

SRS - Software Requirements Specification: Flight Route Generator

Doc. number: SAAB180917

Version: 1.0

Marlon Abeln, Agnes Andersson, Felix Danielsson, Sofie Hallborn, Majlinda Haziri, Mikael Nyström, and Robin Ta

Contents

1	Introduction	2
1.1	Purpose of the document	2
1.2	Intended Users	2
1.3	Project Scope	2
2	Reference documents	2
3	Terminology	2
4	Description	3
4.1	System Objective	3
4.2	System Features	3
4.3	Assumptions and Dependencies	3
4.4	Requirement enumeration	3
5	Background	3
5.1	Actors and their objectives	3
6	Context diagram	4
7	Functional Requirements	4
7.1	Platform	4
7.2	Software	4
7.3	Standards	4
7.4	Sustainability	5
7.5	User Interface	5
7.6	Database	6
7.7	Data	7
8	Quality Requirements	9
8.1	Performance	9
9	Project Requirements	9
9.1	Delivery	9
10	Appendix	10
A	Reference Documents	10
B	Data Models	12

1 Introduction

This section states the goal of the project and the intended users.

1.1 Purpose of the document

The purpose of this document is to describe the Flight Route Generator, a system that is developed as a project for Saab Kockums AB, and its requirements.

1.2 Intended Users

The system is intended for Saab Kockums AB.

1.3 Project Scope

The system, as explained in section 4.1, is developed with the goal to simplify the task of importing Standard Terminal Arrival Routes (STAR) and Standard Instrument Departure (SID) routes into a format that is usable by the current simulation and training software developed by Saab Kockums AB.

2 Reference documents

Listed below are the documents referenced in this document.

1. *Import av information flygrummet och rutter* (Appendix B) in Swedish
2. *Automatiskt genererade flygrutter, Studentprojekt 2018*, Saab Kockums, Version 1.0 in Swedish

3 Terminology

data source	is a source of information such as a database from which information can be queried. See Reference 2, Appendix A.
waypoint	is defined as an intermediate point on a route or a point where a course is changed.
importer	provides the functionality of importing data from data source .
exporter	provides the functionality of converting data to the XML format.
LFV	is the acronym for "Luftfartsverket", which is a Swedish agency for air traffic.
WFS	is the acronym for "Web Feature Service". This is a web-service that is used to handle geographic data.

4 Description

This section contains the description of the system.

4.1 System Objective

The system's objective is to import data from **data source** sources such as the **waypoints** around a given airport, as well as generating STAR and SID routes. It will then allow for data to be stored in an external database and be exported in XML-format through an interface.

4.2 System Features

1. Provide an interface where it is possible to choose an airport.
2. Import data from **LFV's WFS**.
3. Provide an interface that allows selecting **waypoints** into a route.
4. Convert the imported data to be stored in the database.
5. Export data from the database to XML-format, in line with Data Model 1 Appendix B.

4.3 Assumptions and Dependencies

Assumptions:

1. The data received from **LFV's WFS** server is correct.

Dependencies:

1. **LFV's WFS** server.

4.4 Requirement enumeration

The requirements in this document start in section 6. The enumeration of the requirements contains the section number, subsection number, and a unique number for the respective requirement.

5 Background

This section describes the background for the system.

5.1 Actors and their objectives

The system will only be used by one *user* at a time for each computer the system is installed on.

User: A user can search, create, view, and export routes.

6 Context diagram

Requirement 6.0.1. The system shall allow interactions as shown in figure 1.

