



Problem Statement Flight System

1. The Problem

Our company has close relations with the world's leading aircraft manufacturers. We want to introduce a system that can be used in the airplane to make traveling more pleasant.

2. Scenarios

The following is a collection of possible scenarios that must be addressed in the project.

2.1. Trip Dashboard

Julia bought a Lufthansa flight from Munich to Lisbon on June 5 via the Lufthansa website. She opens the system and searches all the available flights between Munich and Lisbon on June 5. She selects the flight and saves it to her flight list. When looking into the details, she recognizes that the sun is shining in Lisbon. When she clicks on Lisbon, she sees the city map with points of interest (POI) and uses a filter to see the 10 best attractions. Julia decides to visit the Belém Tower and saves it in her favorites POI list.

2.2. Passenger Survey

Simon is currently flying from Rome to London. During the flight, he has a nice dinner with spaghetti and a beer. Then he participates in a survey: 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, the system shows him a coupon for a free drink at the Burger restaurant at Heathrow airport.

2.3. In-Flight Service

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. She opens the system and sees instructions on how to fasten the seatbelt, how to use the oxygen mask and how to locate the nearest emergency exit. After the plane reaches cruising altitude, she decides to watch a movie using the airplane entertainment system. While she is watching the movie, she requests a drink available on the menu.

3. 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.

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.

Additional constraints:

- The version control system must be git.
- Source code documentation must be in HTML format.
- The server system must use the Spring Boot framework.

4. Target Environment

The system should run on all desktop operation systems (Windows, macOS, Unix), either as Java or as browser-based application which communicates with the Spring Boot server application.

5. Deliverables

- Requirements analysis document (RAD)
- System design document (SDD)
- Source code under version control including source code documentation

6. Client Acceptance Criteria

The system must demonstrate at least the following functionality: It shows a list of the user's flights and let the user see details of his flight. The system displays the weather of the destination city, and the user can see his/her trip on a map. The application communicates with the server system and conforms to the usability requirements.



Problem Statement Lecture System

1. The Problem

Students want to participate in lectures and see announcements. They want to find fellow students with the same interests and share their opinion about lectures and course material. They also want to discuss exam questions and find the place where the exam takes place.

2. Scenarios

Arjun, a student from India, is studying computer science at TUM. He has business administration as minor subject. He is already used to visit the lectures in the FMI building in Garching from his first two semesters. The business administration lectures however, located in a lecture hall in the TUM city campus in Arcistrasse. He never visited the city campus before, so he does not know how to find the lecture halls for his minor subject. He browses through the lectures in the lecture catalog and finds the lecture "Foundations of Business Administration" with lecture times and the location of the lecture hall on a map. While he is attending the lecture, he makes contact with fellow students who also attend the lecture and checks their comments. He likes one comment "Great exercises" by Jenny, who is also studying informatics. From Jenny's picture, he remembers that they met a week ago at the coffee machine in Garching. He requests friendship with Jenny (she might help him to pass the final) and adds a new comment about exam questions from earlier exams. While he is browsing, Jenny is notified about the friend request and accepts it. Arjun, in turn, is notified that Jenny has accepted his request and now browses through all the lectures that Jenny is visiting. This way he finds another interesting lecture "Cost Accounting" that he visits and saves it into his lecture list.

3. Requirements

The following functional requirements (FR) and nonfunctional requirements (NFR) must be addressed in the project.

FR1: Search for available lectures: A student can see all lectures of the current semester in his major and minor subject. He is able to join the lecture which saves it into his lecture list. He can also drop a lecture.

FR2: Check lecture details: A student can see details about a lecture such as the lecture times, the location of the lecture hall on a map and other lecture attendees including their name and picture.

FR3: Update profile: A student can update his profile settings and his profile picture. He can also change the notification settings.

FR4: Add comments: A student can add comments about a lecture and thus start a discussion. Others can like the comment and write follow-up comments.

FR5: Request friendship: A student can request friendship with another student who then receives a notification about the request. The second student can accept and reject friendship which both notify the first student.



FR6: Browse friend's lectures: A student can browse the lectures of his friends.

FR7: See announcements: A student can see the lecture announcements and comment or like them.

