

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 |
| 1.4 Definitions, acronyms, and abbreviations | 3 |
| 1.5 References | 3 |
| 1.6 Overview..... | 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..... | 4 |
| 3.3.5 Implementation Requirements..... | 4 |
| 3.3.6 Interface Requirements..... | 4 |
| 3.3.7 Packaging Requirements..... | 4 |
| 3.3.8 Legal Requirements | 4 |
| 3.4 System models..... | 4 |
| 3.4.1 Scenarios | 5 |
| 3.4.2 Use case model | 5 |
| 3.4.3 Object model | 5 |
| 3.4.4 Dynamic model | 6 |
| 3.4.5 User interface..... | 8 |
| 4. Glossary | 8 |

Document History

| Rev. | Author | Date | Changes |
|------|----------------------|------------|---------|
| | Patrick Rank | | |
| | Rico Finkbeiner | | |
| | Simone Domenici | 16.07.2022 | |
| | Jonas Hitter | | |
| | Caroline Stallknecht | 16.07.2022 | |

1. Introduction

1.1 Purpose of the system

Restaurant Reservation System where a client can filter Restaurants according to certain criteria's, confirm/cancel a Reservation via E-Mail and save a Calendar Event to this reservation.

1.2 Scope of the system

On any modern device with a browser and an internet connection.

1.3 Objectives and success criteria of the project

It should work

1.4 Definitions, acronyms, and abbreviations

none

1.5 References

Stackoverflow

1.6 Overview

A Restaurant Reservation System

2. Current system

As it is a website it doesn't require a specific OS to work as long device has a browser and internet connection.

3. Proposed system

3.1 Overview

A user can make a reservation.

3.2 Functional requirements

The following functional requirements (FR) were 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.*

3.3 Nonfunctional requirements

3.3.1 Usability

The following NFR regarding usability was addressed:

The system is 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.

3.3.2 Reliability

3.3.3 Performance

3.3.4 Supportability

3.3.5 Implementation Requirements

The following NFR regarding usability was addressed:

Conformance to guidelines: The design of the system should conform to the typical usability guidelines such as Nielsen's usability heuristics.

3.3.6 Interface Requirements

The following NFR regarding usability was addressed:

A server subsystem with a couple of services must be used in the system.

3.3.7 Packaging Requirements

3.3.8 Legal Requirements

3.4 System models

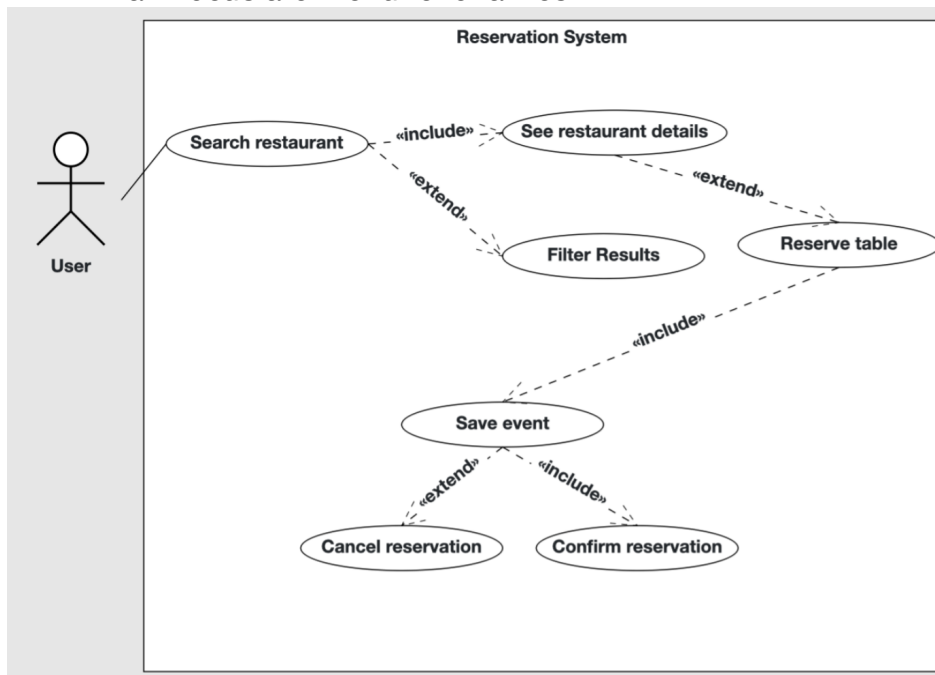
The System models include scenarios, use cases, object model, and dynamic models for the system. This section should contain the complete functional specification,