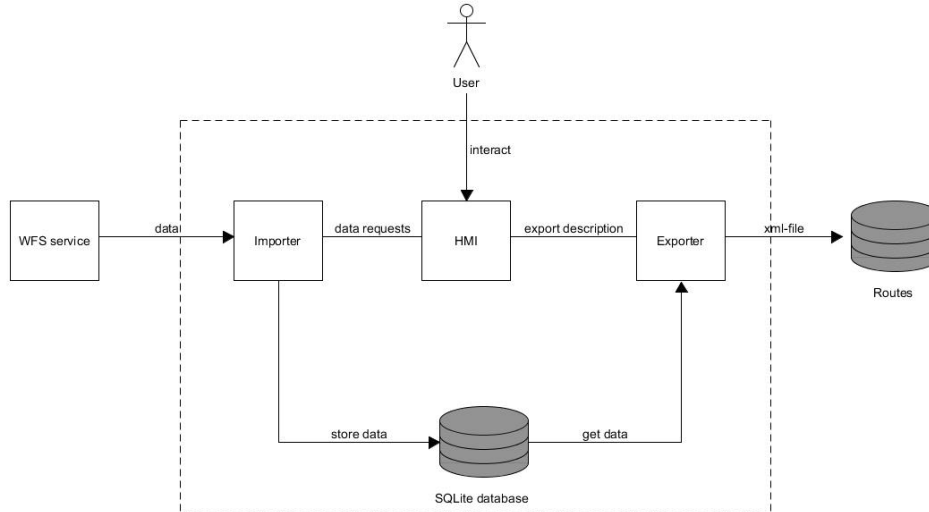


Figure 1: Context Diagram of the system.

7 Functional Requirements

Listed below are the functional requirements that the system shall fulfill.

7.1 Platform

Requirement 7.1.1. The system shall be developed to run on Microsoft Windows 10.

7.2 Software

Requirement 7.2.1. The system source code shall be created using the C# language version 7.2.

Requirement 7.2.2. The system shall be developed on the .NET framework version 4.6.0.

Requirement 7.2.3. The system shall be developed using Visual Studio 2017 version 15.8.4.

Requirement 7.2.4. The system shall be using SQLite version 3.25.2 when storing, updating and deleting in the database.

7.3 Standards

Requirement 7.3.1. Produced documents (which includes the manual for the system, Software requirement Specification and Final report) shall be written in English.

Requirement 7.3.2. The system shall produce an output XML-file.

Requirement 7.3.3. The system shall read a GeoJSON object from **LFV's WFS** server.

Requirement 7.3.4. The system shall not use libraries that are published under any of the following licenses:

- (a) GPLv1
- (b) GPLv2
- (c) GPLv3

7.4 Sustainability

Requirement 7.4.1. The system shall only be dependent on **data sources**, as it's external factors.

Requirement 7.4.2. The source code shall contain comments before important code lines, which describes what the code does.

7.5 User Interface

Requirement 7.5.1. The user interface shall include a search field.

Requirement 7.5.2. The user interface shall include a function where the user chooses among the **waypoints** for the chosen airport to create a route.

Scenario 7.5.3. User search for non-existent airport code:

1. The user accesses the system.
2. The user is asked to provide an airport code in the search field.
3. The user provides a non-existent airport code.
4. A message is displayed with information that the user has provided an incorrect airport code.

Requirement 7.5.3. Scenario 7.5.3 shall be supported by the system.

Requirement 7.5.4. The user interface shall display the results of the operation that the user chooses.

Requirement 7.5.5. The route creating page in the user interface shall look like Figure 2.

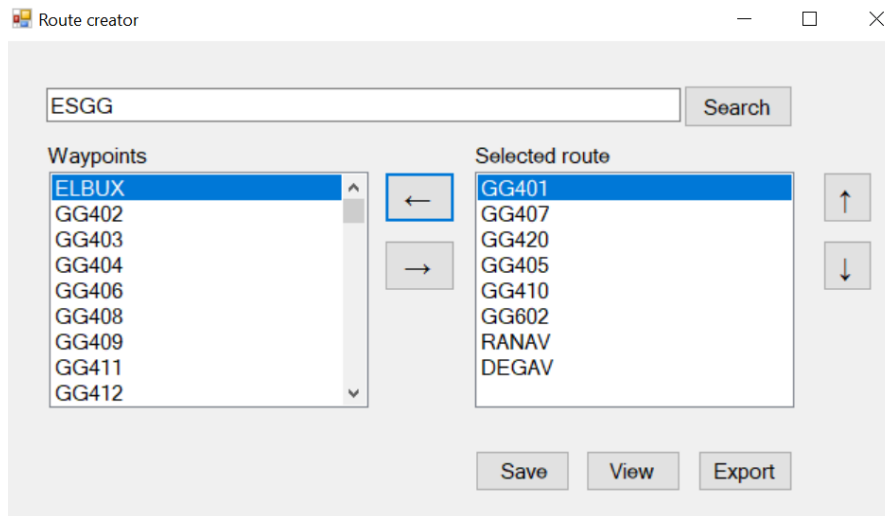


Figure 2: Mock up of the user interface.

