AY 2021/2022

Politecnico di Milano

DD: Design Document

Ottavia Belotti   Alessio Braccini   Riccardo Izzo

Professor
Elisabetta Di Nitto

Version 1.0
November 29, 2021

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to provide a full technical description of the system described in the RASD document. In this design document we discuss about both hardware and software architectures in terms of interaction among the components that represent the system. Moreover there are mentions about the implementation, testing and integration process. This document will include technical language so is primarily addressed to programmers but stakeholders are also invited to read it in order to understand the characteristics of the project.

1.2 Scope

The scope of this design document is to define the behavior of the system in both general and critical cases, and to design the architecture of the system by describing logical allocation of the components and the interaction between them. This document also extends in part to the implementation and testing plan, where one possible course of action is explained, user interface design of user applications and requirements traceability relating to the RASD.

1.3 Definitions, acronyms, abbreviations

Acronyms

- **DREAM**: *Data-driven predictive farming*
- **RASD**: Requirement Analysis and Specification Document
- **DD**: Design Document
- **API**: Application Programming Interface
- **DBMS**: Database Management System
- **UML**: Unified Modeling Language
- **GPS**: Global Positioning System
- **IT**: Information Technology

- **GUI**: Graphic User Interface

1.4 Revision history

1.5 Reference documents

- Specification document: "Assignment RDD AY 2021-2022"
- Requirements Analysis Specification Document (RASD)
- UML documentation: https://www.uml-diagrams.org/
- ArchiMate documentation: https://pubs.opengroup.org/architecture/archimate3-doc/
- Slides of the lectures

1.6 Document structure

- **Section 1** gives a brief description of the design document, it describes the purpose and the scope of it including all the definitions, acronyms and abbreviations used.
- **Section 2** delves deeply into the system architecture by providing a detailed description of the components, the interfaces and all the technical choices made for the development of the application. It also includes detailed sequence, component and ArchiMate diagrams that describes in depth the system.
- **Section 3** contains a complete description of the user interface, it includes all the client-side mockups with some graphs useful to understand the correct execution flow.
- **Section 4** links the RASD and the DD, it maps the goals and the requirements described in the RASD to the actual functionalities presented in this DD.
- **Section 5** presents a description of the implementation, testing and integration phases of the system components.

2 Architectural Design

2.1 Overview

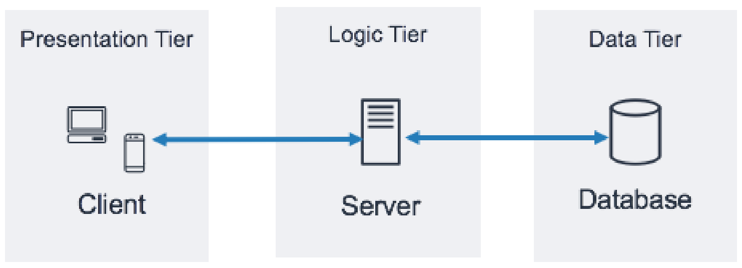

Figure 1: Three-Layer Architecture

The system is a distributed application that follow the common client-server paradigm. The architecture of the application is structured in three logic layers:

- **Presentation Layer (P)**: it manages the presentation logic and handles the user actions. It is characterized by a GUI (Graphic User Interface) that allow the user to interact with the application in a simple and effective way.
- **Logic or Application Layer (A)**: it manages all the functionalities that has to be provided to the users, it is also responsible of data exchange between the client and the data sources.
- **Data Layer (D)**: it manages the access to data sources, it gets data from the database and move them through the other layers. It is essential to guarantee a high level of abstraction from the database in order to provide a model easy to use.