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

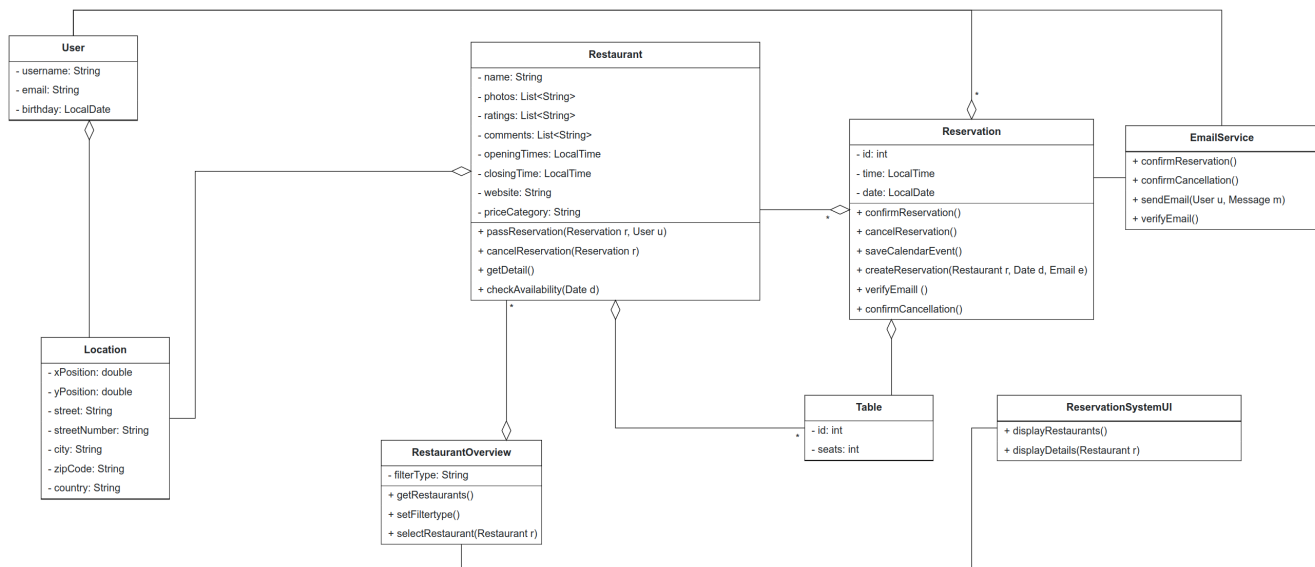
3.4.2 Use case model

A use-case model is a model that show the interaction of different users with the system to solve a problem. As such, it describes the goals of the users, the interactions between the users and the system, and the required behavior of the system in satisfying these goals. In the following case there is only one user and the main focus are the functionalities.



3.4.3 Object model

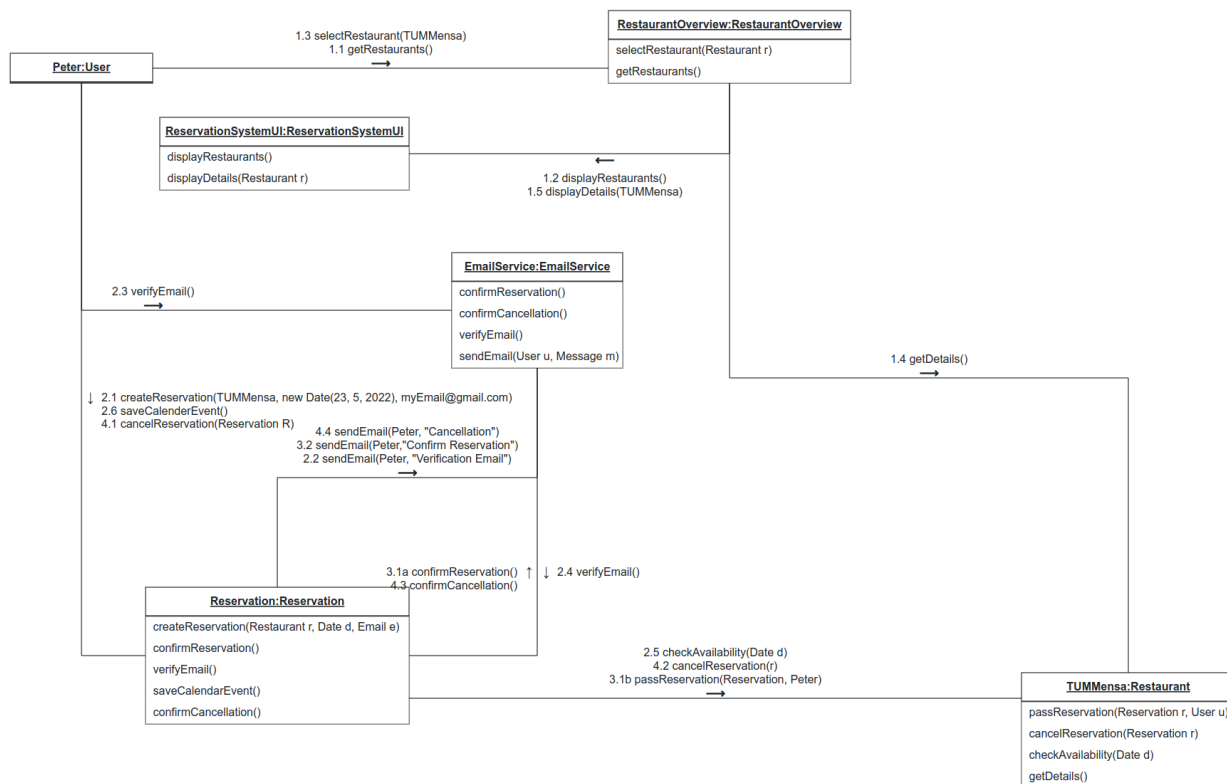
An object model is a logical interface that shows classes, their relationships as well as interactions between them. It enables the creation of an architectural software or system model prior to development or programming. In the following the object model for the reservation system is shown.



