

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук  
Кафедра технологий обработки и защиты информации

Техническое задание  
в соответствии с ГОСТ 34.602-89

Разработка веб-приложения “Avia Tickets”

Исполнитель _____	<i>Н.В. Дёшин, 3 курс, д/о</i>
Исполнитель _____	<i>А.А. Соловьёв, 3 курс, д/о</i>
Исполнитель _____	<i>Д.Р. Коновалов, 3 курс, д/о</i>
Заказчик _____	<i>В.С. Тарасов</i>

Воронеж 2021

## Содержание

Содержание .....	2
1 Общие сведения .....	4
1.1 Наименование сайта.....	4
1.2 Наименование заказчика .....	4
1.3 Наименование исполнителя .....	4
1.4 Основание для разработки .....	4
1.6 Термины и сокращения .....	4
2 Назначение и цели создания .....	7
2.1 Назначение и цели системы .....	7
2.2 Цели создания системы .....	7
3 Характеристика объектов автоматизации.....	8
4 Требования к системе .....	9
4.1 Требования к системе в целом .....	9
4.1.1 Требования к структуре и функционированию системы.....	9
4.1.3 Требования к защите информации от несанкционированного доступа ...	10
4.1.4 Требования к аутентификации .....	10
4.1.5 Требования к численности и квалификации персонала.....	10
4.1.6 Требования к патентной чистоте.....	11
4.1.7 Требования к масштабируемости и открытости.....	11
4.1.8 Обработка ошибок.....	11
4.2 Требования к функциям, выполняемым системой .....	11
4.3 Требования к видам обеспечения .....	12
4.3.1 Требования к информационному обеспечению.....	12
4.3.2 Требования к программному обеспечению.....	12
4.3.3 Требования к лингвистическому обеспечению .....	13
4.3.4 Требования к техническому обеспечению .....	13
5 Состав и содержание работ по созданию (развитию) системы .....	14
6 Порядок контроля и приемки системы .....	16
7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие .....	17
8 Требования к документированию.....	18

9. Источники разработки .....	19
-------------------------------	----

## **1 Общие сведения**

### **1.1 Наименование сайта**

Полное наименование: Система поиска и продажи авиабилетов  
“Avia Tickets”

Краткое наименование “Avia Tickets”

### **1.2 Наименование заказчика**

Старший преподаватель Тарасов Вячеслав Сергеевич, кафедра программирования и информационных технологий.

### **1.3 Наименование исполнителя**

Студент Дёшин Никита Витальевич, кафедра обработки изображений и машинного обучения.

Студент Соловьёв Артём Анатольевич, кафедра обработки изображений и машинного обучения.

Студент Коновалов Даниил Романович, кафедра обработки изображений и машинного обучения.

### **1.4 Основание для разработки**

Необходимость заказчика в наличие веб-приложения, искать и покупать электронные авиабилеты.

### **1.5 Плановые сроки начала и окончания работ**

Плановый срок начала работ - Март 2020 г.

Плановый срок окончания работ – Июнь 2020г.

### **1.6 Термины и сокращения**

**MVC (Model-View-Controller)** - схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

**Model (база данных)** - этот компонент отвечает за данные в проекте, а также реагирует на команды контроллера, изменяя свое состояние.

**Представление данных пользователю, Графический интерфейс(View)** - это компонент, отвечающий за взаимодействие с пользователем. То есть данный компонент определяет внешний вид приложения и способы его использования.

**Контроллер (Controller)** - это компонент, который управляет запросами пользователя. Его основная функция — вызывать и координировать действие необходимых ресурсов и объектов, нужных для выполнения действий, задаваемых пользователем. Обычно контроллер вызывает соответствующую модель для задачи и выбирает подходящий вид.

**front-end** - клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

**back-end** - программно-аппаратная часть сервиса.

**REST API** - это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов.

**GitHub** - крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

**Веб-сервис, интернет-сервис, система, веб-приложение, проект** - идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя.

