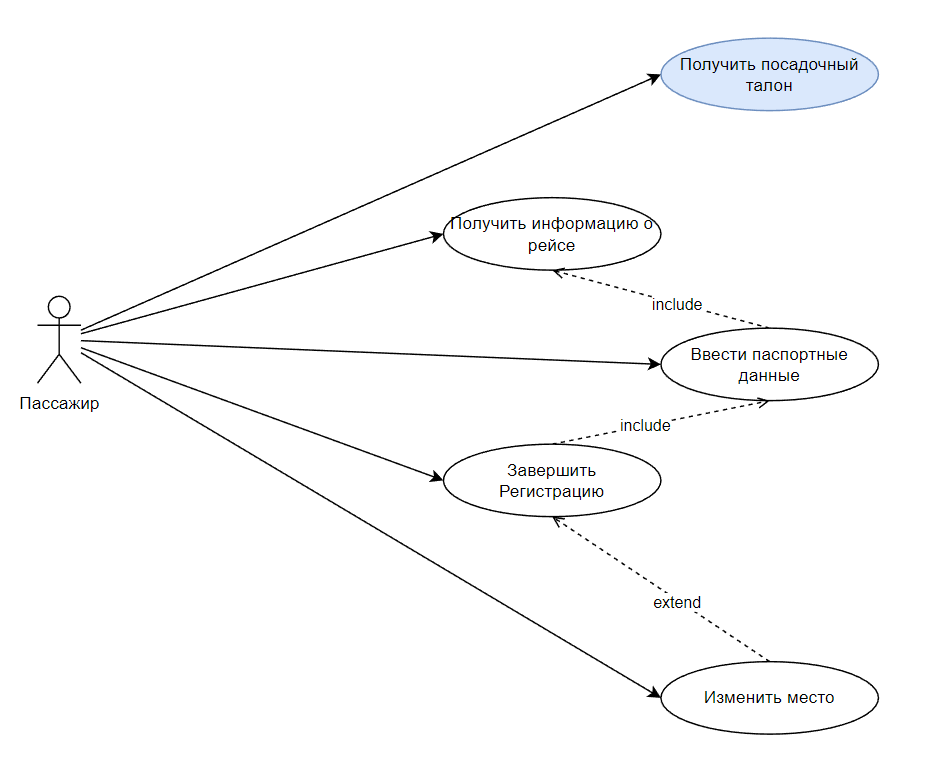


Рисунок 4 – Диаграмма Use Case.

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.

# 2.2 DFD - диаграмма потоков данных

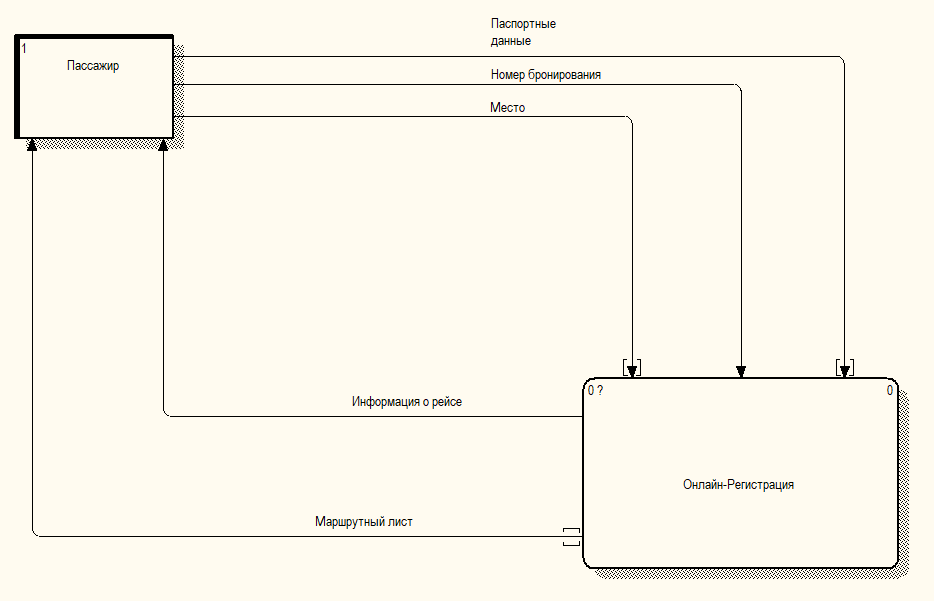


Рисунок 5 – Диаграмма потоков данных. A0

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

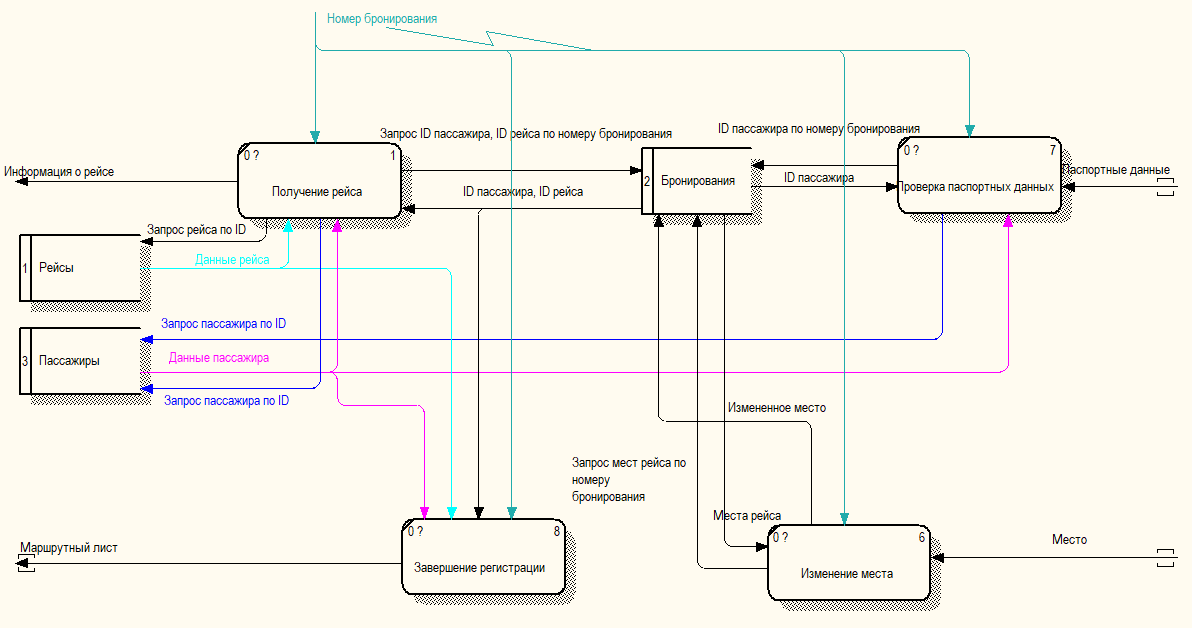


Рисунок 6 – Диаграмма поток данных. Уровень А1.

