

**Министерство образования Республики Беларусь**

**Учреждение образования**

**«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»**

**Факультет компьютерных систем и сетей**

**Кафедра информатики**

**Техническое задание «Создание серверного приложения для  
управления авиаперелётами»**

**Авторы:**

**Решетнёв Арсений Алексеевич 253504**

**Минск 2024**

## **1. Введение**

Целью данного технического задания является разработка серверного приложения для управления авиаперелетами. Приложение должно быть реализовано с использованием языка программирования Python, базы данных SQL и контейнеризации с помощью Docker. Разработка будет основана на принципах объектно-ориентированного программирования (ООП).

## **2. Основания для разработки**

Существует потребность в эффективной системе управления авиаперелетами, которая позволит авиакомпаниям и пассажирам взаимодействовать для бронирования, управления и отслеживания авиаперелетов. Такая система поможет оптимизировать процессы и повысить качество обслуживания.

## **3. Назначение разработки**

Назначением разработки является создание серверного приложения, которое будет обеспечивать требуемую функциональность.

## **4. Требования к программе**

*Функциональные требования:*

- Регистрация и аутентификация пользователей.
- Поиск и бронирование доступных авиабилетов.
- Управление информацией о рейсах, расписаниях и доступности мест.
- Отслеживание статуса авиаперелётов и уведомление пассажиров об изменениях.
- Генерация отчётов и статистики по авиаперелётам.

*Нефункциональные требования:*

- Приложение должно быть реализовано с использованием объектно-ориентированного программирования (ООП).
- Приложение должно быть масштабируемым и обеспечивать высокую производительность.
- Безопасность данных и аутентификация пользователей должны быть обеспечены.
- Приложение должно быть легко развертываемым с использованием контейнеризации Docker.

- Система должна поддерживать работу с большим объемом данных и обеспечивать высокую отказоустойчивость.

## **5. Применяемые технологии**

- Язык программирования: Python.
- База данных: SQL (например, PostgreSQL, MySQL).
- Контейнеризация: Docker.

## **6. Требования к программной документации**

Для данного проекта требуется разработка следующей программной документации:

- Техническое задание (данный документ).
- Диаграммы классов и последовательностей.
- Документация к API приложения.
- Руководство пользователя.

## **7. Технико-экономические показатели**

### **Технические показатели:**

Проект пригоден для самых различных групп лиц.

### **Экономические показатели:**

Для создания проекта будут использоваться только бесплатные инструменты.

## **8. Стадии и этапы разработки**

Разработка серверного приложения для управления авиаперелётами может быть разделена на следующие стадии и этапы:

Стадия 1: Планирование и анализ

- Определение требований и функциональностей приложения.
- Анализ возможных рисков и оценка необходимых ресурсов.
- Разработка плана проекта и определение этапов разработки.

Стадия 2: Проектирование

- Разработка архитектуры приложения.

- Создание диаграмм классов и последовательностей.
- Определение структуры базы данных и схемы хранения данных.

### Стадия 3: Реализация

- Создание серверной части приложения на языке Python с использованием ООП.
- Разработка взаимодействия с базой данных с использованием SQL.
- Развертывание приложения в контейнерах Docker.

### Стадия 4: Тестирование

- Проведение модульного, интеграционного и системного тестирования приложения.
- Выявление и устранение ошибок и неполадок.
- Проверка соответствия приложения требованиям.

### Стадия 5: Внедрение и поддержка

- Установка и настройка приложения на сервере.
- Проведение пользовательского тестирования и обучение пользователей.
- Поддержка и обновление приложения, включая исправление ошибок и добавление новых функций.

## **9. Порядок контроля и приёмки**

Порядок контроля и приёмки разработанного приложения будет определен в соответствии с внутренними процедурами, установленными университетом и его руководством.

## **10. Приложения**

- UML-диаграмма классов

