**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ**

**КУРСОВАЯ РАБОТА**

**Тема:** Система формирования работы аэропорта

**Работу выполнила** Ушакова Кристина Эдуардовна **группа** K3142

(фамилия, имя, отчество) (номер группы)

**Руководитель**  Белозубов Александр Владимирович

(фамилия, имя, отчество)

**Работа защищена** " " 2019 г.

**с** **оценкой**

**Подписи членов комиссии:**  /

/

Оглавление

[Введение 3](#_Toc533384368)

[Анализ и уточнение задания 4](#_Toc533384369)

[Инфологическая модель базы данных 5](#_Toc533384370)

[Описание интерфейсов пользователей системы 6](#_Toc533384371)

[Описание архитектуры системы 7](#_Toc533384373)

[Описание программного кода системы 9](#_Toc533384374)

[Заключение 12](#_Toc533384380)

# **Введение**

Требуется разработать модуль системы учета расписания рейсов аэропорта и структуру JSON-файла, содержащего данные объектов системы. Для каждого рейса необходимо хранить его название, информацию о времени вылета/прибытия, название пункта отправки и пункта назначения, статус рейса (отменен, задерживается, по расписанию), а также название авиакомпании и модель самолета.

Модуль должен предоставлять возможность получения данных о списке рейсов, хранящихся в базе данных, возможность перехода к интерфейсу просмотра краткой информации о списке изменений расписания работы аэропорта по времени, интерфейсу просмотра подробной информации об отдельном рейсе. База данных представляется JSON-файлом. Средства доступа к базе данных: Python + Django.

Анализ и уточнение задания

Требования к заданию:

Система должна обладать следующими возможностями:

* Возможность получения информации о рейсах, которая хранится в базе данных (в виде JSON файлов)
* Возможность получения дополнительной информацией о рейсах

Аналоги:

**Аналог 1: Google Flights**

1. Указана информация о стоимости провоза багажа.
2. Указана статистика в процентах о задержке рейсов выбранной вами авиакомпании.

**Аналог 2:** **tutu**

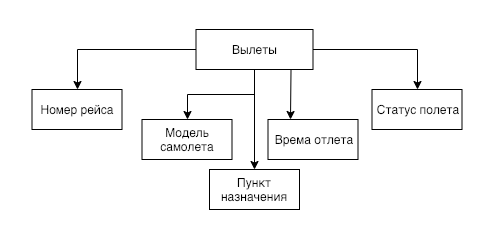
1. Указан проезд до аэропорта в нескольких вариантах.
2. Есть список стран, распределенный по алфавиту, которые вы можете посетить, со списком городов и аэропортов.
3. Есть информация о самолетах, использующихся в мире с указанием производителя и стран, где самолеты производятся.
4. Есть раздел часто задаваемых вопросов, в котором есть ответы на вопросы в роде «Что делать, если рейс отменили» и «Как оплатить багаж в аэропорту»

**Аналог 3:** **Яндекс.Авиабилеты**

1. Имеются отзывы туристов и пассажиров об определенной стране/городе/курорте/авиакомпании
2. Указана статистика ос тоимости пролета, есть возможность подбора более дешевой альтернативы
3. По аналогии с tutu, есть список стран по алфавиту с полной информацией о культуре, городах, курортах, въезде-выезде, визах и так далее.

Инфологическая модель базы данных

Модель параметров для вылетов построена следующим образом: для каждого вылета имеем информацию о номере рейса, модели используемого самолета, пункта назначения, пункта отправления и статуса полета (прилетел вовремя, задерживается, отменен) (Рис. 1).

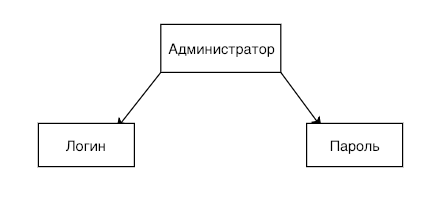
****

*Рис. 1 – модель вылетов*

Модель параметров для прибытий построена аналогичным образом (Рис. 2).

****

*Рис. 2 – модель прибытий*

****

*Рис. 3 – модель параметров администратора*

Описание интерфейсов пользователей системы

В системе имеются два независимых типа пользователя – администратор и гость.

Рассмотрим права гостя системы. При открытии сайта гость видит главную страницу с тремя ссылками (Рис. 10). При переходе по двум из них гость увидит две таблицы с вылетами и прибытиями. Гость также может получить на главной странице контактную информацию для связи с технической поддержкой сайта.

