МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет *компьютерных наук*

Кафедра информационных систем и сетевых технологий

*Техническое задание для web-приложения*

*СИСТЕМА ПОКУПКИ ЭЛЕКТРОННЫХ АВИАБИЛЕТОВ*

*по дисциплине*

*Технологии программирования*

Заказчик: Иванов И. Ю.

Исполнители: Ашурков А. П., Перфильева О. И.

Воронеж2019

# Оглавление

[Введение 3](#_Toc10760522)

[1.Общие сведения 3](#_Toc10760523)

[1.1. Полное наименование системы 3](#_Toc10760524)

[1.2. Заказчик 3](#_Toc10760525)

[1.3. Исполнитель 3](#_Toc10760526)

[1.4. Основания разработки 3](#_Toc10760527)

[1.5. Сроки исполнения работ 4](#_Toc10760528)

[1.6. Порядок оформления и предъявления заказчику результатов работ 4](#_Toc10760529)

[2.Назначение и цели создания (развития) системы 4](#_Toc10760530)

[2.1. Назначение системы 4](#_Toc10760531)

[2.2. Цели создания системы 4](#_Toc10760532)

[3.Характеристика объектов автоматизации 5](#_Toc10760533)

[3.1. Use Сase диаграмма: 5](#_Toc10760534)

[3.2. Диаграмма состояний: 6](#_Toc10760535)

[4.Требования к системе 7](#_Toc10760536)

[4.1. Разработка серверной части 10](#_Toc10760537)

[ER-диаграмма: 10](#_Toc10760538)

[5. Состав и содержание работ по созданию системы 11](#_Toc10760539)

[6. Ограничения в системе 11](#_Toc10760540)

[7. Порядок контроля и приемки системы 11](#_Toc10760541)

[8. Список используемых терминов и сокращений 12](#_Toc10760542)

[Заключение 12](#_Toc10760543)

# Введение

Настоящий документ является Техническим Заданием к проекту «Система покупки электронных авиабилетов»**.** В него входят общие сведения о проекте, цели разработки, требования, пользовательские истории, описание базы данных и информация об интерфейсе. По ходу разработки документ может изменяться и расширяться.

Документ предназначен для использования заказчиком и исполнителем.

Все изменения и уточнения обязательно согласуются с заказчиком.

## 

# 1.Общие сведения

Целью нашего проекта является разработка приложения, позволяющего приобрести, забронировать и (по какой-либо причине) сдать авиабилеты на любой доступный рейс.

### 

## 1.1. Полное наименование системы

«Система покупки электронных авиабилетов»

### 

## 1.2. **Заказчик**

Преподаватель ВГУ, ФКН, Иванов И. Ю.

### 

## 1.3. **Исполнитель**

Студенты ВГУ, ФКН 3 курса 1 группы Ашурков А. П., Перфильева О. И.

### 

## 1.4. Основания разработки

Основанием для исполнения работ по созданию автоматизированной системы, предусмотренной в настоящем ТЗ, является необходимость реализации приложения, надлежащее требованиям заказчика, которое должно выполнять:

1. Просмотр расписания рейсов
2. Возможность бронирования, покупки и сдачи билетов

### 

## 1.5. Сроки исполнения работ

Начало разработки - 25.02.2019

Окончание разработки - 5.06.2019

### 

## 1.6. Порядок оформления и предъявления заказчику результатов работ

Разработанная пилотная версия web-приложения сдается в сроки, установленные настоящим техническим заданием.

Приемка приложения осуществляется комиссией в составе уполномоченных представителей Заказчика и Исполнителя.

Порядок предъявления системы, ее испытаний и окончательной приемки определен в разделе 5 настоящего ТЗ.

## 

# 2.Назначение и цели создания (развития) системы

## 2.1. Назначение системы

Разрабатываемая система предназначена для автоматизации деятельности Пользователей и Администраторов по покупке электронных авиабилетов.

## 2.2. Цели создания системы

Целью данного проекта является реализация приложения, которое должно выполнять:

1)Просмотр расписания рейсов

2)Возможность бронирования, выкупа и сдачи билетов

# 3.Характеристика объектов автоматизации

В системе будут фигурировать две роли: администратор и пользователь. Администратор является привилегированным пользователем, который может добавлять и редактировать информацию.