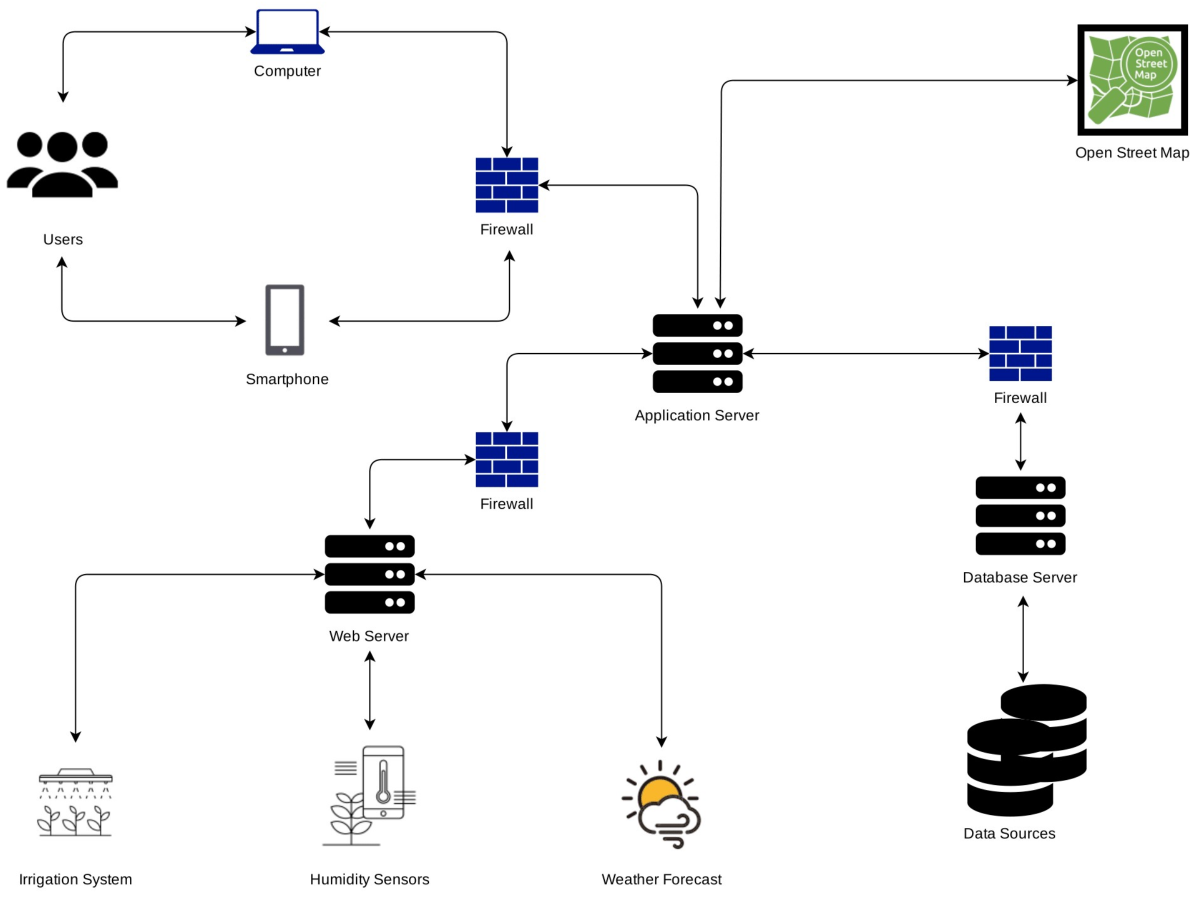

Figure 2: High Level Architecture

The system is based on a four-tier architecture(Client, Web Server, Application Server and Database Server), this ensures more flexibility and scalability. The tiers are separated by firewalls in order to guarantee a higher level of security of the whole system. A thin client is used to prevent heavy computation load client side, all the heavy operations are executed at the server side. The client's devices can be a personal computer or a mobile device, both communicate directly with the application server through a web browser. The application server communicates with the database server to store data. For what concern the other components (water irrigation system, humidity sensors and weather forecasts system) they are connected to the web server that exchange data with the application server. This allow to retrieve data like weather forecasts directly from it in order to show them to the user. The web server also manages the data coming from the irrigation system and the humidity sensors. Finally, to enable the geolocalization, the application server use the API provided by Open Street Map. All the components will be described in depth in the following sections.