Scenario 7.5.6. A user adds a route in the user interface:

1. The user types in an airport name in the search field.
2. A list of **waypoints** for the specified airport appears in the left column.
3. The user selects a **waypoint**.
4. The user presses the right arrow to move the selected **waypoint** to the right column.
5. Repeat step 3 and 4 until all desired **waypoints** are added to the route.
6. The user selects a **waypoint** in the right column.
7. The user presses the down arrow to change the order of a **waypoint**.
8. The user presses "Save".
9. A text field appears.
10. The user enters a name for the route.
11. The user selects STAR.
12. The user presses "Confirm".

Requirement 7.5.6. Scenario 7.5.6 shall be supported by the system.

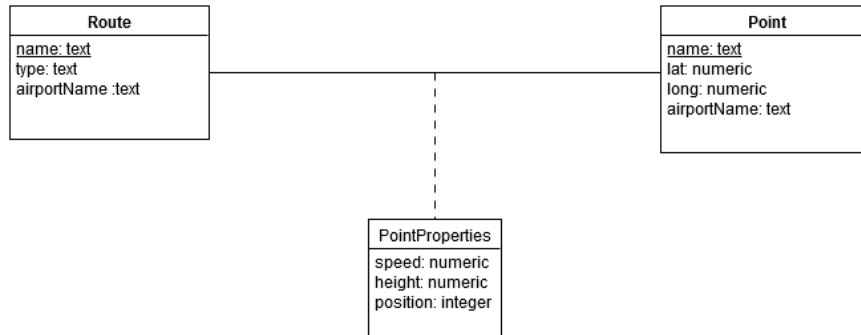
7.6 Database

Requirement 7.6.1. The system shall use an SQLite database.

Requirement 7.6.2. The database shall be stored on the users workstation.

Requirement 7.6.3. The database shall be used as shown in Figure 3.

Figure 3: UML diagram of the database.