## 3.1. Use Сase диаграмма:

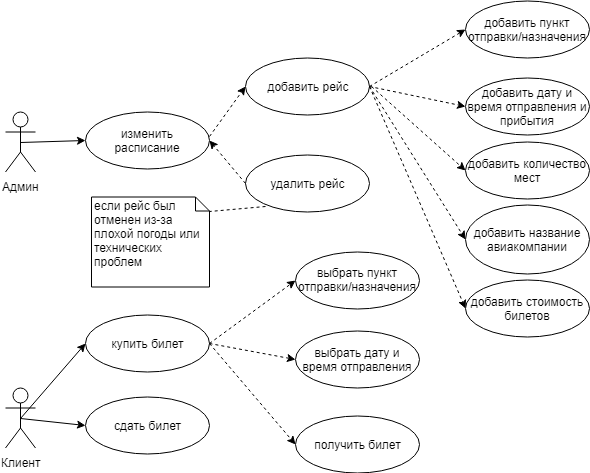
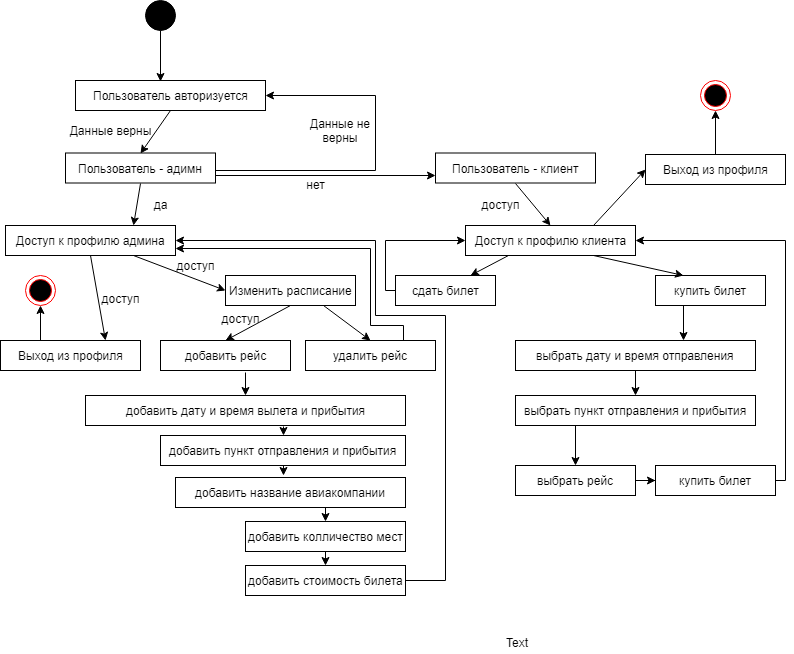


Диаграмма состояний показывает, как объект переходит из одного состояния в другое. В нашем случае, после входа пользователя в систему, он попадет на страницу авторизации, после чего, либо авторизуется как администратор, либо как клиент, если же пользователя еще нет в системе, то он сможет зарегистрироваться. Дальше пользователю откроется доступ к тем или иным действиям, которые он сможет совершить, исходя из того, пользователем какого профиля он будет являться.

## 3.2. Диаграмма состояний:



# 4.Требования к системе

В основе всего проекта лежит «клиент-серверная» технология, которая интересна и актуальна главным образом потому, что обеспечивает простое и относительно дешевое решение проблемы коллективного доступа к базам данных. Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты, потребители этих функций.

В данной работе мы будем использовать HTML, CSS, Python и СУБД SQLite.

* Применение вычислительной техники невозможно без рациональной организации информационной базы и обеспечение эффективного доступа к ней пользователя. Для этой цели и служат базы данных. Широкое использование баз данных различными категориями пользователей привело, с одной стороны, к созданию интерфейсов, требующих минимум времени на освоение средств управления системой, , а с другой - к построению мощных, гибких СУБД, имеющих, в том числе, развитые средства защиты данных от случайного или преднамеренного разрушения. В нашем проекте будем использовать SQLite - это БД которая хранит базу в одном файле и не требует отдельного процесса для запуска, при этом использует не стандартный вариант языка SQL. Такой подход позволяет встроить sqlite прямо в программу, без необходимости установки сервера БД. SQLite, на практике, нередко оказывается в 2-3 раза (и даже больше) быстрее MySQL. Такое возможно благодаря высоко упорядоченной внутренней архитектуре и устранению необходимости в соединениях типа «сервер-клиент» и «клиент-сервер».

Всё это, собранное в один пакет, лишь немногим больше по размеру клиентской части библиотеки MySQL, является впечатляющим достижением для полноценной базы данных. Используя высокоэффективную инфраструктуру, SQLite может работать в крошечном объеме выделяемой для неё памяти, гораздо меньшем, чем в любых других системах БД.

