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

Document History

Rev.	Author	Date	Changes
1	N/A	2022-07-22	Added results of group work.

1. Introduction

1.1 Purpose of the system

The system allows persons to find restaurants and to reserve tables there.

1.2 Scope of the system

The system primarily focuses on the potential customers of restaurants and does not aim to provide management tools for the owners.

1.3 Objectives and success criteria of the project

The project is successful if members learn how to structure large applications and we get working result. Customers should be able to search for and reserve tables at restaurants.

1.4 Definitions, acronyms, and abbreviations

We aim to use clear language without the need for these kind of definitions.

1.5 References

See the problem statement for more information.

1.6 Overview

2. Current system

There is no existing system.

3. Proposed system

3.1 Overview

Users will be able to go to a website where they select a restaurant. Then they will be asked for a time and a table. That table will be reserved for them.

3.2 Functional requirements

Here we quote from the problem statement:

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

Here we quote from the problem statement and add our own:

3.3.1 Usability

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.

3.3.2 Reliability

The server should not crash under any circumstances.

3.3.3 Performance

The user should not need to wait for more than 2s for any action or feedback.

3.3.4 Supportability

3.3.5 Implementation Requirements

NFR3: Server system: A server subsystem with a couple of services must be used in the system.

3.3.6 Interface Requirements

The server and client should communicate over a REST API.

3.3.7 Packaging Requirements

The customer should not need to install any specialized software. The server should be simple to distribute.

3.3.8 Legal Requirements

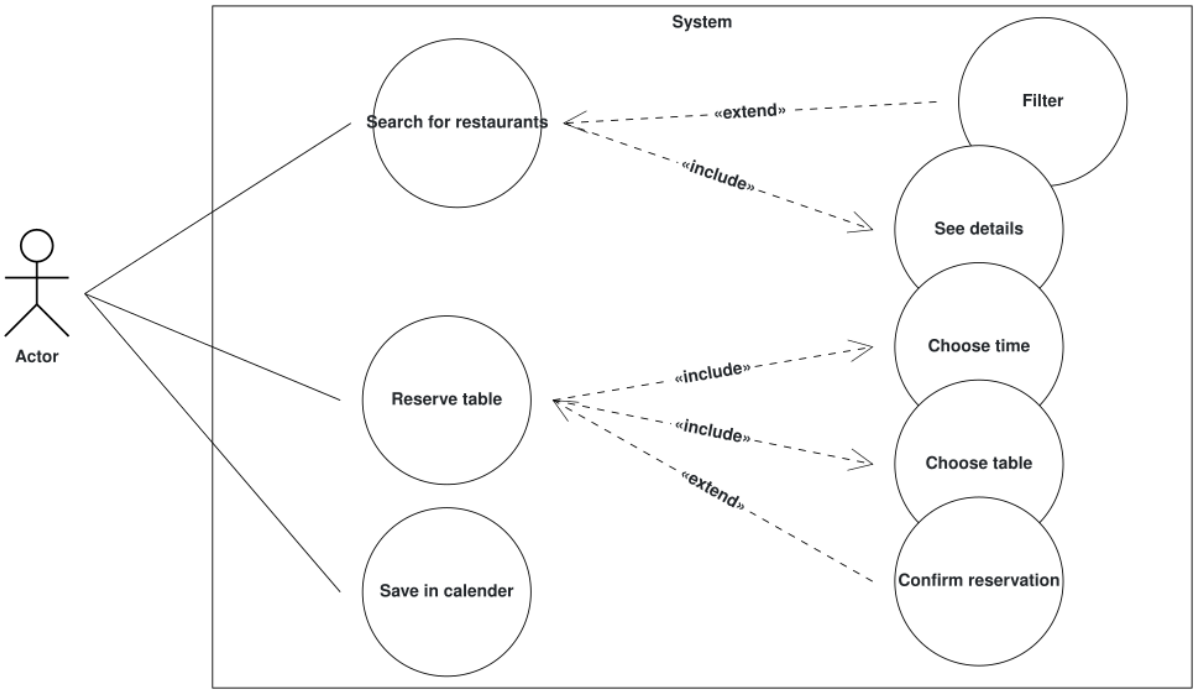
There are no legal requirements.