**Права гостя:**

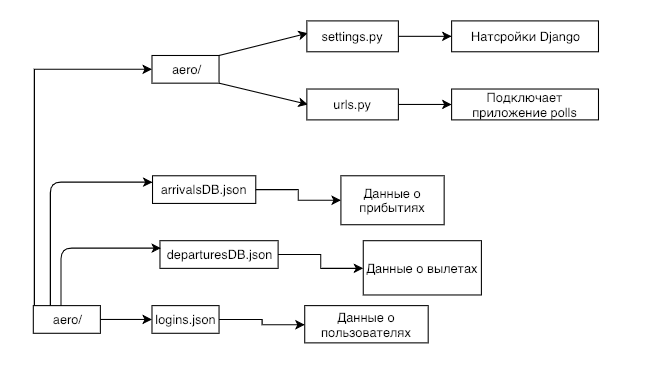
* Просмотр содержимого главной страницы сайта
* Просмотр информации и переход по ссылкам для дальнейнего просмотра информации

Администратор системы имеет админ-панель, доступ к которой осуществляется через всплывающее окно авторизации в верхней части сайта (рис. 12).

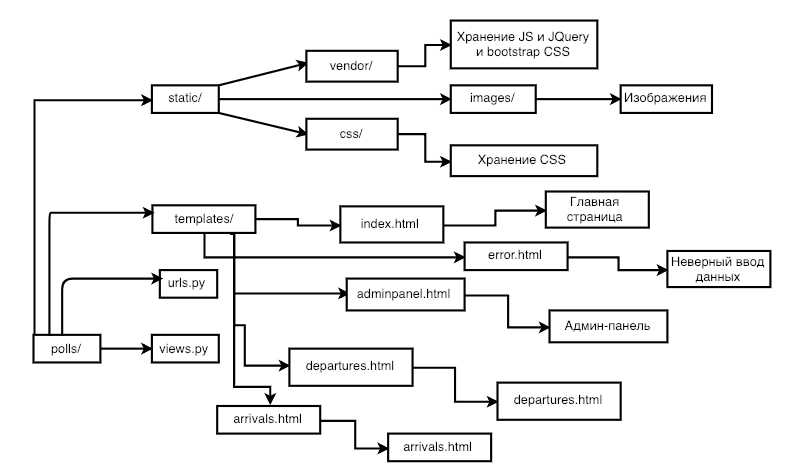
**Права администратора:**

* Добавление и удаление информации в системе

Описание архитектуры системы



*Рис. 4 – схема главной папки aero/*

****

*Рис. 5 – схема папки polls/*

Теперь рассмотрим файлы arrivalsDB.json, departuresDB.json, logins.json. Файлы являются базами данных в формате JSON. Имеют следующее содержание:



