**Инспектор аэропортов**

Выпуск 1.0



Спецификация требований

к программному обеспечению

Версия 1.0

**Содержание**

[**История изменений 4**](#_tb6iiat9e4kl)

[**1. Введение 5**](#_6sne3yq0ghsx)

[1.1. Назначение 5](#_s6htexymc75w)

[1.2. Область применения 5](#_69cvv6wd4k77)

[1.3. Определения, акронимы и сокращения 5](#_t7x2e3917r3u)

[1.4. Ссылки 5](#_kfbxalizr6eq)

[**2. Общее описание 6**](#_deikhqqdqw9n)

[2.1. Видение продукта 6](#_1v4gou18l8na)

[2.2. Перспективы продукта 6](#_7pw3euvmwnej)

[2.3. Взаимодействие продукта с другими продуктами и компонентами 6](#_eh4j2wi97yij)

[2.4. Классы и характеристики пользователей 7](#_shalp0exaken)

[2.5. Ограничения разработки и реализации 7](#_bqz54vl81omp)

[2.6. Допущения и зависимости 7](#_3rx2rnq0sxu2)

[**3. Функциональные требования 8**](#_2hbqmaih2w86)

[3.1. Подключение к БД 8](#_7ybrwrj794te)

[3.1.1. Описание и приоритет 8](#_lr3n0oa87581)

[3.1.2. Причинно-следственные связи 8](#_i7feaiuig7ml)

[3.1.3. Функциональные требования 8](#_igeaoc4jnrfg)

[3.2. Получение начальных данных 9](#_kb2p8sk36i8o)

[3.2.1. Описание и приоритет 9](#_mj194bkw6z55)

[3.2.2. Причинно-следственные связи 9](#_9rhbjttdj7io)

[3.2.3. Функциональные требования 9](#_2857pqcbh26w)

[3.3. Получение расписания рейсов 9](#_s7u4cj5ub05m)

[3.3.1. Описание и приоритет 9](#_afxoptse6jfg)

[3.3.2. Причинно-следственные связи 9](#_xnulkufdufjb)

[3.3.3. Функциональные требования 9](#_xnuch1aulsit)

[3.4. Просмотр статистики 10](#_ln1lq7q9b031)

[3.4.1. Описание и приоритет 10](#_uw9z5fhxnyc7)

[3.4.2. Причинно-следственные связи 10](#_7tn2jwllcmzb)

[3.4.3. Функциональные требования 10](#_x466pyyxrqbf)

[3.8. Взаимодействие с внешними источниками 11](#_nshah68t02x1)

[3.8.1. Описание и приоритет 11](#_t7v34dly9yzg)

[3.8.2. Причинно-следственные связи 11](#_yzdmd7m7a8kr)

[3.8.3. Функциональные требования 11](#_msqmur7kzbx7)

[**4. Нефункциональные требования 12**](#_9uoniurcb730)

[4.1. Интерфейсы 12](#_jrnegzu7ykeh)

[4.1.1. Главная форма приложения 12](#_xdfmz1i5cksb)

[4.1.2. Форма окна загруженности аэропортов 12](#_kxy9yp752shn)

[4.1.2. Программный интерфейс 13](#_ihuu762lget6)

[4.1.3. Интерфейсы передачи данных 13](#_35ke3b1wzesf)

[4.2. Документация для пользователей 13](#_hjhricsxlele)

[4.3. Лицензионные требования 13](#_4mb8rg7p2mqu)

[4.4. Предупреждения, касающиеся законодательства, авторских прав и другие замечания 13](#_yauw7ml37wyn)

[4.5. Требования к производительности 13](#_jumf9duw5rl)

[4.6. Требования к безопасности 14](#_t2uxsnx3qg3t)

[4.7. Атрибуты качества 14](#_h7zjvc2zthwr)

[Приложение A. SQL-запросы необходимые для функционирования приложения 15](#_7pg4j1bfd019)

[Запрос списка аэропортов 15](#_tu8fhnjpdtss)

[Запрос прибывающих самолетов 15](#_qsufeze22ufc)

[Запрос вылетающих самолетов 15](#_smubb6whym7s)

[Запрос общей статистики за год 16](#_toa2sc1z8wsz)

[Запрос статистики по дням за год 16](#_7mgd36wew4c0)

[**Приложение Б. Матрица проверки требований 17**](#_2rbfhjbn5615)

## История изменений

(изменения вносятся по порядку)

|  |  |  |  |
| --- | --- | --- | --- |
| **Специалист** | **Версия** | **Дата** | **Описание** |
| Федоров Д.А. | 1.0 |  | Документ создан |

## 

## 1. Введение

### 1.1. Назначение

Эта спецификация требований к ПО описывает функциональные и нефункциональные требования к выпуску 1.0 приложения “Инспектор аэропортов”.