3.4 System models

3.4.1 Scenarios

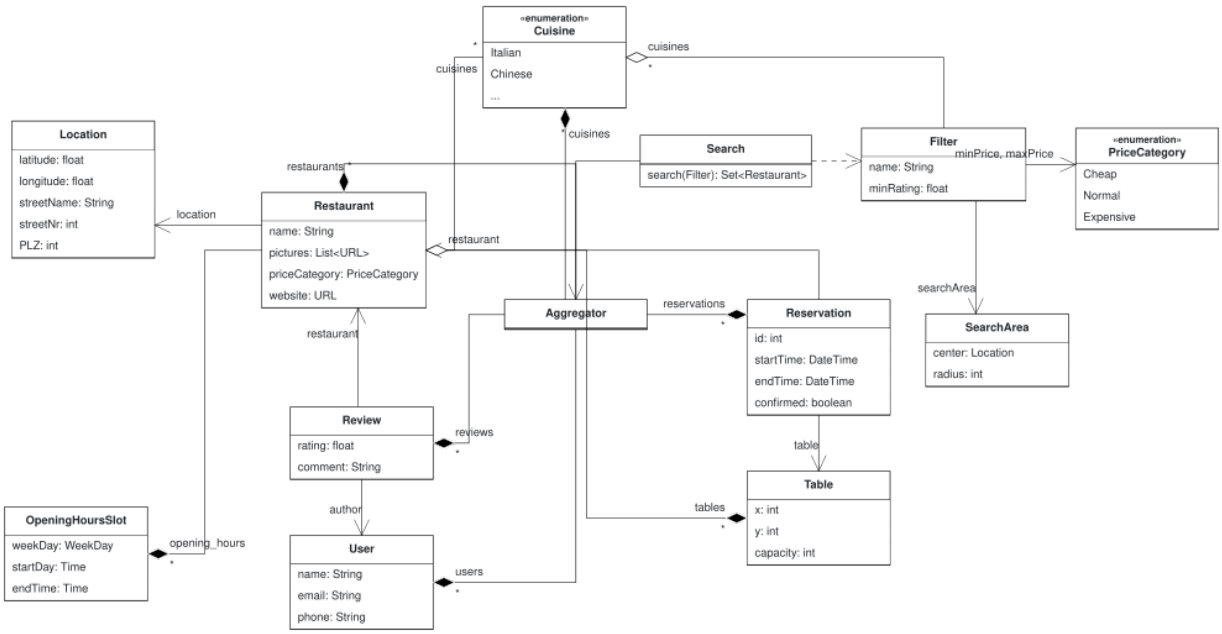
See the problem statement.