*Рис.6 - arrivalsDB.json*

**

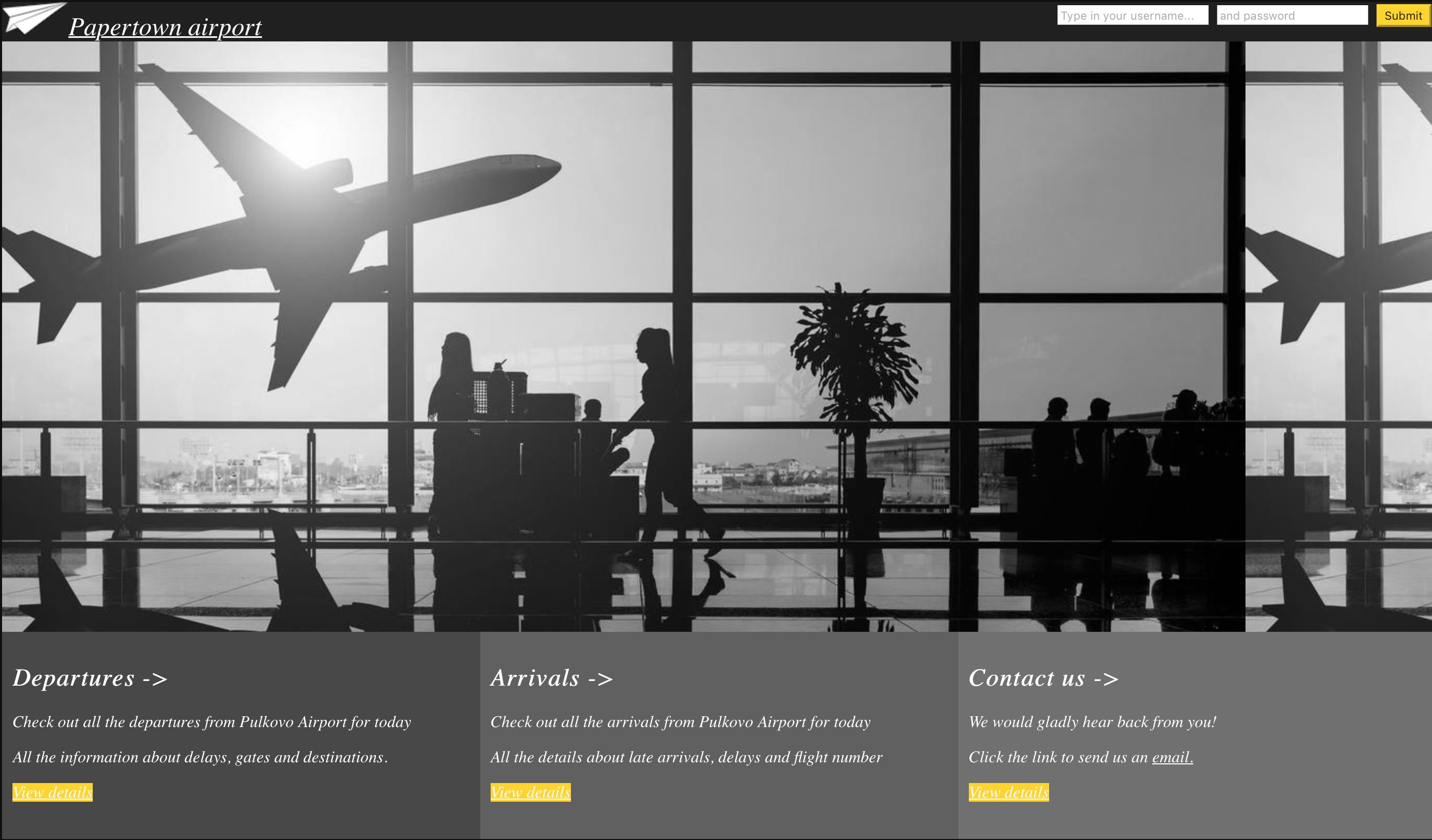
*Рис.7 - departuresDB.json*



*Рис.8 - logins.json*

Описание программного кода системы

1. Главная страница



*Рис.10 –Главная страница - index.html*

На главной странице в навигационной панели имеется кнопка “Отправить”, после нажатия которой включается проверка на правильность ввода данных в поля «Логин» и «Пароль». В случае верного ввода откроется файл adminpanel.html, в ином – error.html. Ниже три кнопки – нажатие на первую и вторую откроет файлы departures.html и arrivals.html соответственно, а третья откроет форму отправки электронного письма системному администратору.



*Рис.11 – departures.html*

**

*Рис.12 – arrivals.html*

Наполнюятся таблицы через функции arrivals(), departures() в views.py.

1. Панель администратора

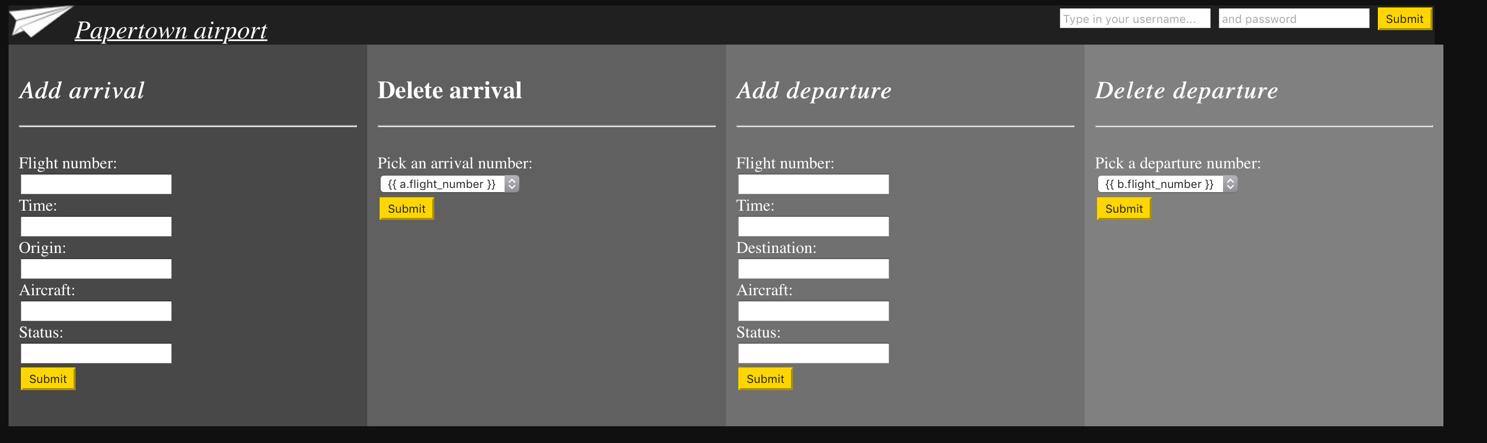
Доступ к панели администратора можно получить, введя логин и пароль в поле «Логин» и «Пароль» соответственно на любой из страниц (рис.10). За вход отвечает функция adminpanel() в файле views.py.



*Рис.13 – error.html*

Первая – добавление рейса. Администратор задает параметры для нового фильма и после нажатия на кнопку “Отправить” рейс появляется на соответствующей странице. За реализацию данного алгоритма отвечает функция addarrival()/adddeparture(), которая посылает POST-запрос в базу данных arrivalsDB.json или departuresDB.json

Вторая часть панели управления позволяет легко удалить рейс из базы данных, выбрав его в списке. За удаление отвечает функция delete(). Список является динамическим и изменяется сразу после добавления/удаления.



*Рис.14 – Удаление и добавление рейсов*

# **Заключение**

В ходе работы над модулем были разработаны структуры JSON-файлов для прибытий и вылетов. Каждый рейс содержит информацию о его номере, пункте назначения/точки вылета, статусе и модели самолета.

**Общее количество строк кода в проекте: 1267**