Этот документ предназначен для команды, которая будут реализовывать и проверять корректность работы приложения.

Кроме специально обозначенных случаев, все указанные здесь требования имеют высокий приоритет и приписаны к выпуску 1.0.

### 1.2. Область применения

Требования настоящий спецификации распространяется на все действия связанные с разработкой ПО “Инспектор аэропортов” и на все этапы разработки ПО.

Спецификация разработана в соответствии с планом разработки ПО и полностью описывает функционал MVP-версии приложения, которое будет представлено заказчику.

Приложение “Инспектор аэропортов” рассчитано на широкую аудиторию и позволяет людям без перехода на сайты аэропортов посмотреть расписание рейсов и оценить загруженность аэропорта. Тем самым данное приложение способствует сокращению времени, затрачиваемое людьми на планирование полетов.

Полная функциональность приложения перечислена в разделе документа “Конкретные требования”.

### 1.3. Определения, акронимы и сокращения

|  |  |  |
| --- | --- | --- |
| **БД** | - | База данных |
| **ВИ** | - | Требования к внешним интерфейсам |
| **О** | - | Ограничение |
| **ПИ** | - | Пользовательский интерфейс |
| **ПО** | - | Программное обеспечение |
| **ПТ** | - | Пользовательское требование |
| **ФТ** | - | Функциональное требование |
| **MVP** | - | Minimal Viable Product |

### 1.4. Ссылки

* [Описание БД](https://edu.postgrespro.ru/demo-20161013.pdf)

## 2. Общее описание

### 2.1. Видение продукта

Основным функционалом приложения “Инспектор аэропортов” является получение информации из БД, предоставленной Росавиацией, по рейсам для основных и районных аэропортов страны. Пользователям доступен список аэропортов и при вводе определенной даты пользователь может посмотреть расписание прилетов или вылетов из интересующего аэропорта. Для демонстрации возможностей приложения Росавиация предоставила БД всех совершенных полетов за период 2016-08-15 02:45:00.000 +0300 по 2017-09-14 20:55:00.000 +0300

Дополнительным функционалом приложения является возможность просмотра графиков загруженности выбранного аэропорта. Загруженность пользователям предоставляется за год в виде диаграммы, на которой отображено суммарное количество принятых и выпущенных рейсов за год, а также график прилетов/вылетов за выбранный месяц.

### 2.2. Перспективы продукта

Дальнейшие развитие приложения планируется по следующим направлениям:

* добавление возможности просмотра статистики по рейсам, загруженности конкретного рейса в зависимости от времени года;
* добавление возможности просмотра изменения стоимости билета для выбранного рейса в течении года;
* возможность бронирования билетов прямо из приложения.

### 2.3. Взаимодействие продукта с другими продуктами и компонентами

Приложение при работе взаимодействует с БД Росавиации. Для хранения данных используется БД PostgreSQL v 15.

### 2.4. Классы и характеристики пользователей

Разграничение прав пользователей в настоящей версии приложения не предусмотрено.

### 2.5. Ограничения разработки и реализации

Приложение должно быть реализовано при помощи фреймворка Qt. Для взаимодействия с БД должны использоваться классы модуля QtSQL, для отображения данных аэропортов должен использоваться модуль QtCharts или библиотека QCustomPlot.

### 2.6. Допущения и зависимости

Допущения и зависимости отсутствуют

## 3. Функциональные требования

### 3.1. Подключение к БД

#### 3.1.1. Описание и приоритет

Функционал позволяет подключаться к БД.

#### 3.1.2. Причинно-следственные связи

Требования раздела пересекаются с требованиями приведенными в разделе **3.8**

#### 3.1.3. Функциональные требования

1. При начале работы приложение должно автоматически подключаться к БД.

Данные для подключения к БД приведены в таблице 1.

Таблица 1 Данные для подключения к БД

|  |  |
| --- | --- |
| **Имя хоста** | 981757-ca08998.tmweb.ru |
| **Порт** | 5432 |
| **Имя базы данных** | demo |
| **Пользователь** | netology\_usr\_cpp |
| **Пароль** | CppNeto3 |

1. Данные для подключения к БД в MVP-версии допускается сохранить напрямую в коде приложения.
2. Приложение в нижней части окна должно отображать статус подключения к БД. Статус может иметь только два значения (Подключено/Отключено).
3. До получения списка аэропортов кнопки главной формы приложения должны быть заблокированы.
4. В случае ошибочного подключения приложение должно отображать окно “QMessageBox” и выводить текст ошибки подключения.
5. В случае ошибочного подключения, после закрытия окна с ошибкой, приложение должно повторить попытку подключения через 5 секунд. Приложение должно пытаться подключиться до установления успешного соединения.

