Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники.

**Техническое задание**

Для информационной системы «Табло авиарейсов»

Преподаватель \_\_\_\_\_\_\_\_\_ Сидоров А.Ю.

подпись, дата

Студент КИ19-06б, 031940348 \_\_\_\_\_\_\_\_\_ Маслов Е.А.

подпись, дата

Красноярск 2020

**СОДЕРЖАНИЕ**

Введение...........................................................................................................3

1. Область применения...............................................................................3
2. Основание для разработки……………………………………………….3
3. Требования к програме…….........................................................................3

3.1. Требования к функциональным характеристикам программы…….3

3.2. Требования к надежности программы……………………………….5

3.3. Требования к информационной и программной совместимости…5

3.4. Условия эксплуатации……………………………………………….5

1. Требования к программной документации……………………………….6
2. Описание входных и выходных данных………………………………..6
3. Диаграмма прецедентов………………………………………………….8
4. Стадии и этапы разработки………………………………………………10

7.1. Стадии разработки………………………………………………….10

7.2. Этапы разработки……………………………………………………11

1. Порядок контроля и приёмки…………………………………………..12

ПРИЛОЖЕНИЕ………………………………………………………………13

**Введение**

Информационная система «Табло авиарейсов».

Предназначена для автоматизации работы с пассажирами, позволяет пассажирам просматривать, а также фильтровать рейсы на информационном табло, позволяет диспетчерам вносить изменения на табло авиарейсов(добавлять новые рейсы, удалять и изменять), предупреждать пассажиров о каком либо событии связанным с рейсами (рейс отложен и т.п.).

1. **Область применения**

Программа может применяться в аэропортах.

1. **Основание для разработки**

Разработка информационной системы ведётся в рамках дисциплины “Междисциплинарный курсовой проект базового уровня”

Согласно учебному плану для получения зачёта.

1. **Требования к программе**
   1. **Требования к функциональным характеристикам программы**

Анализируя задание, разрабатываемая система поддержки информационного табло, должна обеспечивать следующие функции:

1. Отображение на табло информации о рейсах:
   1. Номер рейса.
   2. Пункт вылета.
   3. Время вылета.
   4. Пункт прилёта.
   5. Время прилёта.

1.6. Статус (отложен, вылетел, прилетел).

1. Изменение в состояние табло (информация о задержке рейса и т. д).
2. Ввод и хранение данных о пассажирах:

3.1. Логин.

3.2. Пароль.

1. Ввод и хранение информации о диспетчерах:

4.1 Логин.

4.2 Пароль.

С данной информационной системой должны работать следующие группы пользователей:

1. Пассажир.
2. Диспетчер.
3. Администратор.

При работе с системой пассажир должен иметь возможность:

1. Просматривать информацию о рейсах.
2. Фильтровать информацию на табло.

При работе с системой диспетчер должен иметь возможность:

1. Вносить изменения в состояние табло, если произошло некоторое событие (например, вылет рейса отложен на N минут, произошла посадка самолета указанного рейса и т. д).

При работе с системой администратор должен иметь возможность:

1. Вносить изменения о пользователях.
2. Вносить изменения в состояние табло, если произошло некоторое событие (например, вылет рейса отложен на N минут, произошла посадка самолета указанного рейса и т. д).
   1. **Требования к надежности программы**

Информационная система должна обеспечивать проверку корректности входных и выходных данных, должна работать устойчиво.

* 1. **Требования к информационной и программной совместимости**

Проект разрабатывается на языке программирования C++ средствами графической библиотеки QT.

* 1. **Условия эксплуатации**

Для эксплуатации информационной системы “Табло авиарейсов” от пользователя требуется умение работать на персональном компьютере.

Программа не требует установленного Framework QT на компьютере пользователя.

1. **Требования к программной документации**

Программная документация для данного проекта делится на техническое задание и пояснительную записку. Программная документация должна полностью отображать возможности и цели программы. Техническое задание объясняет какой будет программа и ставит определенную цель выполнения проекта. Пояснительная записка, в свою очередь, должна полностью объяснять работу программы, содержать в себе макеты интерфейса и описание всех уже прописанных классов, заголовочных файлов, файлов основного программного кода, всю связь между слотами и программным кодом.