Преимущества:

1. Файловая структура - вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины
2. Отсутствие необходимости настройки сервера СУБД
3. Полностью свободная лицензия
4. Кроссплатформенность
5. Высокая скорость простых операций выборки данных
6. Поддержка транзакций, триггеров, представлений (views), вложенных запросов
7. Безопасность. БД хранится в одном файле, права доступа к которому можно контролировать стандартными средствами ОС
8. Очень экономичная, в плане ресурсов, архитектура

SQLite прекрасно подойдет для проектов у которых мало операций записей, не нужна система прав доступа к БД и ограничены ресурсы сервера.

* HTML (HyperText Markup Language) - язык разметки гипертекста – предназначен для создания Web-страниц. Под гипертекстом в этом случае понимается текст, связанный с другими текстами указателями-ссылками. HTML представляет собой достаточно простой набор кодов, которые описывают структуру документа. HTML позволяет выделить в тексте отдельные логические части (заголовки, абзацы, списки и т.д.), поместить на Web-страницу подготовленную фотографию или картинку, организовать на странице ссылки для связи с другими документами. HTML не задает конкретные и точные атрибуты форматирования документа. Конкретный вид документа окончательно определяет только программа-браузер на компьютере пользователя Интернета.

HTML применяется для того, чтобы определить как должен отображаться ваш контент: в виде абзаца, списка, заголовка, ссылки, изображения, мультимедийного проигрывателя, формы или же в виде одного из множества других доступных элементов, а также возможного нового элемента. У него нет аналогов.

* CSS (англ. Cascading Style Sheets - каскадные таблицы стилей) - формальный язык описания внешнего вида документа, написанного с использованием языка разметки. CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля. Для включения стилей в html-документ применяются - каскадные таблицы стилей CSS (Cascading Style Sheet), с их возможностями значительно упрощается разработка web-страниц и управление полученным сайтом.
* В данном приложении мы будем использовать язык программирования Python - высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

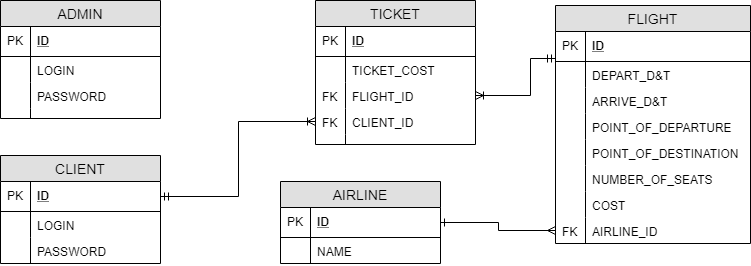
Мы будем использовать Python-фреймворки, т.к. фреймворки позволяют легко и быстро создать базовую логику бэкенда. Она включает в себя сопоставление разных URL-адресов с частями Python-кода, работу с базами данных, создание HTML-представлений для отображения на устройствах пользователя.

В данном приложении мы будем использовать web-фреймворк Flask, т.к. он обеспечивает простоту, гибкость и полный контроль над проектом. Он позволяет пользователю самостоятельно решать, как реализовывать те или иные вещи.

## 4.1. Разработка серверной части

Серверная часть приложения представляет собой базу данных. Разрабатываемая база данных будет состоять из 5 связанных между собой сущностей. Таких таблиц как: ADMIN, CLIENT, FLIGHT, TICKET, AIRLINE.