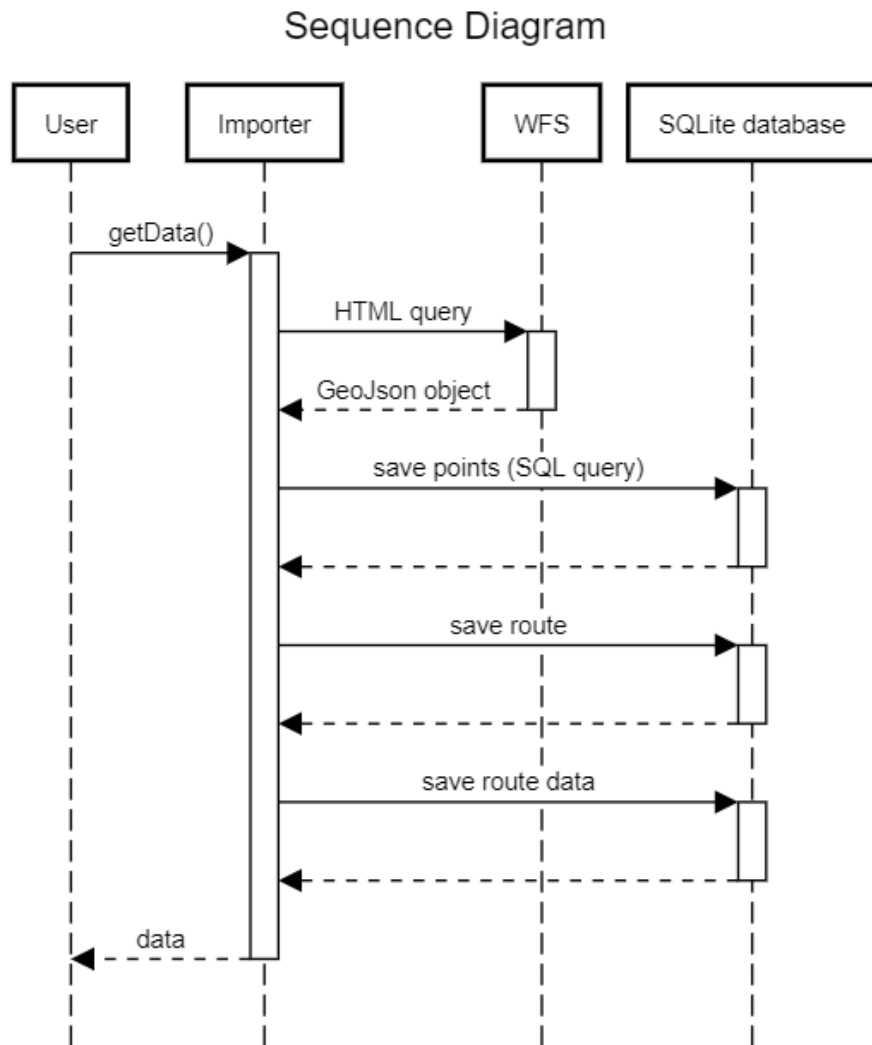


7.7 Data

Requirement 7.7.1. The user shall only be able to enter an airport code at least 1 character long, ASCII (decimal) values 48-57, 65-90, and 97-122 allowed.

Requirement 7.7.2. The downloaded data shall be in GeoJSON format.

Requirement 7.7.3. The download of data shall be done as displayed in Figure 4.

Figure 4: Sequence diagram of the query to a **data source**

Requirement 7.7.4. The processed data shall be stored in the database.

Requirement 7.7.5. The stored data shall be converted to an XML file according to the skeleton in Data Model 1 Appendix B when accessed by the **system**.

Requirement 7.7.6. The coordinates shall be stored in radians.

Requirement 7.7.7. The coordinates shall be converted to radians when imported from the WFS.

Requirement 7.7.8. The coordinate shall be accurate to at least the 6th decimal.

Requirement 7.7.9. In the output XML file, see Data Model 1 Appendix B, the following attributes shall be set to 0 :

- (a) distance
- (b) angle
- (c) type

Requirement 7.7.10. The section element in the XML file, see Data Model 1 Appendix B, shall be increased by 1 for every new section element.

Requirement 7.7.11. The startEvents element in the XML file, see Data Model 1 Appendix B, shall be empty.

Requirement 7.7.12. The events element in the XML file, see Data Model 1 Appendix B, shall be empty.

8 Quality Requirements

Listed below are the requirements specified for the quality of the system.

8.1 Performance

No performance requirements specified.

9 Project Requirements

The following requirements apply to every part of the project.

9.1 Delivery

Requirement 9.1.1. The system shall be ready for delivery by December 7th, 2018.

Requirement 9.1.2. Support for the system shall be offered until December 20th, 2018.

Requirement 9.1.3. The delivery shall include:

- (a) The source code of the system
- (b) User Manual: Flight Route Generator version 1.0
- (c) SRS - Software Requirement Specification: Flight Route Generator version 1.0

Requirement 9.1.4. The delivery shall be performed via GitHub under an MIT-license.

10 Appendix

A Reference Documents

Import av information flygrummet och rutter

Saab Training & Simulation bygger system för att utbilda flygledare. I detta system simuleras markfordon och flygplan. Flygledarna kan observera dessa på radar och genom en 3D utblick motsvarande vad de ser från tornet. Flygplanen styrs av psedopiloter men mycket av flygningen följer standardiserade rutter t ex, SID (Standard Instrument Departure) och STAR (Standard Arrival Route). Mellan flygplatserna finns det även ett antal rutter som kan användas vid planering och flygning. Dessa rutter finns publicerade i AIP (Aeronautical Information Publication).