**Header** - визуальный элемент, расположенный в верхней части страницы.

**Footer** - визуальный элемент, расположенный в нижней части страницы.

**Гость-неавторизованный** на веб-сервисе человек, пользующийся ограниченным функционалом веб-сервиса.

**Авторизованный пользователь** - авторизованный на портале человек, пользующийся функционалом веб-сервиса.

**Администратор** - человек, имеющий доступ к расширенному функционалу веб-сервиса.

**Личный кабинет** - это раздел сервиса, в котором Пользователь может

получить доступ к своим данным

**Контент** - наполнение сайта.

**Jwt** - это открытый стандарт (RFC 7519) для создания токенов доступа, основанный на формате JSON. Как правило, используется для передачи данных для аутентификации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения своей личности.

**Spring Framework** - универсальный фреймворк с открытым исходным кодом для Java-платформы.

**Spring Security** - фреймворк, предоставляющий механизмы построения систем аутентификации и авторизации, а также другие возможности обеспечения безопасности для промышленных приложений, созданных с помощью Spring Framework

**Angular** - открытая и свободная платформа для разработки веб-приложений, написанная на языке TypeScript.

**Authorization header** - Заголовок HTTP запроса, который включает в себя данные пользователя для проверки подлинности пользовательского агента с сервером обычно после того, как сервер ответил со статусом 401 Unauthorized и заголовком WWW-Authenticate (en-US).

## **2 Назначение и цели создания**

### **2.1 Назначение и цели системы**

Назначением системы является реализация процесса поиска и покупки авиабилетов.

### **2.2 Цели создания системы**

Данный проект является веб-сервисом поиска и покупки авиабилетов.

Цели создания:

- Реализация сервиса по поиску авиабилетов.
- Реализация возможности дистанционной покупки авиабилетов. (В рамках учебного проекта, было решено использовать заглушку покупки, то есть пользователь может ввести данные своей карты, их проверят на валидность, но самой транзакции не произойдет)

### **3 Характеристика объектов автоматизации**

Данная система предназначена для заказчика, которому необходим веб-сервис поиска и покупки авиабилетов. Для администрирования данного продукта необходим минимум один сотрудник из штата заказчика.

Для взаимодействия с данной системой выделяют следующие виды пользователей:

- Гость
- Авторизированный пользователь
- Администратор

Гость имеет возможность в процессе взаимодействия с данной системы:

- Пользоваться поиском авиабилетов
- Пользоваться фильтрами поиска
- Регистрироваться и авторизоваться на сайте

Авторизированный пользователь имеет возможность в процессе взаимодействия с данной системы:

- Пользоваться поиском авиабилетов
- Пользоваться фильтрами поиска
- Изменять персональные данные в личном кабинете
- Покупки одного так и сразу нескольких билетов путём добавления в корзину
- Просмотра рекомендованных билетов, при указании персональных данных в личном кабинете пользователя, если пользователь укажет свой город проживания, то ему будут дополнительно предложены рейсы из его города, на странице поиска билетов

Администратор имеет возможность в процессе взаимодействия с данной системы:

- Удалять пользователей.
- Удалять авиабилеты



- Добавлять авиабилеты
- Редактировать авиабилеты

## **4 Требования к системе**

### **4.1 Требования к системе в целом**

Разрабатываемый проект должен удовлетворять следующим основным требованиям:

- Должен стабильно работать в браузере Google Chrome начиная от версии 89.0.4389.90.
- Иметь пользовательский интерфейс.
- Реализовывать задачи, стоящие перед данным проектом.
  - Поиск авиабилетов
  - Использование фильтров по поиску авиабилетов
  - Покупка авиабилетов
  - Удаление, редактирование и добавление авиабилетов администратором.

#### **4.1.1 Требования к структуре и функционированию системы**

Созданное приложение будет иметь архитектуру, соответствующую шаблону MVC, а также иметь разделение на back-end и front-end, взаимодействие между которыми происходит с помощью REST API.

Стек используемых технологий (может меняться и дополняться в виду отсутствия конкретных требований заказчика, не противореча требованиям системы установленных в текущем документе):

Back-end:

- Java8

Front-end:

- HTML
- CSS
- Angular

### **4.1.3 Требования к защите информации от несанкционированного доступа**

Система должны быть надёжно защищены от самых распространённых способов взлома сайтов и программ, по мнению: OWASP Top 10 Application Security Risks 2020, а именно от SQL инъекций.