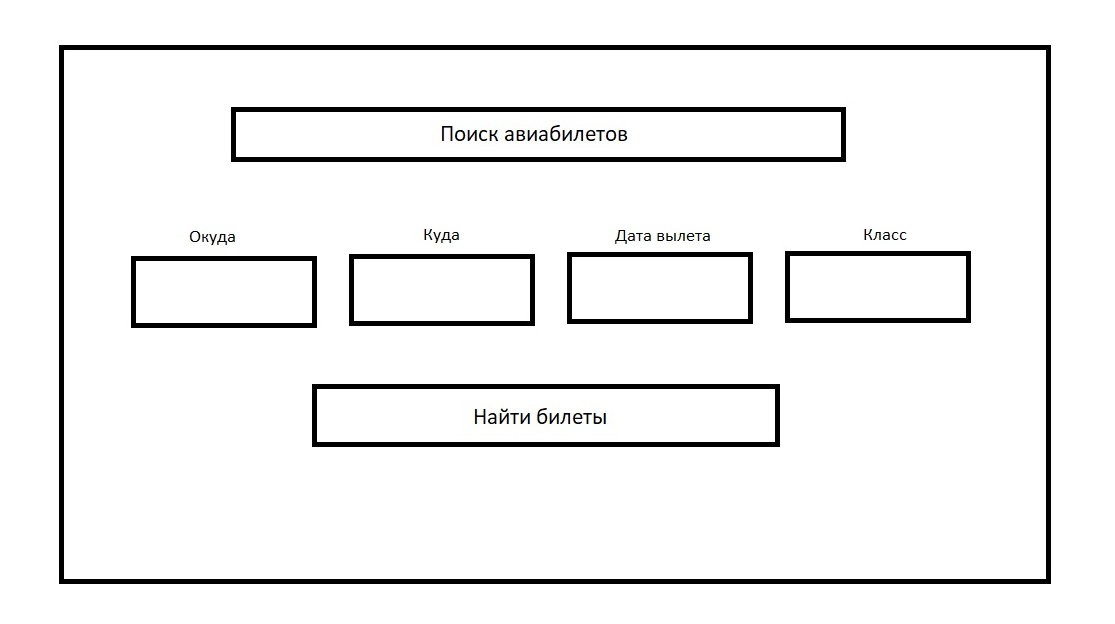
ER-диаграмма:



**4.2. Разработка клиентской части**

Разработка клиентской части подразумевает создание главной страницы сайта, навигации по сайту, а также создание страниц для просмотра, удаления, добавления, поиска и фильтрации данных таблицы. Всё это было создано с помощью html, css, python.

Всего будет 4 страницы. Ниже приведено схематичное представление главной страницы.



На данном представлении главной страницы мы можем наблюдать наличие 4 полей для ввода или выбора нужных данных, заголовок и кнопку “Найти билеты”, после нажатия на которую, мы получим список рейсов или их отсутствие.

# 5. Состав и содержание работ по созданию системы

Стадии работ над проектом выполняются в соответствии с ГОСТ 34 и перечислены ниже.

Стадия 1 - Разработка ТЗ (11.03.2019)

Стадия 2 - Составление плана тестирования (8.04.2019)

Стадия 3 - Выделение функциональных модулей системы, планирование development stories, начало разработки (6.05.2019)

Стадия 4 - Реализация функциональных модулей системы (6.05.2019)

Стадия 5 - Написание отчета по выполненной работе (6.05.2019)

\*Даты, указанные в скобках, являются сроками сдачи той или иной стадии.

# 6. Ограничения в системе

В данном приложении мы ограничиваем функцию бронирования. У нас будет возможна только функция покупки и возврата билета.

## 

# 7. Порядок контроля и приемки системы

Сдача-приёмка работ производится поэтапно, в соответствии с календарным графиком. Основанием для сдачи-приёмки работ служит Отчёт о завершении работ по стадии, предоставляемый Исполнителем. Для сдачи-приёмки представляется также данное ТЗ.

Сдача-приёмка осуществляется комиссией в составе уполномоченных представителей Заказчика и Исполнителя.

Все создаваемые в рамках настоящей работы программные изделия передаются Заказчику, как в виде готовых модулей, так и в виде исходных кодов, представляемых в электронной форме, хранящихся в репозитории на GitHub.

# 8. Список используемых терминов и сокращений

*БД* - база данных

*Администратор* - привилегированный пользователь

*Пользователь* - человек, который использует данное приложение

*СУБД* - система управления базами данных

*MySQL* - это реляционная система управления базами данных

*Реляционный* - выражающий отношение

*Реляционная БД* - это набор данных с предопределенными связями между ними

*HTML* - язык разметки гипертекста – предназначен для создания Web-страниц. Под гипертекстом в этом случае понимается текст, связанный с другими текстами указателями-ссылками

*CSS* - формальный язык описания внешнего вида документа, написанного с использованием языка разметки

*Python* - высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода

*Фреймворк* - программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта

# Заключение

Постоянно увеличивается число пользователей Интернета и лиц, осуществляющих покупки посредством него. Данный проект является удобной WEB - системой, которая даст возможность пользователям заказать или купить нужные билеты, не выходя из дома или офиса. Интернет-магазин удобен для людей, которым не хватает времени для походов по магазинам.

Таким образом, данное приложение должно осуществлять покупку, бронь и возврат билетов на любой (доступный, в случае покупки и брони) рейс.