När systemet ska anpassas till en ny flygplats lägger man manuellt alla rutter som leder in till flygplatsen. Detta är ett tidsödande arbete som vi vill kunna automatisera. Informationen för rutterna finns publicerad i AIP och är möjligt att importera.


Projektet går ut på att undersöka möjligheten till detta och hur data från AIP kan omvandlas för att användas i simuleringen. Även olika alternativ för att hämta AIP-data behöver beaktas. Efter detta så tas en prototyp fram där data från AIP importeras och omvandlas till formatet som används för scenarion i simuleringen.



Reference 1 - Import av information flygrummet och rutter.

Studentprojekt 2018

Automatiskt genererade flygrutter



COMPANY UNCLASSIFIED | NOT EXPORT CONTROLLED | NOT CLASSIFIED
Saab Kockums | Studentprojekt 2018 | Version 1.0

SAAB

Formalia

- **Säkerhetsfrågor**
 - Besöksbrickor
 - Fotoförbud
- **Projektspecifikt**
 - Ingen platsmöjlighet
 - Kontaktpersoner
 - kristian.hasselqvist@saabgroup.com
 - pierre.aupeix@saabgroup.com

COMPANY UNCLASSIFIED | NOT EXPORT CONTROLLED | NOT CLASSIFIED
Saab Kockums | Studentprojekt 2018 | Version 1.0

SAAB

Upplägg

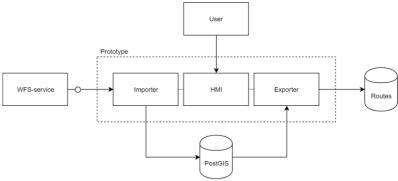
Del 1	Del 2
Utvinnning av data	Transformering av data
<ul style="list-style-type: none"> • Hitta en datakälla • Skapa ett dataset <ul style="list-style-type: none"> • Waypoints • Rutter 	<ul style="list-style-type: none"> • Omvandla data till simuleringsrutter <ul style="list-style-type: none"> • XML • Events • Koordinatsystem

Prototyp

COMPANY UNCLASSIFIED | NOT EXPORT CONTROLLED | NOT CLASSIFIED
Saab Kockums | Studentprojekt 2018 | Version 1.0

SAAB

Överblick



```

graph TD
    User --> HTML
    WFS-service --> Importer
    Importer --> HTML
    HTML --> Exporter
    Exporter --> Routes
    Importer <--> PostGIS
    Exporter <--> PostGIS
    subgraph Prototype
        Importer
        HTML
        Exporter
    end
  
```

COMPANY UNCLASSIFIED | NOT EXPORT CONTROLLED | NOT CLASSIFIED
Saab Kockums | Studentprojekt 2018 | Version 1.0

SAAB

Resurser

- **WFS-service**
 - en.wikipedia.org/wiki/Web_Feature_Service
 - daim.lfv.se/geoserver/mais/wfs
 - daim.lfv.se/echarts/
- **AIP-data**
 - aro.lfv.se/Editorial/View/IAIP?folderId=19
- **QGis**
 - <https://qgis.org/>

COMPANY UNCLASSIFIED | NOT EXPORT CONTROLLED | NOT CLASSIFIED
Saab Kockums | Studentprojekt 2018 | Version 1.0

SAAB

Reference 2 - Automatiskt genererade flygrutter, Studentprojekt 2018, Saab Kockums, Version 1.0.

B Data Models

```
<?xml version="1.0"?>
- <routeData>
  <startEvents/>
  - <breakpoints>
    - <breakpoint fix="" lat="1.003243" lon="0.286839" distance="0" angle="0" section="0" type="0">
      <events/>
    </breakpoint>
    - <breakpoint fix="" lat="1.00309" lon="0.281303" distance="0" angle="0" section="1" type="0">
      <events/>
    </breakpoint>
    - <breakpoint fix="" lat="1.003539" lon="0.281853" distance="0" angle="0" section="2" type="0">
      <events/>
    </breakpoint>
  </breakpoints>
</routeData>
```

Data Model 1 - XML skeleton for the save file