Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

на информационную систему «Аэропорт»

Вариант 10

Руководитель \_\_\_\_\_\_\_\_\_\_ Сидоров А.Ю.

подпись, дата

Студент КИ19–09Б, 031938808 \_\_\_\_\_\_\_\_\_\_ Бичаева В.А.

подпись, дата

Красноярск 2020

# **Обзор.**

**Задание курсового проекта:**

Информационная система «Аэропорт». Информация: рейсы – номер, пункт отправления, пункт назначения, марка самолета, число мест, число свободных мест, список пассажиров; пассажиры – фамилия, имя, отчество, паспорт, рейс. Операции – продажа билетов на определенный рейс, возврат проданных билетов в кассу. Предусмотреть следующие группы пользователей: администратор, кассир, пассажир.

Информационная система «Аэропорт» предназначена для автоматизации продажи, покупки билетов потребителями на авиарейсы различных авиакомпаний.

От программы требуется предусмотреть корректность введённых данных, а также поиск данных по заданным параметрам.

От разработчика требуется создание понятного для любого пользователя интерфейса. Также необходимо предусмотреть разделение прав различных групп пользователей (администратор, кассир, пассажир).

В качестве платформы для разработки информационной системы «Аэропорт» выбран кроссплатформенный фреймворк Qt на языке программирования С++.

# 2. Модель программы.

Пользовательский интерфейс программы должен содержать:

1. Окно авторизации или же окно регистрации при первом запуске программы.

2. Окно поиска рейсов, учитывающий дату, место вылета, пункт назначения.

3. Окно покупки и возврата билетов.

После авторизации пользователь попадает в меню выбора дат вылета и прибытия, а также место вылета ( название аэропорта) и пункт назначения. После того, как пользователь указал необходимые параметры, выдаются все доступные рейсы, удовлетворяющие параметрам поиска. Если же пользователь нашел интересующий его рейс, то он может приобрести билет(ы) на рейс. Окно покупки вызывается при нажатии кнопки «Купить билет(ы)». Возврат купленных билетов возможен из главного меню.

Администратор будет иметь полномочия увольнения кассира, добавления и блокировки пользователей.

# 3. Форматы входных и выходных данных.

Предполагается, что данные администраторов, кассиров, пользователей и рейсов будут записываться в разные структуры.

Все входные данные будут заноситься через визуальный интерфейс программы группами пользователей.

Все выходные данные будут записываться в двоичные файлы.

# 4. Тестирование.

Для каждого разрабатываемого модуля будет составлен и осуществлён индивидуальный набор тестов, сгруппированных в тест-кейсы.

**Тест-кейс «Авторизация»:**

1. Если незарегистрированный пользователь пытается авторизоваться, то он получает отказ в авторизации.

2. Если зарегистрированный пользователь пытается авторизоваться, то он получает подтверждение об авторизации в системе.

3. Если пользователь вводит неверные данные, то выдается ошибка о некорректности введённых им данных.

**Тест-кейс «Регистрация»:**

1. Если пользователь пытается ввести неверные данные, то выдается ошибка о некорректности введенных им данных.

**Тест-кейс «Покупка билета»:**

1. Если пользователь решает купить билет и заполняет все необходимые поля, то покупка билета проходит успешно.

2. Если пользователь пытается купить билеты без заполнения каких-либо ключевых полей, то выдается ошибка о необходимости заполнить пропущенные поля.

**Тест-кейс «Возврат билета»:**

1. Если пользователь осуществляет возврат купленного им билета, то возврат произведен успешно.

2. Если пользователь пытается вернуть билеты, которые он не покупал, то выдается ошибка о невозможности выполнения требуемого действий.

# 6. График выполнения курсового проекта.

* 9 октября – 20 октября: Разработка интерфейса. Реализация авторизации/регистрации;
* 20 октября – 1 ноября: Написание основного кода программы;
* 1 октября – 15 ноября: Реализация функции управления пользователями. Создание групп пользователей, управление правами. Завершение разработки программы;
* 17 ноября – 29 ноября: Тестирование программы на основе тест-кейсов.
* 30 ноября – 10 декабря: Написание пояснительной записки.