1. **Описание входных и выходных данных**

**Входные данные**:

Файл: **users.bin**.

Формат файла: двоичный файл.

Описание: В файле **users.bin** храняться данные о пользователях.

(Логин, пароль, статус.)

Типы данных:

Login – тип QString, может содержать числа и буквы.

password - тип QString, может содержать числа и буквы.

При попытке войти в систему, эти данные загружаются и сверяются с введёнными пользователем.

Файл: **planes.bin**.

Формат файла: двоичный файл.

Описание: В файле **planes.bin** храняться данные о рейсах.

(Номер рейса, Пункт вылета, Время вылета, Пункт прилёта, Время прилёта, Статус (отложен, вылетел, прилетел))

Типы данных:

number – QString.

Departure – QString.

Departure\_time – QString.

Arrival – QString.

Arrival\_time – QString.

Status - QString.

При открытии главного окна пользователя, эти данные загружаться и выводятся на экран в виде списка рейсов, в таком же формате как и хранятся в файле.

**Выходные данные**:

Файл: **users.bin**.

Формат файла: двоичный файл.

Описание: В файле **users.bin** храняться данные о пользователях.

(Логин, пароль.)

Типы данных:

Login – тип QString, может содержать числа и буквы.

password - тип QString, может содержать числа и буквы.

В файл вносятся изменения о пользователях.

Изменять, добавлять, удалять данные в этом файле имеет возможность только администратор.

Файл: **planes.bin**.

Формат файла: двоичный файл.

Описание: В файле **planes.bin** храняться данные о рейсах.

(Номер рейса, Пункт вылета, Время вылета, Пункт прилёта, Время прилёта, Статус (отложен, вылетел, прилетел))

Типы данных:

number – QString.

Departure – QString.

Departure\_time – QString.

Arrival – QString.

Arrival\_time – QString.

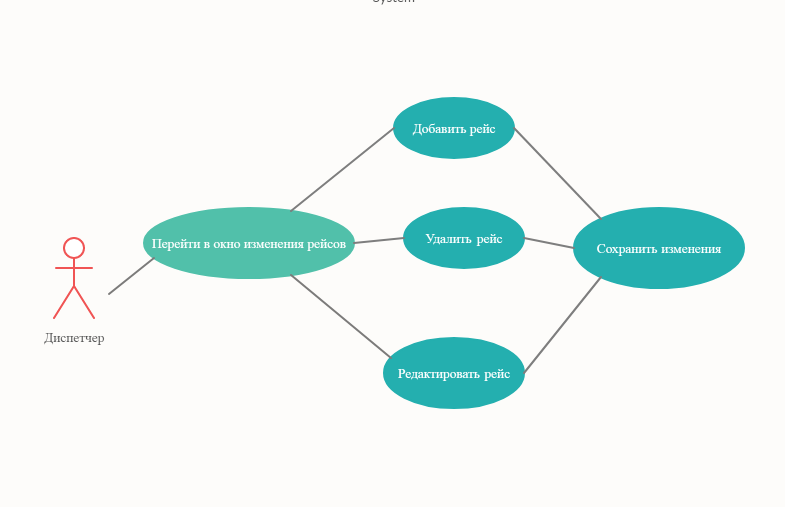
Status - QString.

При изменении рейса, удалении рейса и добавления нового рейса, данные в файле обновляются.

Вносить изменения в файл могут диспетчер и администратор.