Так же приложение хранит пароли пользователей в зашифрованном виде, по средствам использования BCryptPasswordEncoder класса, содержащегося в Spring Security.

### **4.1.4 Требования к аутентификации**

- Должны осуществляться идентификация и проверка доступа при входе в систему по логину и паролю.
- Система защиты должна подвергать проверку подлинности идентификации путем аутентификации.
- Система защиты должна препятствовать доступу к защищаемым ресурсам не идентифицированным пользователям и пользователей не прошедших идентификацию. Это будет реализовано, средствами Spring Security и Jwt токена. Токен будет отправляться в Authorization header при выполнении запросов к защищенным ресурсам.

### **4.1.5 Требования к численности и квалификации персонала**

Пользователь с ролью администратор должен обладать знаниями и навыками необходимыми для поддержания работоспособности системы.

#### **4.1.6 Требования к патентной чистоте**

Данный проект должен не нарушать никаких лицензий и патентов. В случае нарушения всю ответственность несет сторона Исполнителя.

#### **4.1.7 Требования к масштабируемости и открытости**

Проект должен предоставлять возможность добавлять новую функциональность без изменения базового функционала приложения.

#### **4.1.8 Обработка ошибок**

В случае возникновения ошибок пользователь должен получать соответствующие сообщение об ошибке.

Приложение должно поддерживать обработку следующих основных ошибок:

- Некорректно введенные данные при регистрации, авторизации и оплате авиабилета;

#### **4.2 Требования к функциям, выполняемым системой**

Система должна удовлетворять следующим функциональным требованиям:

- Приложение должно предоставлять следующие возможности для неавторизованного пользователя:
  - Возможность поиска авиабилета
  - Возможность использования фильтров поиска авиабилета
  - Возможность регистрироваться и авторизоваться на сайте
- Приложение должно предоставлять следующие возможности для авторизованного пользователя:
  - Возможность поиска авиабилета
  - Возможность использования фильтров поиска авиабилета
  - Возможность просмотра личного кабинета
  - Возможность редактирования персональных данных в личном кабинете
  - Возможность добавить в корзину билет
  - Возможность удалить билет из корзины

- Возможность купить все билеты из корзины
- Возможность просматривать историю покупки
- Приложение должно предоставлять дополнительные возможности для администратора приложения, эти возможности будут, предоставлены через панель администратора. Дополнительные возможности:
  - Возможность удалять авиабилеты
  - Возможность удалить пользователя
  - Возможность редактировать авиабилеты
  - Возможность добавлять авиабилеты

### **4.3 Требования к видам обеспечения**

#### **4.3.1 Требования к информационному обеспечению**

Требования по применению систем управления базами данных: использование СУБД PostgreSQL.

#### **4.3.2 Требования к программному обеспечению**

Основные требования к программному обеспечению:

- PostgreSQL
- Spring Boot
- Angular

В результате разработки данные требования могут расширяться.

### **4.3.3 Требования к лингвистическому обеспечению**

Приложение должно поддерживать русский язык. Язык ввода-вывода данных русский.

При реализации клиентской части системы должен использоваться фреймворк Angular.

Для реализации серверной части должен использоваться высокоуровневый язык программирования Java с использованием фреймворка Spring.

Для реализации алгоритмов манипулирования данными в системе поиска команд необходимо использовать стандартный язык запроса к данным SQL.

Для организации диалога системы с пользователем должен применяться графический пользовательский интерфейс.

### **4.3.4 Требования к техническому обеспечению**

Система должна выполняться в браузере Google Chrome начиная от версии 89.0.4389.90.

## 5 Состав и содержание работ по созданию (развитию) системы

Таблица 1 – состав работ.