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

## 2.3 Концептуальная модель хранилища данных

Основываясь на DFD диаграмму была разработана Концептуальная модель хранилища данных

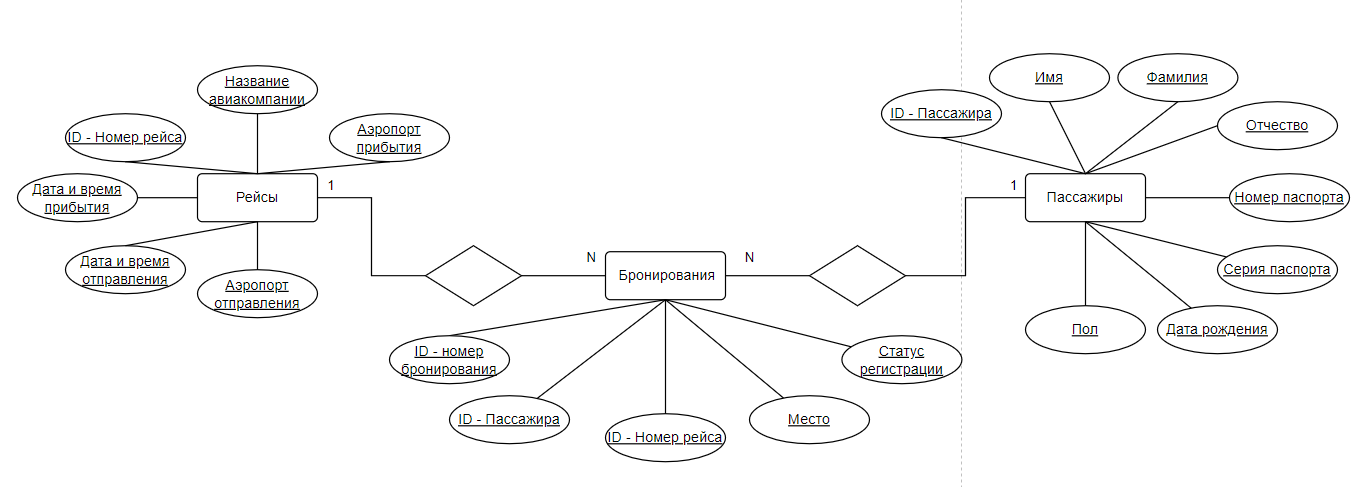


Рисунок 7 – ER-модель.

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.***

## 2.4 Описание вариантов использования

**2.4.1 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.

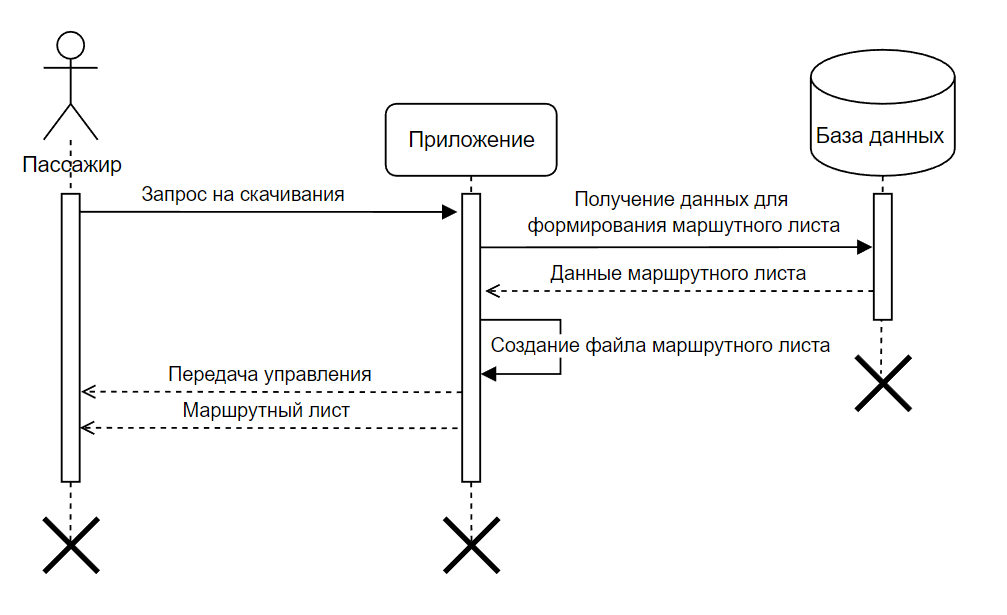


Рисунок 8 – Диаграмма последовательностей варианта использований

*«Получить маршрутный лист»*.

**2.4.2 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.

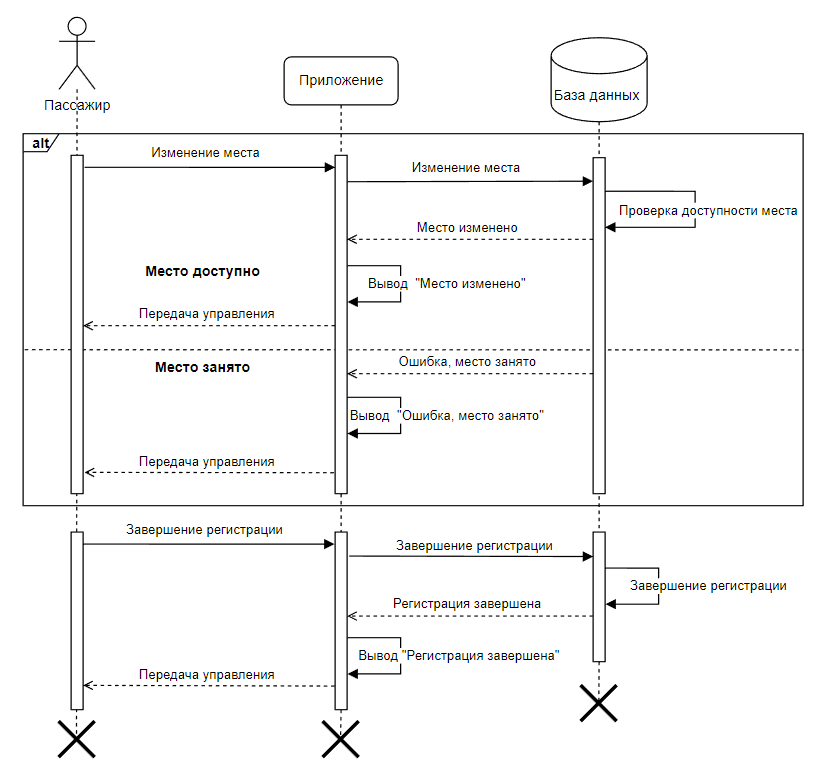


Рисунок 9 – Диаграмма последовательностей варианта использований *«Завершить регистрацию»*.