### 

### 3.2. Получение начальных данных

#### 3.2.1. Описание и приоритет

Функционал позволяет приложению получить перечень начальных данных, необходимых для работы.

#### 3.2.2. Причинно-следственные связи

Требования раздела пересекаются с требованиями приведенными в разделе **3.8**

#### 3.2.3. Функциональные требования

1. Приложение после подключения к БД должно получать список аэропортов из БД.
2. Приложение должно отображать список аэропортов в главном окне в виде выпадающего списка.
3. Названия аэропортов в выпадающем списке должны быть на русском языке.

### 3.3. Получение расписания рейсов

#### 3.3.1. Описание и приоритет

Функционал позволяет пользователю просмотреть расписание рейсов за выбранный день для выбранного аэропорта и направления (вылет/прилет)

#### 3.3.2. Причинно-следственные связи

Требования раздела пересекаются с требованиями приведенными в разделе **3.8**

#### 3.3.3. Функциональные требования

1. Приложение должно по нажатию на кнопки в ПИ получать из БД список рейсов для выбранного аэропорта, по выбранному направлению (прилет/вылет) и за выбранную пользователем дату.
2. Если пользователь получает статистику для вылетов приложение в таблицу главной формы должно выводить следующую информацию по рейсам:

* Номер рейса
* Время вылета
* Аэропорт назначения

1. Если пользователь получает статистику для прилетов приложение в таблицу главной формы должно выводить следующую информацию по рейсам:

* Номер рейса
* Время вылета
* Аэропорт отправления

1. Пользователь должен иметь возможность выбора направления рейсов (вылет/прилет).
2. Приложение должно ограничивать возможность одновременного выбора пользователем сразу двух направлений.
3. Пользователь должен иметь возможность ввода интересующей его даты.
4. Вводимые значения даты должны быть ограничены диапазоном 2016-08-15 - 2017-09-14

### 3.4. Просмотр статистики

#### 3.4.1. Описание и приоритет

Функционал позволяет пользователям просматривать загруженность выбранного аэропорта в течении года или в течении выбранного месяца.

#### 3.4.2. Причинно-следственные связи

Требования раздела пересекаются с требованиями приведенными в разделе **3.8**

#### 3.4.3. Функциональные требования

1. Приложение должно иметь возможность получения данных по загруженности аэропорта из БД: количество вылетов/прилетов в день для выбранного аэропорта.
2. Приложение должно иметь возможность отображать загруженность для выбранного аэропорта. Загруженность должна отображаться в графическом виде в отдельном окне по нажатию пользователем на кнопку отображения загруженности.
3. При открытом окне загруженности главная форма должна блокироваться.
4. Форма отображения загруженности аэропорта должна закрываться при нажатии на кнопку “Закрыть” (или крестиком).
5. В окне загруженности должно быть две вкладки: загруженность аэропорта за год и загруженность аэропорта за выбранный месяц.
6. Во вкладке отображения загруженности за год должна отображаться столбиковая диаграмма. В каждом столбике отображается суммарное количество всех вылетов и прилетов для выбранного аэропорта за каждый месяц.
7. Во вкладе отображения загруженности за месяц, должен быть выпадающий список с месяцами начинающимися с января, заканчивающимся декабрем.
8. Во вкладке отображается линейный график, по оси Х числа месяца(начиная с 1), по оси У суммарное количество прилетов/вылетов в выбранном аэропорте за этот день.
9. При открытии вкладки должен быть установлен месяц “Январь” и выведен график.
10. При изменении месяца в списке график должен автоматически перестраиваться.
11. Пользователь должен иметь возможность выбора месяца из выпадающего списка

### 3.8. Взаимодействие с внешними источниками

#### 3.8.1. Описание и приоритет

Функционал позволяет приложению взаимодействовать с БД установленной на сервере, в которой хранится информация обо всех 2016-08-15 02:45:00.000 +0300 по 2017-09-14 20:55:00.000 +0300

#### 3.8.2. Причинно-следственные связи

Осуществляется взаимодействие с функционалом описанным в разделах **3.1; 3.2; 3.3; 3.4**

#### 3.8.3. Функциональные требования

1. Приложение должно обеспечивать взаимодействие с БД Postgresql v15.