1. **Диаграмма прецедентов**

Диаграмма прецедентов диспетчера:

****

**Название прецедента:**Перейти в окно изменения рейсов

**Действующее лицо:**Диспетчер

**Цель:**Изменение данных о рейсах

**Предусловия:**Диспетчер осуществил вход в систему

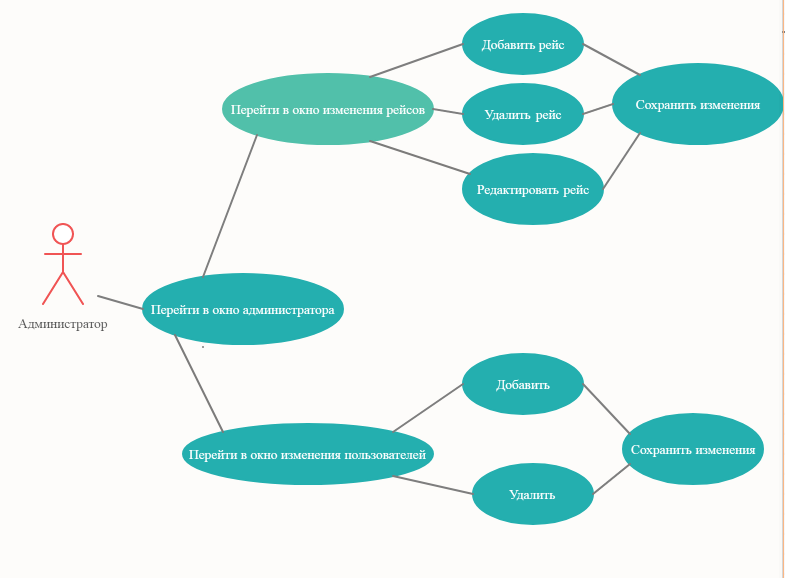
**Главная последовательность:**

1. Диспетчер в главном окне выбирает конкретный рейс двойным кликом мыши по нему.
2. Открывается окно диспетчера изменения рейсов.
3. Диспетчер выбирает пункт “Управление рейсами”.
4. Далее диспетчер выбирает конкретный рейс и нажимает одну из кнопок: “Удалить”, ”Добавить”, ”Редактировать”.
5. При нажатии на кнопку “Редактировать”, открывается окно редактирования / добавления рейса, диспетчер вносит необходимые поправки в рейс и нажимает кнопку

“ Ок”, тем самым сохраняя изменения о рейсе.

1. При нажатии на кнопку “Удалить”, рейс удаляется.
2. При нажатии на кнопку “Добавить”, открывается окно редактирования / добавления рейса, где диспетчер вносит данные о рейсе и нажимает кнопку “ Ок”, тем самым сохраняя и добавляя новый рейс.
3. После любого из данных действий, диспетчер нажимает кнопку “Принять”, после чего открывается главное окно.

Диаграмма прецедентов администратора:



**Название прецедента:**Перейти в окно изменения рейсов

**Действующее лицо:**Администратор

**Цель:**Изменение данных о рейсах

**Предусловия:**Администратор осуществил вход в систему

**Главная последовательность:**

1. Администратор в главном окне нажимает кнопку “Администрирование”.
2. Открывается окноадминистратора.
3. Администратор выбирает пункт “Управление рейсами”.
4. Далее администратор выбирает конкретный рейс и нажимает одну из кнопок: “Удалить”, ”Добавить”, ”Редактировать”.
5. При нажатии на кнопку “Редактировать”, открывается окно редактирования / добавления рейса, администратор вносит необходимые поправки в рейс и нажимает кнопку

“ Ок”, тем самым сохраняя изменения о рейсе.

1. При нажатии на кнопку “Удалить”, рейс удаляется.
2. При нажатии на кнопку “Добавить”, открывается окно редактирования / добавления рейса, где администратор вносит данные о рейсе и нажимает кнопку “ Ок”, тем самым сохраняя и добавляя новый рейс.
3. После любого из данных действий, администратор нажимает кнопку “Принять”, после чего открывается главное окно.