**2.4.3 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.

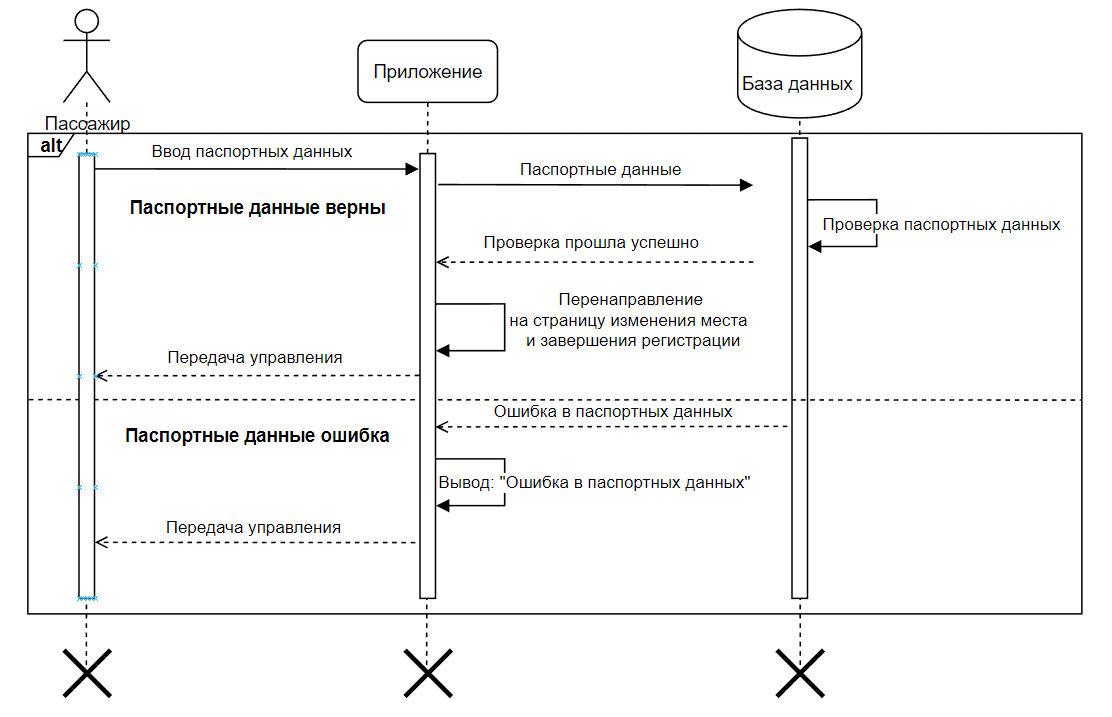


Рисунок 10 – Диаграмма последовательностей варианта использований

*«Ввести паспортные данные»*.

**2.4.4 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.

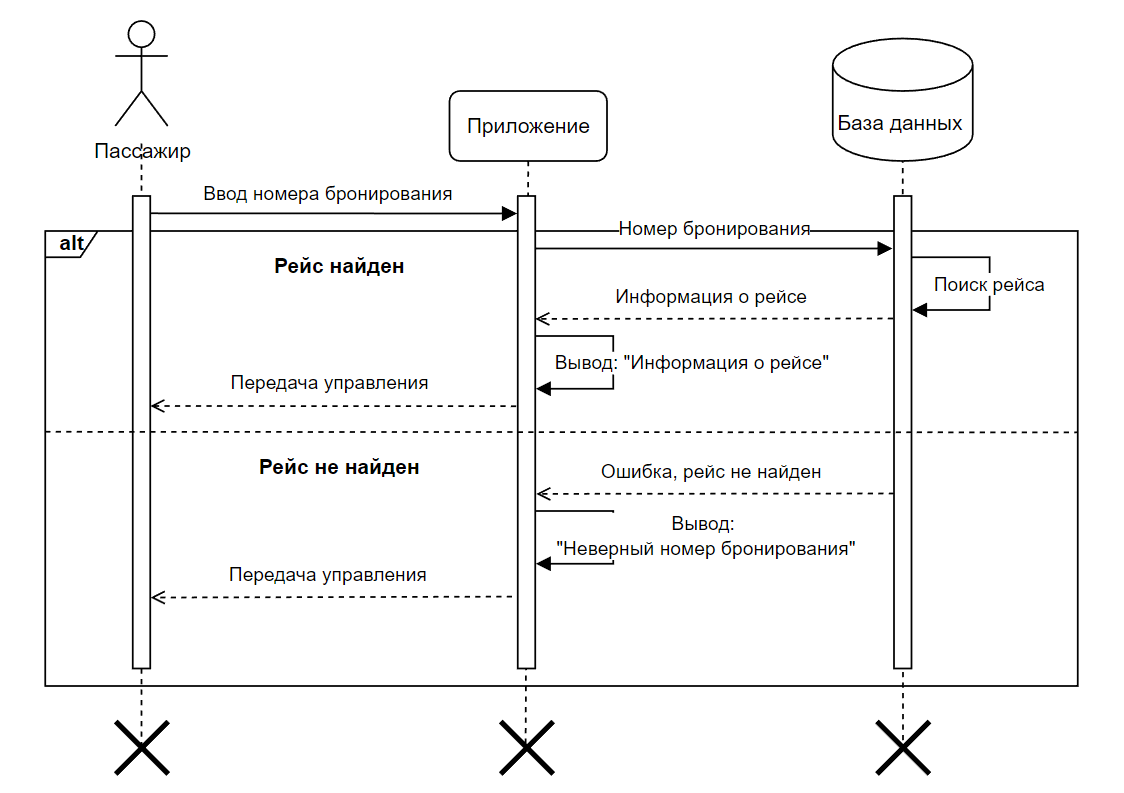


Рисунок 11 – Диаграмма последовательностей варианта использований

*«Получить информацию о рейсе»*.

