

## Purpose

The results of the requirements elicitation and the analysis activities are documented in the Requirements Analysis Document (RAD). This document completely describes the system in terms of functional and nonfunctional requirements and serves as a contractual basis between the client and the developers.

## Audience

The audience for the RAD includes the client, the end users, the project manager, and the developers.

## Table of Contents

1.	Introduction .....	3
1.1	Purpose of the system .....	3
1.2	Scope of the system .....	3
1.3	Objectives and success criteria of the project.....	3
2.	Current system .....	3
3.	Proposed system.....	3
3.1	Overview .....	3
3.2	Functional requirements .....	3
3.3	Nonfunctional requirements .....	4
3.3.1	Usability.....	4
3.3.2	Reliability .....	4
3.3.3	Performance .....	4
3.3.4	Supportability .....	5
3.3.5	Implementation Requirements .....	5
3.3.6	Interface Requirements .....	5
3.3.7	Packaging Requirements .....	5
3.3.8	Legal Requirements .....	5
3.4	System models.....	5
3.4.1	Scenarios.....	5
3.4.1.1	.....	5
3.4.1.2	.....	6
3.4.1.3	.....	6
3.4.2	Use case model .....	7
3.4.3	Object model .....	7

## Document History

Rev.	Author	Date	Changes
	Benedikt Kellner	18.07.2022	Document finished

## 1. Introduction

### 1.1 Purpose of the system

The purpose of the system is to create an application which makes travelling in airplanes more pleasant.

### 1.2 Scope of the system

The people who will use the system are the passengers of the flights.

### 1.3 Objectives and success criteria of the project

In the end, the system should run on any device, and give the user all information he needs about his flights and his destinations.

## 2. Current system

The project is a greenfield project, so we are starting from scratch.

## 3. Proposed system

*The third section documents the requirements elicitation and the analysis model of the new system.*

### 3.1 Overview

*The system should give the user the possibility to get all necessary flight information about his booked flights, as well as it should inform the user about the weather and interesting things to do in the different cities. But the system is not just a tool to plan the upcoming trips: It should also make the current flight more comfortable and safer.*

### 3.2 Functional requirements

- **FR1:** Show flight information: The system shows flight information (flight number, start time, end time, gate, terminal, seat, airplane type, airline, etc.) of all the user's flights and notifies the user if a flight is canceled or delayed.
- **FR2:** Add new flight trip: The user can search through available flights and can select the one's he has booked. The user can build flight journeys (trips) with multiple connecting flights or one return flight and can see the trips on a map.
- **FR3:** Display destination information: The user can display important information such as POIs in the destination place (e.g., hotels, restaurants,

attractions) and weather data. POIs should be shown in a map and the user should be able to filter them. When clicking on a POI, the user can see additional information and save the POI in the favorites.

- **FR4:** Give feedback: A passenger can give feedback after he took the flight about the flight itself, catering, entertainment, service or comfort. Passengers who take the survey will be rewarded, e.g., by miles, souvenirs, coupons, or price drawings.
- **FR5:** Request service: During a flight, a passenger can request the service. This notifies the service staff so that they can come to the passenger's seat.
- **FR6:** Watch flight safety instructions: The user can watch the flight safety instructions.

### 3.3 Nonfunctional requirements

- **NFR1:** Usability: The system should be intuitive to use, and the user interface should be easy to understand. All interactions should be completed in less than three clicks.
- **NFR2:** Conformance to guidelines: The design of the system should conform to the typical usability guidelines such as Nielsen's usability heuristics.
- **NFR3:** Server system: A server subsystem with a couple of services must be used in the system. However, additional services like destination information for weather and POIs should be obtained from external services.

#### 3.3.1 Usability

- The users should be able to search for new flights directly on the start page.
- Viewing your selected flights should be possible with one click.
- On the main page there should not be more than 5 buttons.

#### 3.3.2 Reliability

- The system should be runnable all the time.
- All exceptions should be caught.
- The system should just show nothing if a flight which doesn't exist is entered in the search bar.
- The user should not be able to upload files, in order to protect the system from attacks.

#### 3.3.3 Performance

- Loading new flights should not take longer than 1 second

- The compilation time should not take longer than 5 seconds
- The resolution of all images should be over 1024 pixel
- The system should always be available if the user wants to use it
- The user should be able to use the application from multiple devices at the same time

#### 3.3.4 Supportability

- The system should also be able to support new search requests.
- The system should be a REST API, so no changes on the user interface should not influence the server side

#### 3.3.5 Implementation Requirements

- Spring boot should be used for the server side
- The client side can be freely chosen, it could be implemented in JavaFX, React, ...

