Purpose

The system design is documented in the System Design Document (SDD). It describes additional design goals set by the software architect, the subsystem decomposition (with UML class diagrams), hardware/software mapping (with UML deployment diagrams), data management, access control, control flow mechanisms, and boundary conditions. The SDD serves as the binding reference document when architecture-level decisions need to be revisited.

Audience

The audience for the SDD includes the system architect and the object designers as well as the project manager.

Table of Contents

1. Introduction 2
   1.1 Overview 2
   1.2 Definitions, acronyms, and abbreviations 2
   1.3 References 2
2. Design Goals 2
3. Subsystem decomposition 2
4. Hardware/software mapping 2
5. Persistent data management 2
6. Access control and security 2
7. Global software control 2
8. Boundary conditions 3

Document History

Rev.	Author	Date	Changes
	Patrick Rank		
	Rico Finkbeiner		
	Simone Domenici	19/07/2022	
	Jonas Hitter		
	Caroline Strallknecht		

1. Introduction

1.1 Overview

A Restaurant Reservation System

1.2 Definitions, acronyms, and abbreviations

None

1.3 References

None

2. Design Goals

Usability was our Design Goal: the user can complete simple interactions in 3 clicks and complex interactions in 6 clicks

3. Subsystem decomposition

We divided our Project in two Subsystems: Client and Server.

4. Hardware/software mapping

There are Docker images for the client and the server which can enable deployment on servers.

5. Persistent data management

We used a self-made JSON DB to store the Data on the Server.

6. Access control and security

As the user does just enter his data at every reservation, it does not get stored and thus no Access Control or additional Security is required.

7. Global software control

Our Server can manage concurrent clients.

8. Boundary conditions

Start up Server: run DemoApplication.java
Start up Client: run npm start in Terminal