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

Document History

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

1. Introduction

The purpose of the Introduction is to provide a brief overview of the software architecture. It also provides references to other documents.

1.1 Overview

Our project uses a client server architecture.

1.2 Definitions, acronyms, and abbreviations

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

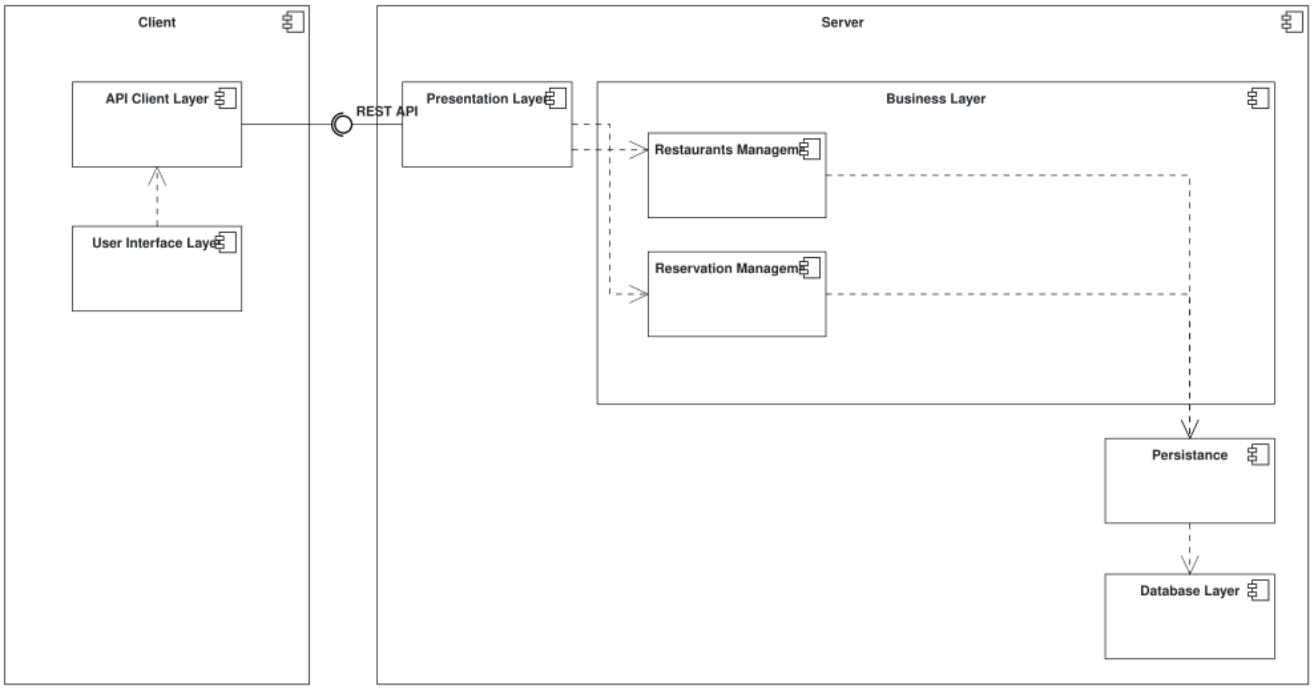
1.3 References

See the problem statement and Requirements Analysis Document.

2. Design Goals

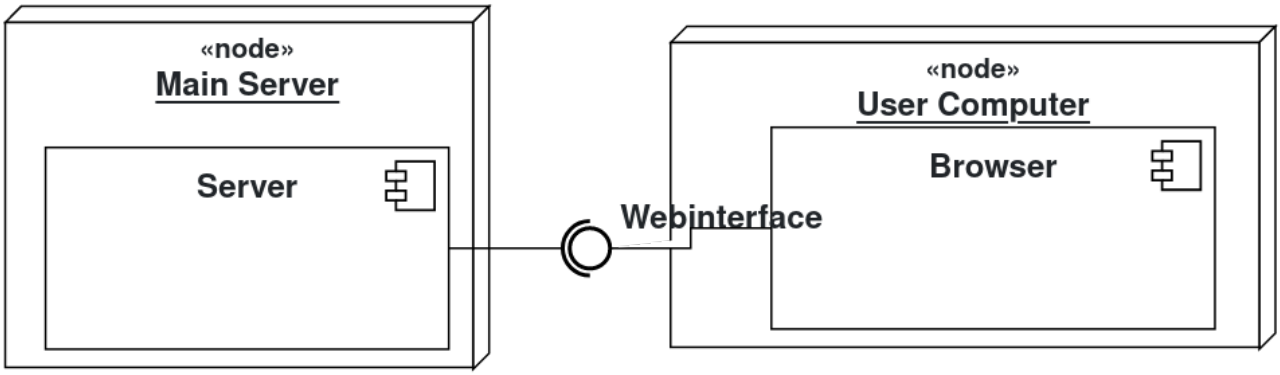
We aim for rapid development. Well-defined interfaces are no priority as both client and server can be redeployed instantly. The user interface should be user-friendliness, see the NFRs in the RAD Document for more information.

3. Subsystem decomposition



4. Hardware/software mapping

There is a server providing both the business logic and user interface running on a central node. The user accesses the user interface in their browser.



5. Persistent data management

There is no persistent data storage for privacy reasons.

6. Access control and security

Users can only see their own reservations. The one logged in user is determined at compile time for this early prototype.

7. Global software control

The service is provided by a monolithic server.

8. Boundary conditions

The one monolithic component can be started from source using `./gradlew bootRun` or from the distributed JAR using `java -jar "AAMMN Reservation System-1.0.0.jar"`