**Название прецедента:**Перейти в окно изменения пользователей

**Действующее лицо:**Администратор

**Цель:**Изменение данных о пользователях

**Предусловия:**Администратор осуществил вход в систему

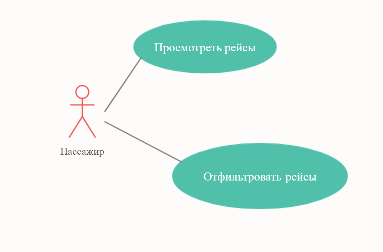
**Главная последовательность:**

1. Администратор в главном окне нажимает кнопку “Администрирование”.
2. Открывается окноадминистратора.
3. Администратор выбирает пункт “Управление пользователями”.
4. Далее администратор нажимает одну из кнопок: “Удалить”, ”Добавить”.
5. При нажатии на кнопку “Добавить”, открывается окно добавления пользователя, администратор вносит необходимые данные о пользователе и нажимает кнопку

“ Ок”, тем самым добавляя нового пользователя.

1. При нажатии на кнопку “Удалить”, нужно сначала выбрать пользователя, после чего пользователь удаляется.
2. После любого из данных действий, администратор нажимает кнопку “Принять”, после чего открывается главное окно.

Диаграмма прецедентов пассажира:



**Название прецедента:**Просмотреть рейсы

**Действующее лицо:**Пассажир

**Цель:** Посмотреть на список рейсов с целью узнать интересующую информацию

**Предусловия:**-

**Главная последовательность:**

1. Пассажир в главном окне просматривает список рейсов.

**Название прецедента:**Отфильтровать рейсы

**Действующее лицо:**Пассажир

**Цель:** Отфильтровать список рейсов по интересующему параметру.

**Предусловия:**-

**Главная последовательность:**

1. Пассажир в главном окне просматривает список рейсов.
2. В текстовое поле пассажир вводит фильтр(ы) (Время вылета, Пункт прилёта, время вылета, время прилёта, статус) и фильтрация происходит автоматически.

**7. Стадии и этапы разработки**

**7.1. Стадии разработки**

Разработка должна быть проведена в три стадии:

1. Разработка технического задания;
2. Создание приложения.
3. Создание пояснительной записке к проекту;

**7.2. Этапы разработки**

|  |  |
| --- | --- |
| **Этап разработки** | **Сроки** |
| Реализация окна, доступного пассажиру. | 15.10 – 22.10 |
| Реализация окон, доступных администратору. | 22.10 – 19.11 |
| Реализация окон, доступных диспетчеру. | 19.11 – 28.11 |
| Написание пояснительной записки. | * 1. – 3.12 |

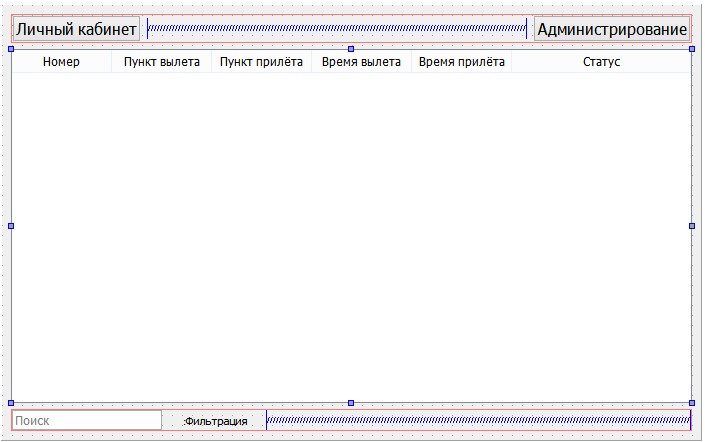
**7. Порядок контроля и приёмки**

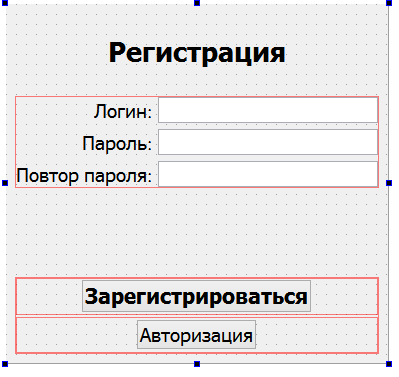
Производится проверка корректного выполнения функций программы, осуществляется функциональное тестирование программы и визуальная проверка интерфейса.