## 2.5 Диаграммы состояний

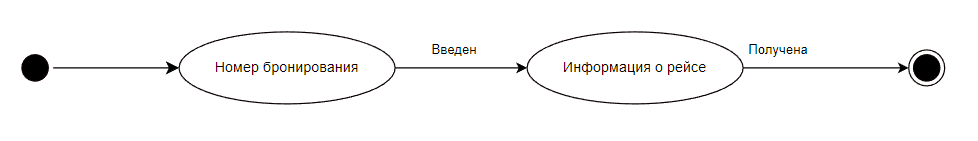


Рисунок 12 – Диаграмма состояний. Получения информации о рейсе

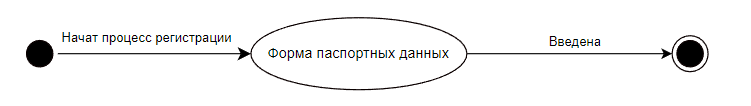


Рисунок 13 – Диаграмма состояний. Ввода паспортных данных

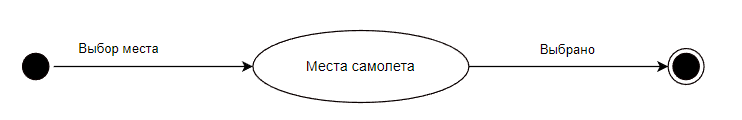


Рисунок 14 – Диаграмма состояний Изменения места

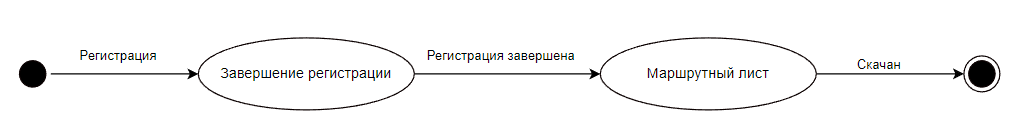


Рисунок 15 – Диаграмма состояний Завершение регистрации.

## 2.6 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.

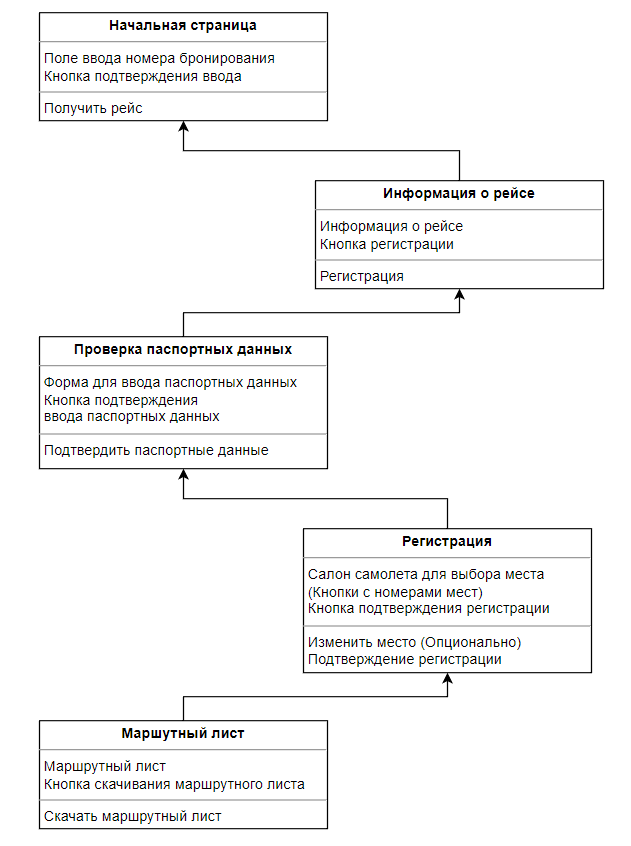


Рисунок 16 – Диаграмма интерфейсных классов

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.

## 3 Проектирование

## 3.1 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.

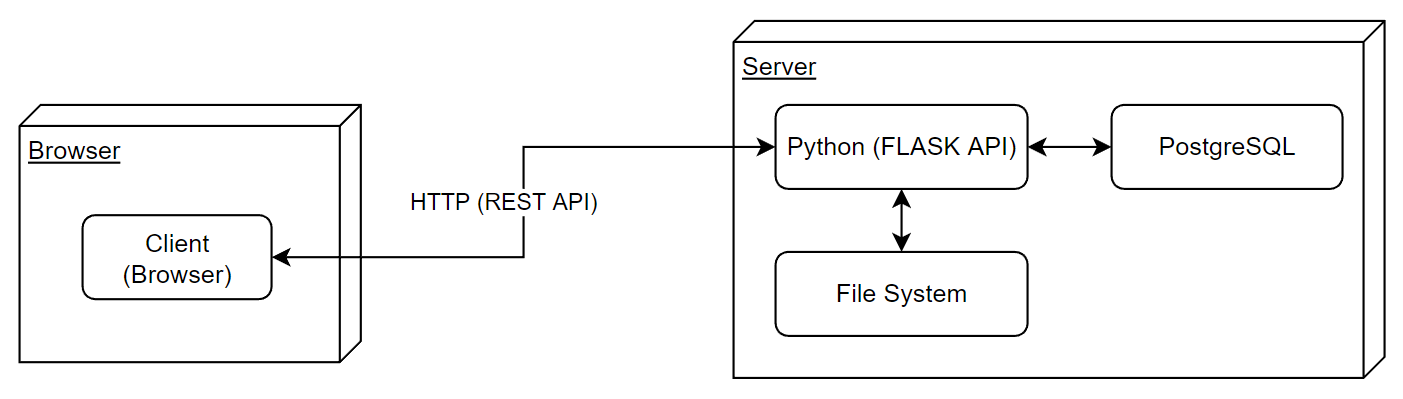


Рисунок 17 – Диаграмма развертывания приложения

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.