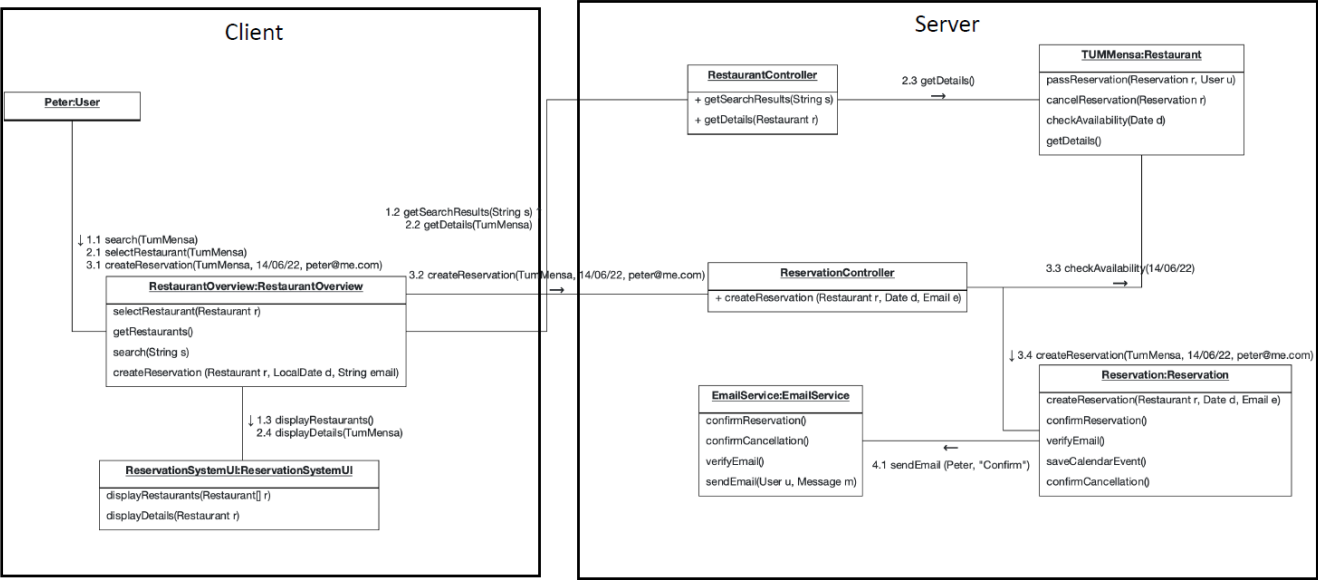
3.4.4 Dynamic model

The dynamic model represents the time-dependent aspects as events of a system. It shows the flow of events and messages that happen between the objects.

In the first diagram the user makes a reservation and the basic flow of events is shown.



In the second diagram the user Peter makes a reservation at TUM Mensa. In advance he specifically searched for the TUM Mensa, displayed its details and made a reservation for a specific date. Also the clear distinction between client and server is shown.



3.4.5 User interface

4. Glossary

npm