FR8: Post updates to timeline: A student can post updates to his timeline. Friends are notified about updates and can comment and like them. Certain updates are posted automatically such as saving a lecture into the lecture list or commenting in a lecture.

FR9: See lectures calendar: A student can see all lectures in a calendar and save the events to the local calendar.

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.

Additional constraints:

- The version control system must be git.
- Source code documentation must be in HTML format.
- The server system must use the Spring Boot framework.

4. Target Environment

The system should run on all desktop operation systems (Windows, macOS, Unix), either as Java or as browser-based application which communicates with the Spring Boot server application.

5. Deliverables

- Requirements Analysis Document (RAD)
- System Design Document (SDD)
- Source code under version control including source code documentation

6. Client Acceptance Criteria

The system must demonstrate at least the following functionality: It shows a list of lectures that a student can join. The student can see the attendees of a lecture with their picture and can request friendship with other students. The application communicates with the server system and conforms to the usability requirements.



Problem Statement Reservation System

1. The Problem

Restaurant visitors want to be able to reserve their table in a restaurant in advance so that they have a fixed place when they arrive.

2. Scenarios

Hans lives in Munich and knows that all the restaurants are really crowded over the weekend and getting a reservation spontaneously is almost impossible. He wants to visit an Italian restaurant on Saturday evening together with three friends and opens the Reservation system. He searches for Italian restaurants and sees different choices on the map. He also sees ratings as well as price categories. He chooses the Nero Pizza & Grill in Rumfordstraße 34 to see more details. He likes the pictures and reviews of the restaurant and clicks on the reservation button to see whether the restaurant has free tables on Saturday evening. The system shows a timetable for the upcoming seven days where unavailable times are greyed out. Hans chooses Saturday 17:00 and sees a layout of all the tables in the restaurant. Not yet reserved tables are selectable. Reserved tables are greyed out. Hans chooses the table next to the window for four people and confirms his choice. The system automatically creates a calendar event in Hans's calendar. On Friday, Hans receives a reminder about the reservation and sees that he must confirm the table again, so the reservation does not get canceled. On Saturday, Hans receives a notification at 16:00. He opens the map to navigate to the restaurant.

3. Requirements

The following functional requirements (FR) and nonfunctional requirements (NFR) must be addressed in the project.

FR1: Search for restaurants: The user can search for restaurants on a list and on a map that displays up to 50 restaurants.

FR2: See restaurants details: The user can see pictures, ratings and comments of the restaurant as well as opening times and a link to the website.

FR3: Filter search results: He can filter the results by the restaurant type, the prize category, by distance around a certain location, by the average rating and by free time slots for reservations for specified dates and number of visitors.

FR4: Reserve table: A user can see the times when he can reserve a table in the chosen restaurant. After clicking on the time, the user sees an overview of all tables in the restaurant. He can choose the exact table the free one in the overview and thus reserve the table for the specified number of visitors.

FR5: Save calendar event: When the user reserves a table, an event in the local calendar is created for the reservation.



FR6: Confirm reservation: A user is reminded about a reservation one day before the actual date of the reservation and must confirm it until latest 12 hours before the actual date. If the user does not confirm, his reservation is cancelled automatically.

FR7: Cancel reservation: A user can cancel his reservation at any time up to two twelve hours before the actual date of the reservation. After the confirmation (see FR5), the user cannot cancel the reservation anymore.

NFR1: Usability: The system should be intuitive to use, and the user interface should be easy to understand. Simple interactions should be completed in less than three clicks. Complex interactions should be completed in less than six 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.

Additional constraints:

- The version control system must be git.
- Source Code Documentation must be in HTML format.
- The server system must use the Spring Boot framework.

4. Target Environment

The system should run on all desktop operation systems (Windows, macOS, Unix), either as Java or as browser-based application which communicates with the Spring Boot server application.

5. Deliverables

- Requirements Analysis Document (RAD)
- System Design Document (SDD)
- Source code under version control including source code documentation

6. Client Acceptance Criteria

The system must demonstrate at least the following functionality: It shows a list of restaurants where a visitor can reserve tables. The user can choose a restaurant and reserve a table on a specified timeslot for the specified number of persons. The application communicates with the server system and conforms to the usability requirements.