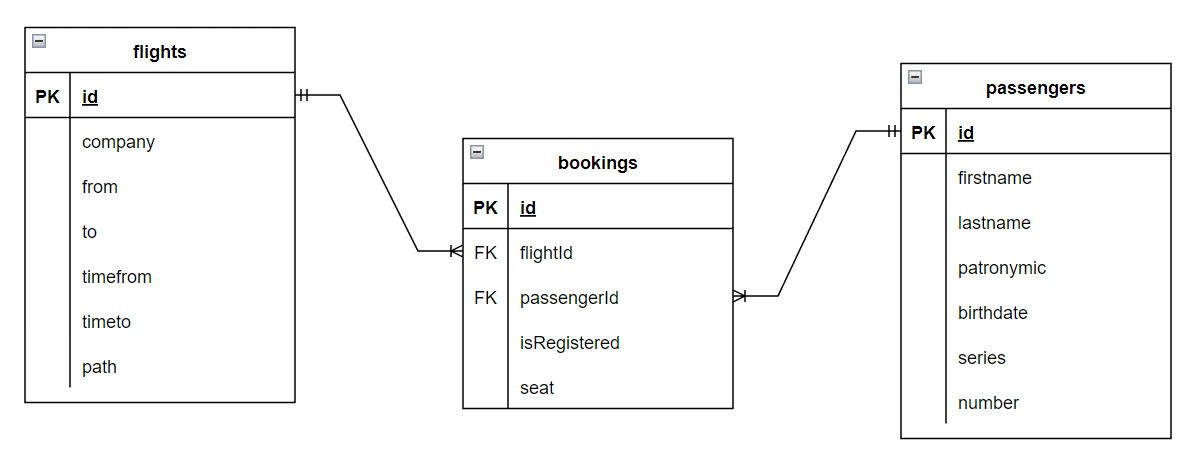


Рисунок 18 – Логическая модель базы данных

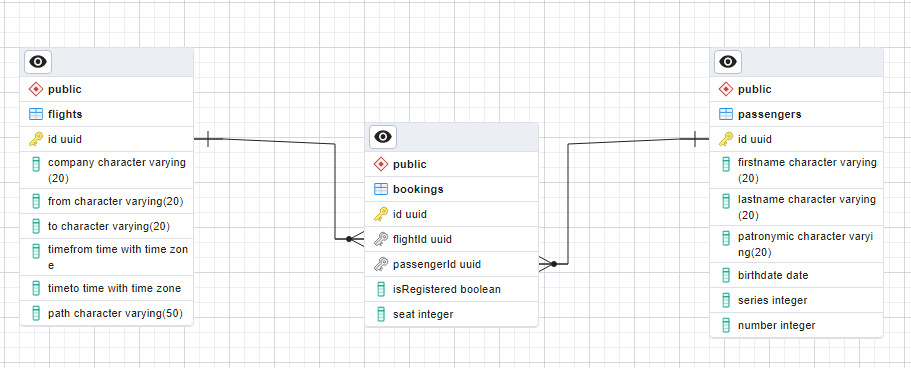


Рисунок 19 – Физическая модель базы данных

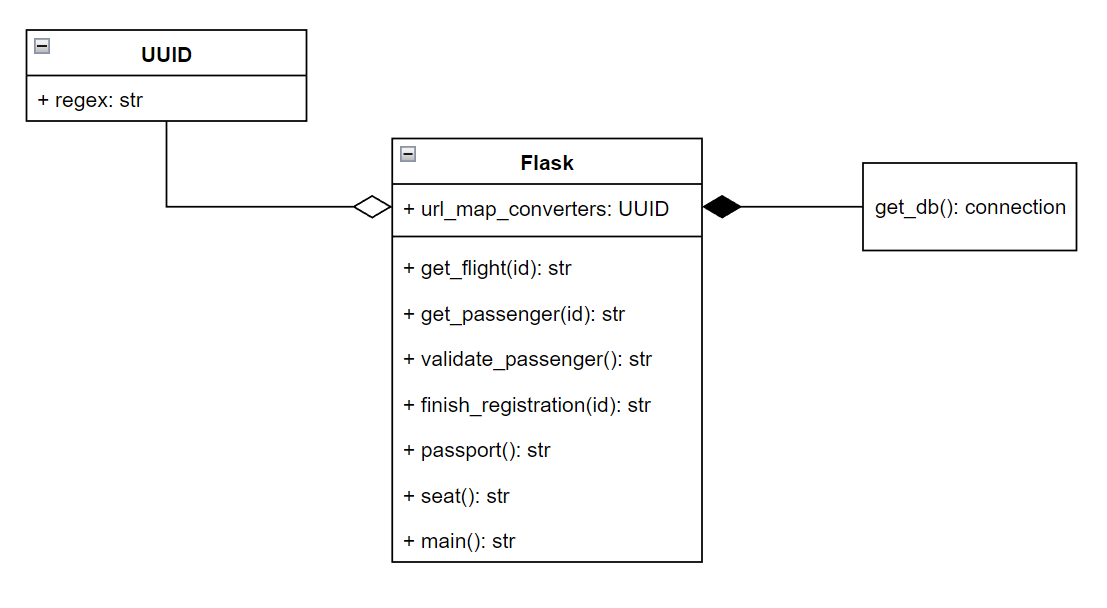


Рисунок 20 – Диаграмма классов приложения

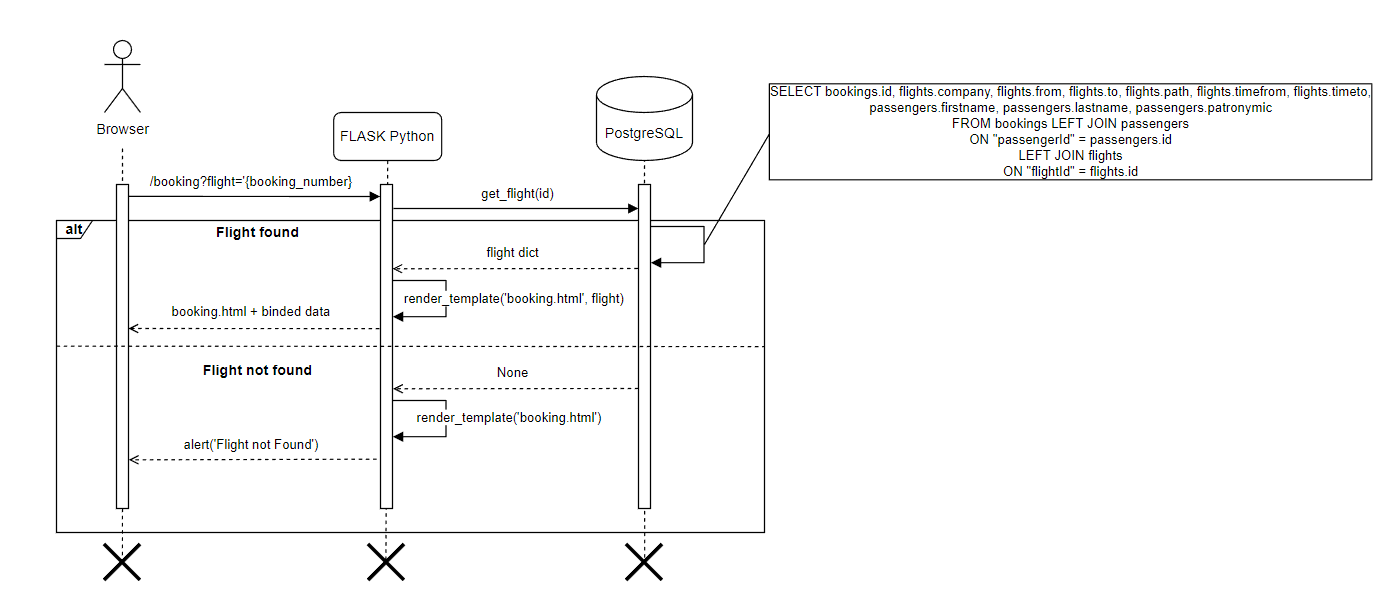


Рисунок 21 – Проектирование Use Case ввод номера бронирования

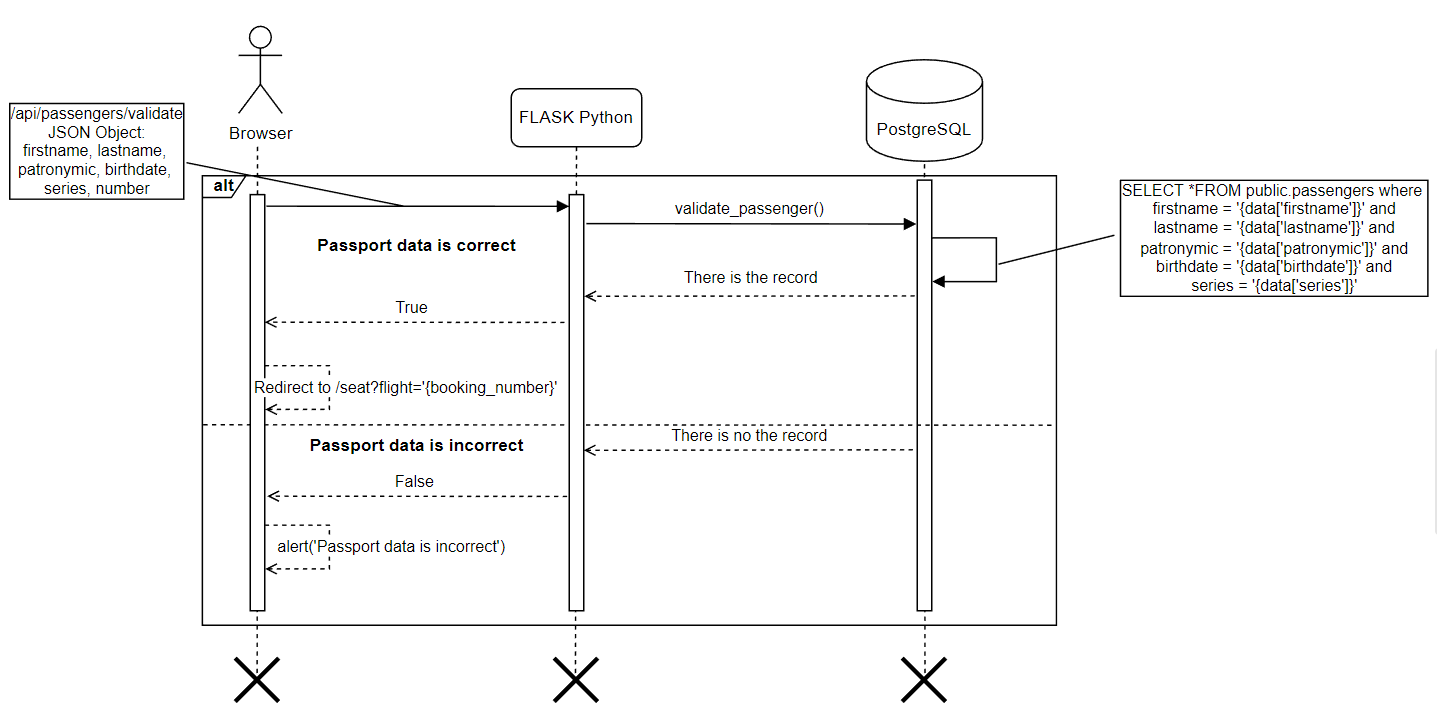


Рисунок 22 – Проектирование Use Case проверка паспортных данных

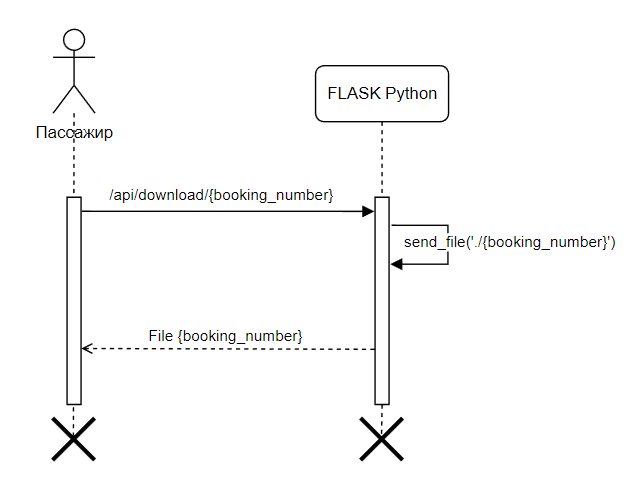


Рисунок 23 – Проектирование Use Case получение маршрутного листа

# 3.2 Реализация

Таблица 5 – Спецификация функций-обработчиков

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Маршрут | Название | Параметры | Описание |
| 1 | /api/bookings/<uuid:id> | get\_flight | uuid | Ищет по номеру бронирования рейс и пассажира |
| 2 | /api/passengers/<uuid:id> | get\_passenger | uuid | Ищет пассажира по его ID |
| 3 | /api/bookings/<uuid:id> | finish\_registration | uuid | Завершает регистрацию пассажира, изменяя его статус регистрации |
| 4 | /download/<uuid:id> | download | uuid | Отдает файл маршуртного листа |
| 5 | /api/passengers/validate | validate\_passenger | - | Проверяет паспортные данные введенные пассажиром |
| 6 | /booking | booking | - | Возвращает интерфейс для начала регистрации |
| 7 | /passport | passport | - | Возвращает форму для ввода паспортных данных |
| 8 | /seat | seat | - | Возвращает форму для выбора места |
| 9 | /route | route | - | Возвращает интерфейс для скачивания маршрутного листа |
| 10 | / | main | - | Возвращает форму для ввода номера бронирования |