2.2 Component view

General Component View

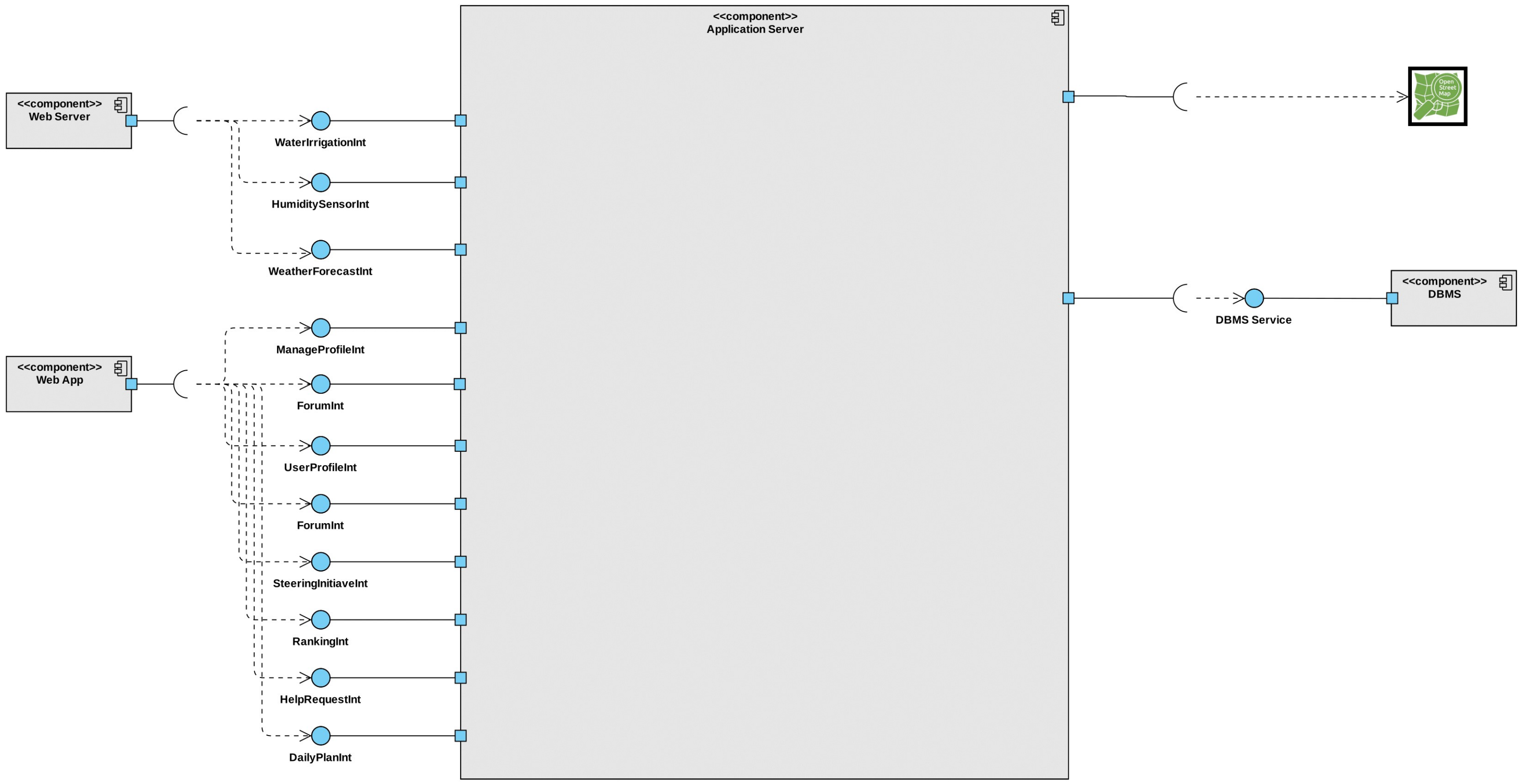

Figure 3: General Component View

Application Server Component View

2.3 Deployment view

2.4 Runtime view

2.5 Component interfaces

2.6 Selected architectural styles and patterns

2.7 Other design decisions

3 User Interface Design

4 Requirements Traceability

5 Implementation, Integration and Test Plan

6 Effort Spent

7 References