POLITECNICO
MILANO 1863

Software Engineering II

A.Y. 2021/22

DREAM

Data-dRiven prEdictive fArMing

Design Document

Authors:
Mauro Famà
Giacomo Lombardo

January 9, 2022

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to comprehensively describe the design of the DREAM platform. It will be described the architecture of the system and its characteristics, moreover will be analyzed its components explaining their functionalities.
This document was written following the RASD of the same project, in fact the assumptions and design choices are based on the assumptions already explained, however maintaining independence between the documents. This document focuses on the explanation about the Software-To-Be, its modules, its interfaces and its implementation.

1.2 Scope

The main objective of the platform is to create a support tool and communication channel for farmers in Telangana, leveraging the IT infrastructure of government resources. The platform also aims to be a supervisory and control system for the authorities of the Ministry of Agriculture in Telangana, providing accurate and bulk data of farmers' performance, reducing the overhead due to manual retrieval of this information.
The system will have two separate and exclusive interfaces, one dedicated to PMs in Telangana and one dedicated to farmers in the region. Farmers will be provided with a browser executable web application, it will provide a dashboard to monitor the sensors embedded in their fields, weather forecasts for the local area and summarize the resources obtained and consumed by the plantations. In addition to the monitoring functionality, from the farmer's dashboard it will be possible to compile reports to be sent to the PM responsible for the area, combining data entered manually by the farmer, data created by IoT devices and data retrieved from the internet.
The web application also provides access to the forum dedicated to the Telangana farming community, connecting all participants in the DREAM program through its threads. Farmers will also be able to submit their help requests into the system, which will be automatically forwarded to the PM responsible for the area, who in turn will have a dedicated dashboard exclusively for the administrative side. From his application, the PM will be able to forward responses to help requests, monitor the performance of farmers in his area, all while viewing data from sensors and weather forecasts. From his dashboard, the PM will have the tools to perform end-of-quarter performance evaluations, including direct communication channels to affected farmers.
There is no platform that currently provides these functions, in fact this document refers to the design of a system created from scratch, without any integration of existing systems, but exploiting the IT infrastructure already physically installed in the Telangana region.

1.3 Definitions, Acronyms, Abbreviations

1.4 Revision History

1.5 Reference Documents

1.6 Document Structure

2 Architectural Design

2.1 Overview

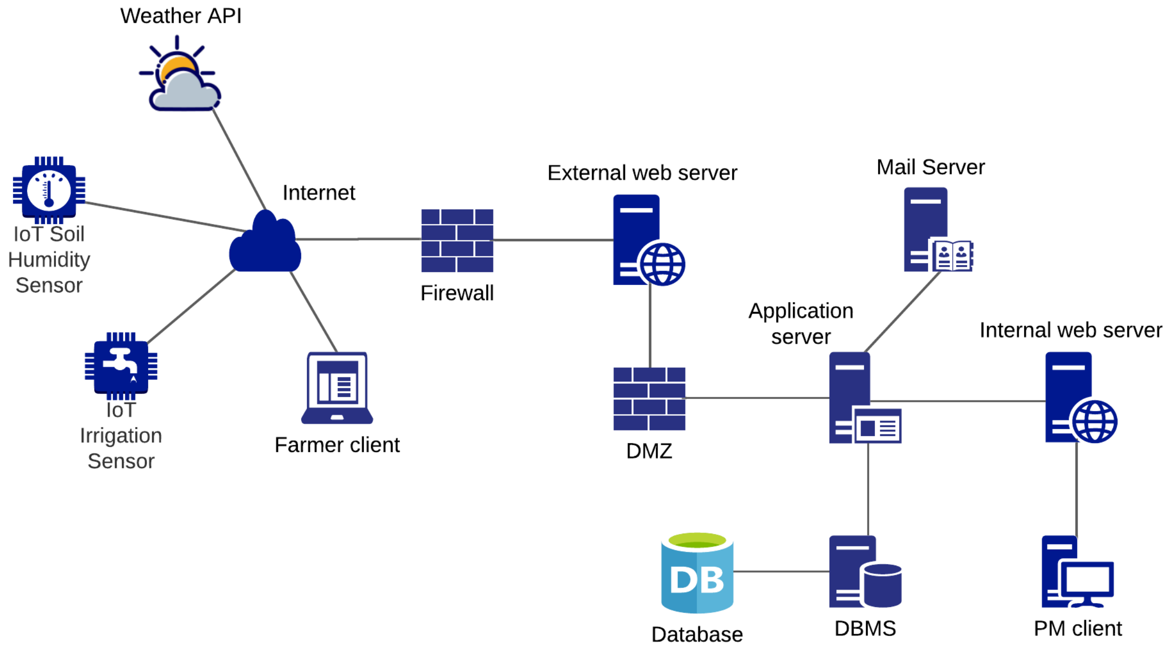

Figure 1: System architecture

The DREAM application is designed with a multiter architecture, the overall system is split into multiple pieces where the database and data management is separate from the application server. We have a three-tier application with web servers, application server, and a database functioning as the three tiers of the application. This application has both external (Farmers) and internal (PMs) users which use a web-based application provided by web servers. This web-based component then communicates back to a common application server. IoT sensors are connected to the internet in order to communicate with the farmers'

application, indeed, MQTT communication protocol is used to retrieve sensor data by the farmers' machine. All the weather informations are taken from the weather forecast of Telangana through the system'a APIs. Finally, the application server communicate with a database through a Database Management System.
A full description of each component will be given in the following sections.

2.2 Component view

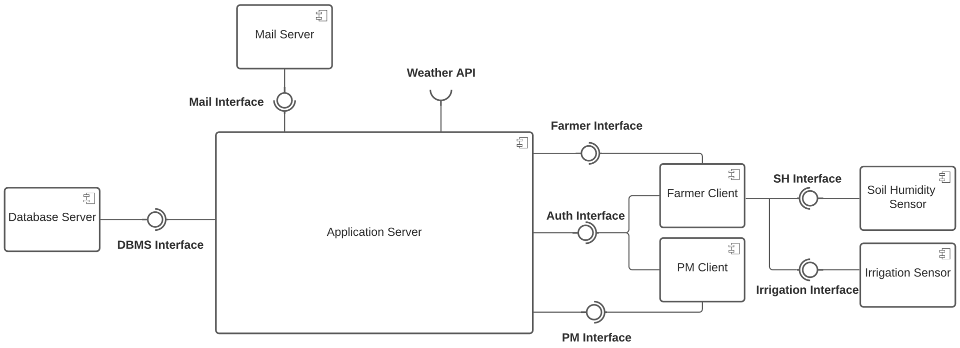

Figure 2: High Level Component Diagram

2.3 Deployment view

2.4 Runtime view

2.5 Component Interfaces

2.6 Selected Architectural styles and patterns

- 3 tier client-server
- middleware based
- rest api

2.7 Other design decisions

- event based communication for sensor

3 User Interface Desgin

4 Requirements Traceability

5 Implementation, integration and test plan

6 Effort spent

References