## 3.3 Описание алгоритма

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

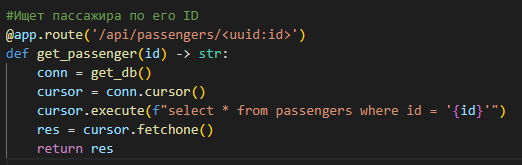


Рисунок 24 – Функция обработчик манипулирующая с данными из БД

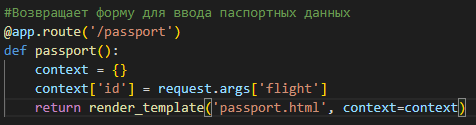


Рисунок 25 – Функция обработчик, возвращает интерфейс (страницу)

## 3.4 Интерфейс

Макет интерфейса были созданы в программе Figma

Ссылка на макет интерфейса: (<https://www.figma.com/file/RoJDDCTYDhHtewMhLhpKgM>)

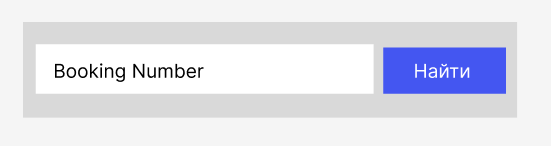


Рисунок 26 – Начальная страница (Ввод номера бронирования)