#### 3.3.6 Interface Requirements

- The system should be able to get data from a weather api

#### 3.3.7 Packaging Requirements

- The software should be delivered via artemis

#### 3.3.8 Legal Requirements

### 3.4 System models

*The System models include scenarios, use cases, object model, and dynamic models for the system. This section should contain the complete functional specification, including mock-ups, paper-based prototypes or storyboards illustrating the user interface of the system and navigational paths representing the sequence of screens.*

#### 3.4.1 Scenarios

##### 3.4.1.1

- 1) Name: Trip Dashboard
- 2) Participating actors: Julia: Passenger
- 3) Flow of events
  1. Julia bought a Lufthansa flight from Munich to Lisbon on June 5 via the Lufthansa website.
  2. She opens the system and searches all the available flights between Munich and Lisbon on June 5
  3. The flights are shown on the display
  4. She selects the flight and saves it to her flight list.

5. The flight is added to her flight list.
6. When looking into the details
  7. The weather of Lisbon is displayed
8. she recognizes that the sun is shining in Lisbon
9. When she clicks on Lisbon
  10. City map gets displayed
11. she sees the city map with points of interest (POI) and uses a filter to see the 10 best attractions
  12. The top 10 attractions get displayed
13. Julia decides to visit the Belém Tower and saves it in her favorites POI list
  14. The favorite POI list is added by the Belem tower

#### 3.4.1.2

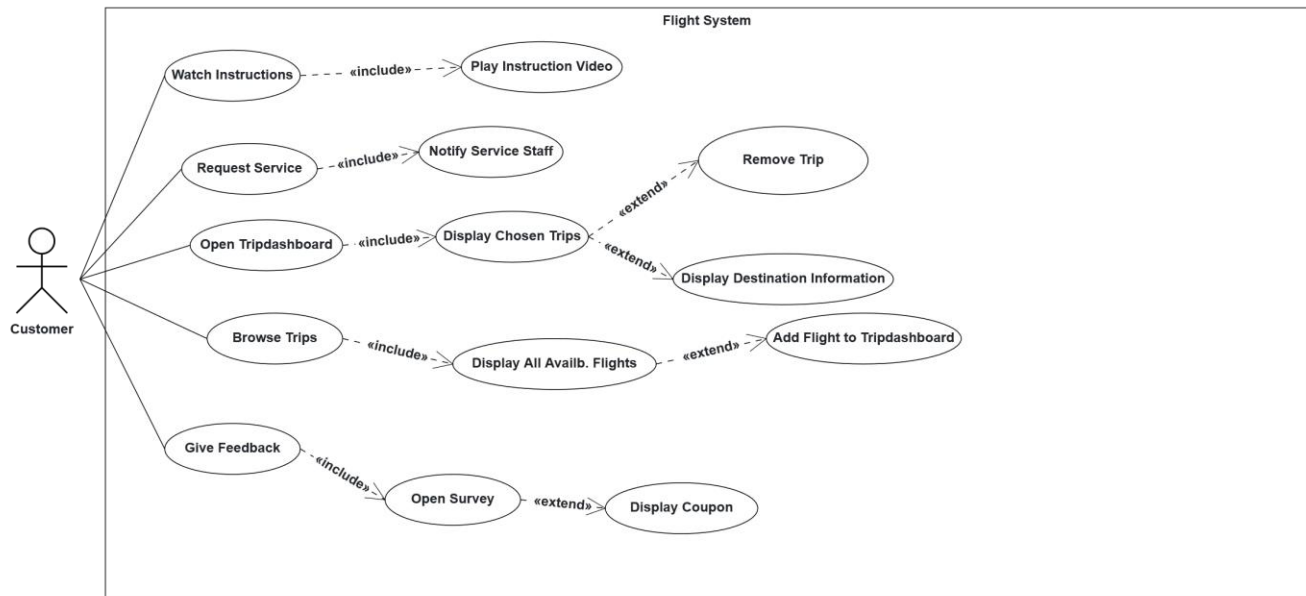
- 1) Name: Passenger Survey
- 2) Participating actors: Simon: Passenger
- 3) Flow of events
  1. Simon is currently flying from Rome to London. During the flight, he has a nice dinner with spaghetti and a beer.
  2. Then he participates in a survey
  3. Display survey
  4. Simon is asked to rate the entertainment system and the catering. He gives both a five-star rating. Then he is asked about the comfort. Simon chooses a four-star rating, because the toilet is not clean. He describes the problem in the comment section. After he quits the survey
  5. The survey is stored
  6. Grant coupon to Simon

#### 3.4.1.3

- 1) Name: In-Flight Service
- 2) Participating actors: Maria: Passenger
- 3) Flow of events
  1. Maria is flying from Barcelona to Paris with Lufthansa. Because the steward had a strong German accent, she did not understand the safety instructions before the plane takes off.
  2. She opens the system
  3. Display instructions
  4. and sees instructions on how to fasten the seatbelt, how to use the oxygen mask and how to locate the nearest emergency exit
  5. After the plane reaches cruising altitude, she decides to watch a movie using the airplane entertainment system.
  6. Display available movies and the menu

7. While she is watching the movie
8. Play movie
9. she requests a drink available on the menu.
10. Store drink request

### 3.4.2 Use case model



### 3.4.3 Object model