## 4. Нефункциональные требования

### 4.1. Интерфейсы

#### 4.1.1. Главная форма приложения

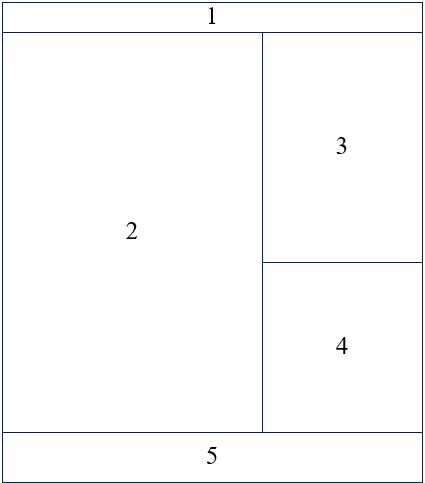


Рисунок 1. Общий вид главного окна приложения

#### 4.1.2. Форма окна загруженности аэропортов

#### 

Рисунок 2. Общий вид формы отображения загруженности

1. Главная форма приложения должна быть разделена на зоны:
2. Панель меню
3. Таблица отображения рейсов
4. Зона выбора аэропорта и направления вылетов
5. Кнопки управления
6. Зона отображения статуса подключения к БД
7. Окно загруженности аэропортов должно быть разделено на зоны:
8. Информационная зона с отображением названия аэропорта для которого просматривается загруженность
9. Зона построения графиков
10. Кнопка выхода из окна
11. Требования пункта **ПИ1** и **ПИ2** являются рекомендательными, разработчик приложения вправе вносить изменения в компоновку главной формы.
12. Формы должны быть эргономичны, иметь дружественный вид. Требование будет оцениваться заказчиком субъективно.

#### 4.1.2. Программный интерфейс

Специфические требования для программных интерфейсов отсутствуют. Перечень команд SQL необходимых для взаимодействия с БД приведен в **приложении A**

#### 4.1.3. Интерфейсы передачи данных

Специфические требования к интерфейсам передачи данных отсутствуют.

### 4.2. Документация для пользователей

Для данной версии приложения документация для пользователей не предусмотрена.

### 4.3. Лицензионные требования

1. Все права на разрабатываемое приложения принадлежат Росавиации
2. Приложение распространяется абсолютно бесплатно

### 4.4. Предупреждения, касающиеся законодательства, авторских прав и другие замечания

Вся информация представленная в БД является конфиденциальной и охраняется ФЗ №152 “О персональных данных”.

### 4.5. Требования к производительности

1. Приложение должно загружаться на устройствах менее чем за 10 сек.

### 4.6. Требования к безопасности

1. Предоставленные пользователями данные должны храниться и обрабатываться в соответствии с ФЗ №152 “О персональных данных”.

### 4.7. Атрибуты качества

1. Приложение должно разрабатываться как кросс-платформенное. Должно функционировать на Windows, Linux и MacOS.

# Приложение A. SQL-запросы необходимые для функционирования приложения

В приложении приведены запросы к БД, необходимые для получения набора данных в соответствии с настоящими требованиями.

В приложении приведены единичные запросы. При реализации приложения некоторые переменные запросов будут динамическими, это необходимо учитывать в коде.

## Запрос списка аэропортов

Запрос возвращает список аэропортов и их коды в системе ICAO

SELECT airport\_name->>'ru' as "airportName", airport\_code FROM bookings.airports\_data

## Запрос прибывающих самолетов

Запрос возвращает данные по прибывающим рейсам.

SELECT flight\_no, scheduled\_arrival, ad.airport\_name->>'ru' as "Name" from bookings.flights f

JOIN bookings.airports\_data ad on ad.airport\_code = f.departure\_airport

where f.arrival\_airport = ‘**airportCode’**

**airportCode -** код аэропорта отправления

## Запрос вылетающих самолетов

Запрос возвращает данные по прибывающим рейсам.

SELECT flight\_no, scheduled\_departure, ad.airport\_name->>'ru' as "Name" from bookings.flights f

JOIN bookings.airports\_data ad on ad.airport\_code = f.arrival\_airport

WHERE f.departure\_airport = ‘**airportCode’**

**airportCode -** код аэропорта отправления

## Запрос общей статистики за год

Запрос возвращает количество рейсов для каждого месяца совершенных в период с 2016-08-31 по 2017-08-31, для выбранного аэропорта.