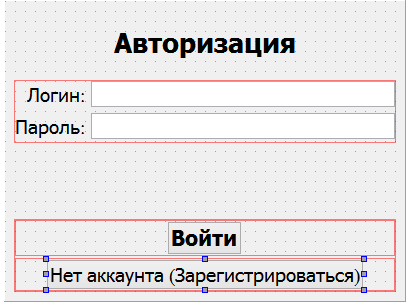
Проект будет утверждён при корректной работе программы при различных входных и выходных данных, при работе под разными пользователями, а так же при соответствии программы требования технического задания.

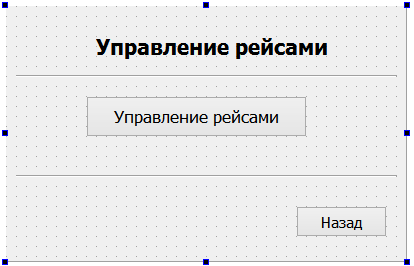
**ПРИЛОЖЕНИЕ**

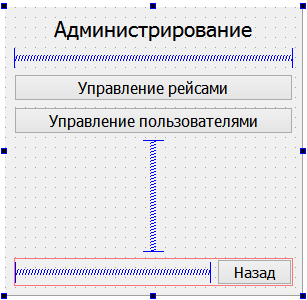
**Макет интерфейса**

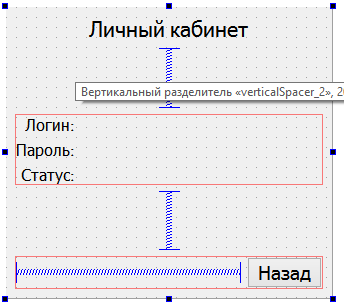
****

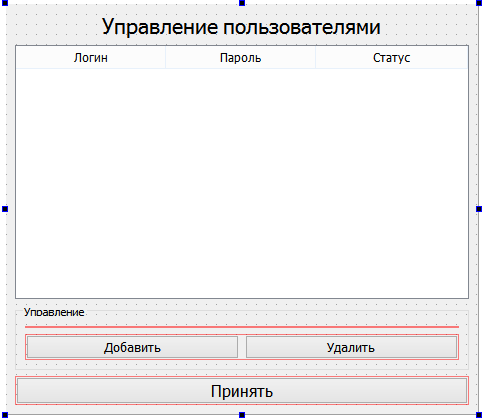
****

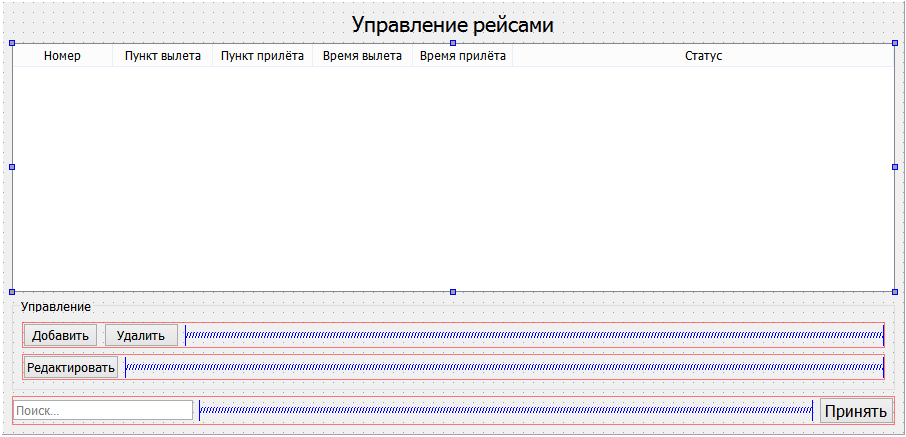
****

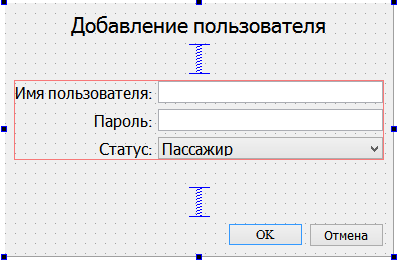
****

****

****

****

****

****

****