3.4.2 Use case model



3.4.3 Object model

3.4.4 Dynamic model



3.4.5 User interface

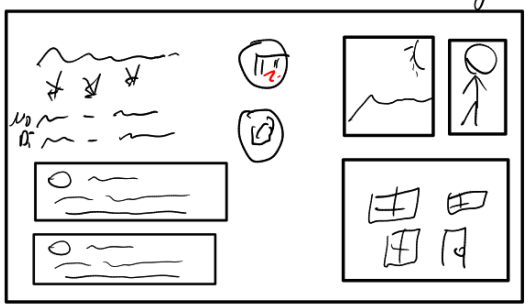
1. Search: Map, Filters, List

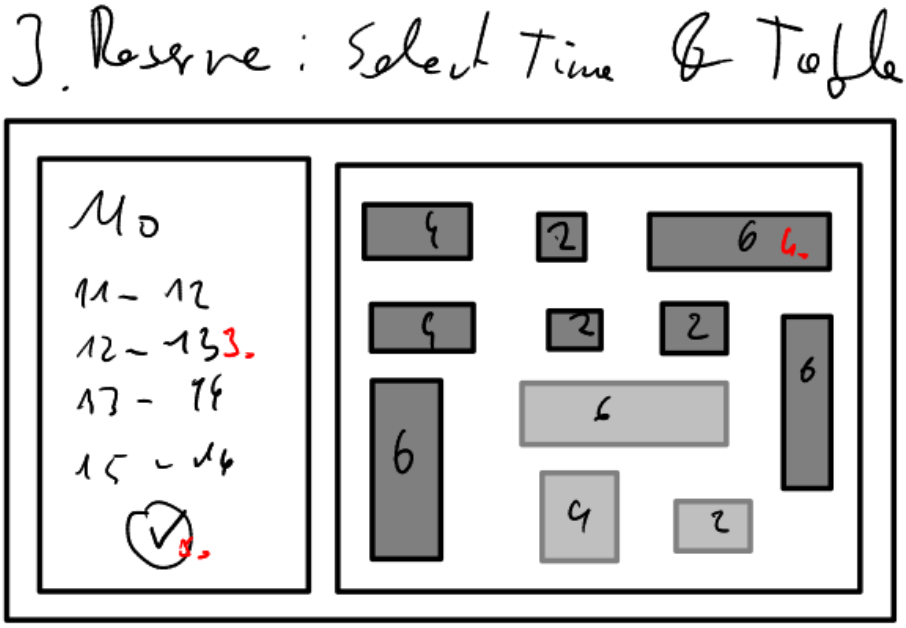
maybe add filters

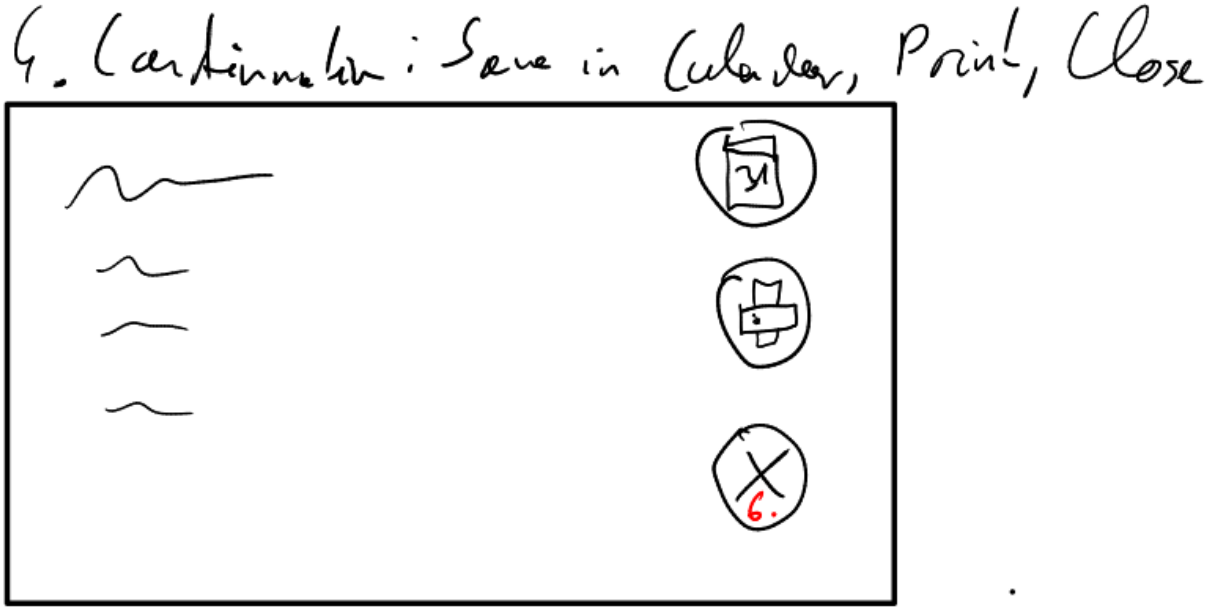
3 clicks for simple

6 clicks for complex

2. Detailed Overview: Relize, Pictures, Opening Hours, Contact, Link to Website, Reserve




3. Reserve: Select Time & Table
Mo
11 - 12
12 - 13
17 - 19
15 - 16
4.
6.


4. Confirmation: Save in Calendar, Print, Close
6.

4. Glossary