SELECT count(flight\_no), date\_trunc('month', scheduled\_departure) as "Month" from bookings.flights f

WHERE (scheduled\_departure::date > date('2016-08-31') and scheduled\_departure::date <= date('2017-08-31')) and ( departure\_airport = **airportCode** or arrival\_airport = airportCode )

group by "Month”

**airportCode -** код аэропорта отправления

## Запрос статистики по дням за год

Запрос возвращает количество рейсов для выбранного аэропорта за каждый день, в интервале

SELECT count(flight\_no), date\_trunc('day', scheduled\_departure) as "Day" from bookings.flights f

WHERE(scheduled\_departure::date > date('2016-08-31') and scheduled\_departure::date <= date('2017-08-31')) and ( departure\_airport = **airportCode** or arrival\_airport = **airportCode**)

GROUP BY "Day"

**airportCode -** код аэропорта отправления

# Приложение Б. Матрица проверки требований

Таблица 2. Список выполненных проверок

|  |  |  |
| --- | --- | --- |
| **Идентификатор требования** | **Результат проверки** | **Наименование функций, реализующих это требование** |
| Функциональные требования | | |
| ФТ1 | ОК | void DatabaseConnection::connectToDatabase() |
| ФТ2 | ОК | void DatabaseConnection::setConnectionData() |
| ФТ3 | ОК | void AirportInspector::databaseConnectionStatusDay(bool status) |
| ФТ4 | ОК | void AirportInspector::printListAirports(QVector<QString> \_airport\_code, QVector<QString> \_list\_airports) |
| ФТ5 | ОК | void AirportInspector::databaseConnectionError(DatabaseConnection& database) |
| ФТ6 | ОК | void AirportInspector::databaseConnectionError(DatabaseConnection& database) |
| ФТ7 | ОК | void AirportInspector::printListAirports(QVector<QString> \_airport\_code, QVector<QString> \_list\_airports) |
| ФТ8 | ОК | void AirportInspector::printListAirports(QVector<QString> \_airport\_code, QVector<QString> \_list\_airports) |
| ФТ9 | ОК | void AirportInspector::printListAirports(QVector<QString> \_airport\_code, QVector<QString> \_list\_airports) |
| ФТ10 | ОК | void AirportInspector::Scoreboard(QVector<QVector<QVector<QString>>> scoreboard) |
| ФТ11 | ОК | void AirportInspector::Scoreboard(QVector<QVector<QVector<QString>>> scoreboard) |
| ФТ12 | ОК | void AirportInspector::Scoreboard(QVector<QVector<QVector<QString>>> scoreboard) |
| ФТ13 | ОК | void AirportInspector::on\_lw\_Airport\_clicked(const QModelIndex &index) |
| ФТ14 | ОК | void AirportInspector::on\_pb\_AirportCongestion\_clicked() |
| ФТ15 | ОК | void AirportInspector::on\_pb\_AirportCongestion\_clicked() |
| ФТ16 | ОК | void WorkloadSchedule::on\_pb\_Close\_clicked() |
| ФТ17 | ОК | Tab, tab\_2 |
| ФТ18 | ОК | void WorkloadSchedule::printYear(QVector<QDate> workload) |
| ФТ19 | ОК | void WorkloadSchedule::printmonth(qint32 month\_print) |
| ФТ20 | ОК | void WorkloadSchedule::printmonth(qint32 month\_print) |
| ФТ21 | ОК | void WorkloadSchedule::printmonth(qint32 month\_print) |
|
| ФТ22 | ОК | void WorkloadSchedule::printmonth(qint32 month\_print) |
| Пользовательские требования | | |
| ПТ1 | ОК | tb\_Arrival, tb\_Departure |
| ПТ2 | ОК | void AirportInspector::on\_calendarWidget\_clicked(const QDate &date) |
| ПТ3 | ОК | void WorkloadSchedule::on\_lw\_MonthUp\_clicked(const QModelIndex &index)  void WorkloadSchedule::on\_lw\_MonthDown\_clicked(const QModelIndex &index) |
| Ограничения | |  |
| О1 | ОК | void AirportInspector::Scoreboard(QVector<QVector<QVector<QString>>> scoreboard) |
| О2 | ОК | void AirportInspector::maxMinDate(QVector<QDate> max\_min) |
| Требования к внешним интерфейсам | | |
| ВИ1 | ОК | class DatabaseConnection |
| Требования к пользовательскому интерфейсу | | |
| ПИ1 | ОК | Airportinspector.ui |
| ПИ2 | ОК | Workloadschedule.ui |
| ПИ3 | ОК |  |
| ПИ4 | ОК |  |
| Требования к производительности | | |
| ТП1 | ОК |  |
| Требования к безопасности | | |
| ТБ2 | ОК |  |
| Требования к атрибутам качества | | |
| АК1 | ОК |  |