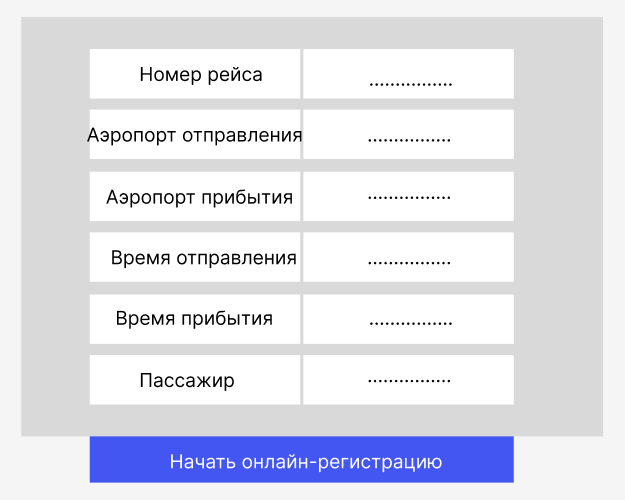


Рисунок 27 – Макет страницы информации о рейсе

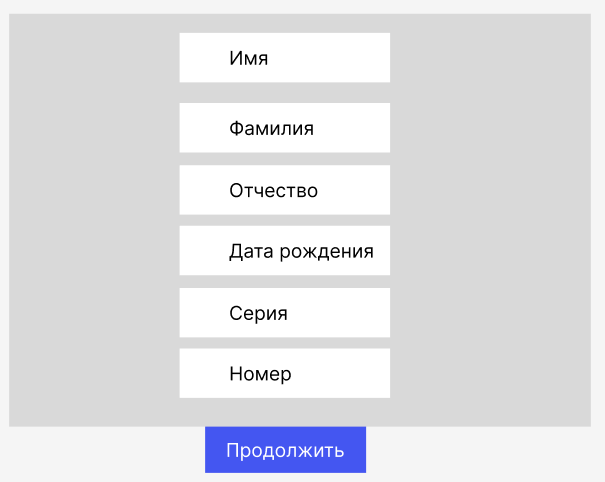


Рисунок 28 – Макет страницы ввода паспортных данных



Рисунок 29 – Макет страницы выбора места

## 4 Тестирование

## 4.1 Исходный код авто тестов

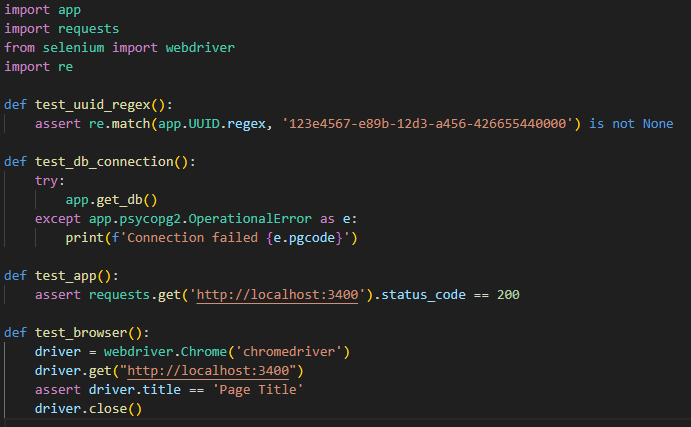
****

Рисунок 30 – Авто тесты ПО

Таблица 6 – 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. |

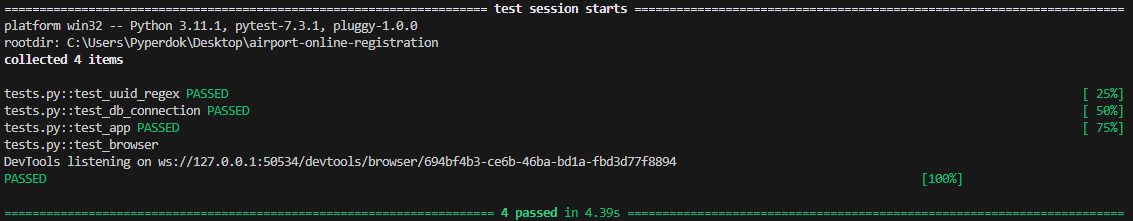


Рисунок 31 – Результат тестирования

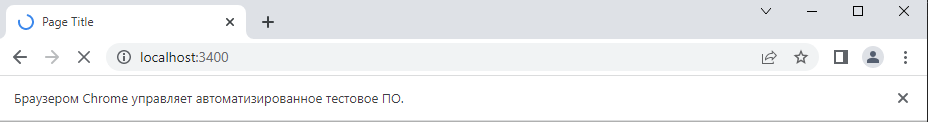
****

Рисунок 32 – Результат End2End теста используя Selenium

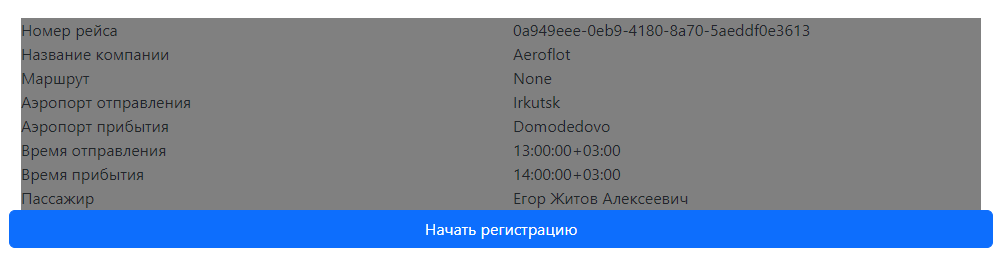
(Проверка доступности содержимого главной страницы)

## 5 Документация пользователя

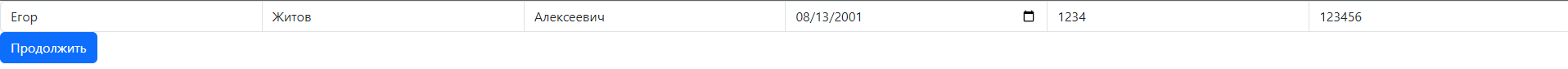
Чтобы начать процесс регистрации, вам необходимо ввести номер бронирования, который вы получили при покупке авиабилета. После того, как вы введете номер, нажмите кнопку "Найти".



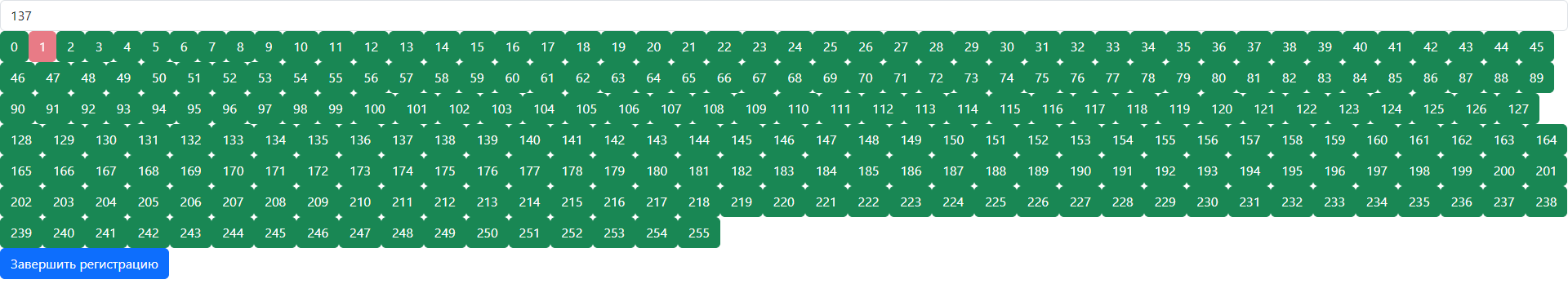
На следующей странице вы увидите информацию о рейсе и вашем билете. Если информация верна, нажмите кнопку "Начать регистрацию".

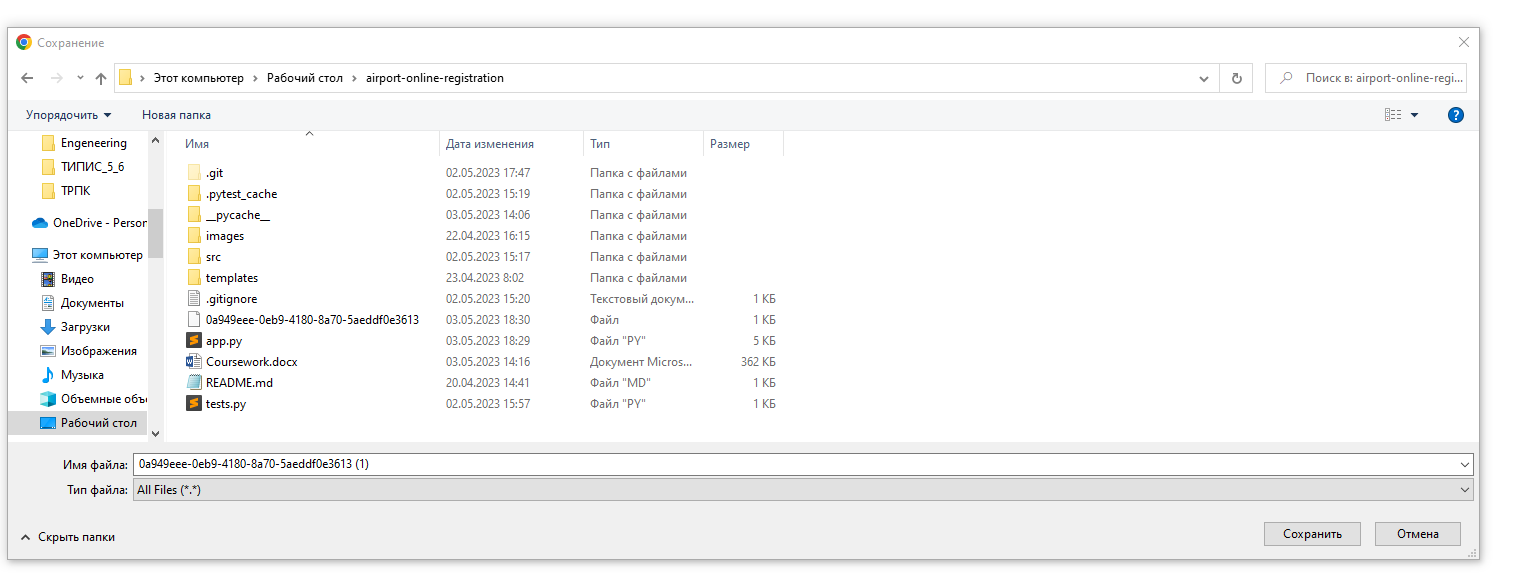


Далее вам будет предложено ввести ваши паспортные данные, такие как Имя, Фамилия, Отчество, Дата рождения, Серия и Номер. После того, как вы введете эту информацию, нажмите кнопку "Продолжить".



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.

## 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).

GitHub репозиторий проекта:

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