Этап	Содержание работ	Порядок приемки и документы	Сроки	Ответственный
1. Составление ТЗ	Разработка функциональных и нефункциональных требований к системе	Утверждение ТЗ	25.03.2021, 18.00 по МСК	Разработка — Исполнитель; Согласование — Заказчик
2. Техническое проектирование	Разработка дизайн-макета публичного веб-приложения	Ссылки на Figma.com, Miro.com	25.03.2021 18.00 по МСК.	Исполнитель
3. Разработка программной части	Разработка серверного модуля, модуля хранения данных. Внедрение модуля хранения файлов	Приемка осуществляется в процессе испытаний	В течение 56 дней со дня утверждения ТЗ (25.03.2021 )	Исполнитель
	Разработка панели администрирования			Исполнитель
	Разработка схемы развертывания данной системы на сервере			Исполнитель
4. Тестирование на локальном сервере	<ul style="list-style-type: none"> <li>— Проверка соответствия (не)функциональным требованиям.</li> <li>— Проверка комплекта документации.</li> <li>— Доработки и повторные испытания до устранения недостатков</li> </ul>	Согласованность с ТЗ  Ведение журнала ошибок и исправлений.	2 дня с дня завершения разработки (20.05.2021 )	Исполнитель
5. Тестирование на хостинге с поддержкой java spring и PostgreSQL	<p>Проверка взаимодействия со смежными внешними системами</p> <p>Доработки и повторные испытания до устранения недостатков</p>		До 5 дней после завершения автономных испытаний (20.05.2021 )	Исполнитель
6. Разработка Курсового	Разработка Курсового проекта, содержащего	С начала формирования ТЗ и о	25.03.2021	Исполнитель

проекта и документации	аналитическую информацию о проекте на основе ТЗ	опытной эксплуатации проекта		
8. Опытная эксплуатация	Эксплуатация с привлечением небольшого количества участников (несколько аукционов среди знакомых). Доработки и повторные испытания до устранения недостатков	Ведение советующего внутреннего документа	25-27.05.2021	Исполнитель
9. Коммерческая эксплуатация	Коммерческая эксплуатация системы	Соответствие ТЗ и Курсовому проекту	30.05.2021	Исполнитель
10. Поддержка	Осуществление работоспособности приложения, добавление новых функций и устранение возникающих ошибок	Ведение советующего внутреннего документа	30.05.2021	Исполнитель

## **6 Порядок контроля и приемки системы**

Контроль разработки системы осуществляется посредством запланированных встреч между руководителем данного проекта и заказчиком. Готовая система с полной документацией будет представлена заказчику в запланированную согласно данному документу дату. Заказчик определит соответствие системы его требованиям и осуществит её приём.

Исполнитель должен предоставить следующий комплект поставки при сдаче проекта:

- Техническое задание
- Тестовые сценарии
- Демонстрационная версия проекта со всеми ключевыми сценариями
- Аналитику проекта
- Исходный код Системы
- Исполняемые модули Системы



## **7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**

При подготовке объекта автоматизации к вводу системы в действие необходимо выполнить следующие работы (сторона исполнителя за это не отвечает):

- Необходимо осуществить, набор нового персонала, в случае необходимости(одного администратора).
- Провести обучение персонала, ознакомить персонал с интерфейсом системы.
- Для веб-приложений: разработка общих разделов сайта и пользовательского соглашения(согласия на обработку персональных данных).
- Развертывание системы на глобальных коммерческих серверах.
- Необходимо провести настройку системы доступа и создание учетных записей.

## **8 Требования к документированию**

Документирование проекта в рамках Технического Задания ведётся в соответствии с ГОСТ 34.602-89.

Вся документация должна быть подготовлена и передана, как в печатном, так и в электронном виде (в формате docx или pdf), а также размещена на GitHub.

Также осуществляется предоставление Курсового проекта на основе данного Технического Задания.

## 9. Источники разработки

1. Техническое задание на создание автоматизированной системы [Текст]  
ГОСТ 34.602-89. – Изд. июнь 2009 г. – Взамен ГОСТ 24.201-85; введ.  
24.03.89
2. OWASP Top Ten / [сайт]. – URL: <https://owasp.org/www-project-top-ten/>  
(дата обращения 15.03.2021)
3. Spring / [сайт]. – URL: <https://spring.io/projects/spring-boot>(дата обращения  
20.03.2021)