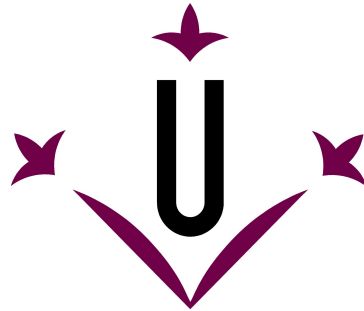Escola Politècnica Superior

Master of Computer Engineering

ICT Project: Development and Implementation

Universitat de Lleida

Sprint 1:
Requirement definition and analysis

Students:
Roger Truchero Visa   Gerard Donaire Alos
David Sarrat González   Adrià Casals Espax
Joan Pau Castells Gasia
Date: October 25, 2019

Contents

Introduction

The main idea is to design a watering system that allows us from a mobile app to water our plants remotely. This system will have different options of watering: Manual, planned and automatic sheet that the user can configure from the mobile app.

Additionally, we will install a plague recognition system formed by a neuronal convoluted network model that will feed images that you will receive from the webcam of the Raspberry Pi. All this information can be obtained with the option to monitor.

This means that we can go on vacation peacefully without having to worry about our plants and with the option to see their current status.

Part I
Sprint 1

1 Requirements

1.1 Non-functional requirements

1. **Product**

   (a) **Accessibility and Usability**

       i. Regardless of the style of interaction, the interface has to be simple and intuitive, providing a high level of interactivity and usability. The tasks will be as visual as possible, since the application will be oriented to all types of age ranges.

   (b) **Concurrency**

       i. The app has to be multi-user, allowing the usage of several users simultaneously of the application.

   (c) **Portability**

       i. **Adaptability (multi-device):** It is required that the design is "responsive" in order to ensure proper display on multiple devices, such as tablets, smartphones...

   (d) **Support**

       i. The mobile app will be developed by the Android platform.

2. **External**

   (a) **Privacy**

       i. All data management has to conform to the requirements of the organic law of data protection in order to preserve privacy in the processing of personal data.

1.2 Functional requirements

1.3 List of functionalities and features

List of functionalities and features that all services (Android app, web service and Arduino) will accomplish

1.3.1 Android app

- **The Android app will have to** allow communication with the web service.

- **The Android app will have to** allow each user to log in.

- **The Android app will have to** allow each user to configure/create an irrigation unit independently.

- **The Android app will have to** show all irrigation units configured by the user.

- **The Android app will have to** allow the manual watering option for each watering unit.

- **The Android app will have to** allow planned irrigation of each irrigation unit.

- **The Android app will have to** allow the option of monitoring the crop with data obtained from sensors and the webcam.

- **The Android app will have to** allow the change of language.

1.3.2   Web service

- **The web service will have to** allow communication with the Android application and the Arduino devices.

- **The web service will have to** allow the execution of a cron to make requests to the Arduino every X time so that a user can automatically receive notifications of his crop.

- **The web service will have to** allow the execution of a machine learning script in charge of pest detection and return the data back to the Android application.

- **The web service will have to** be able to determine when watering is optimal (if the automatic watering function has been used without programming) and send the request to the Arduino.

- **The web service will have to** allow CRUD operations directly from the DB.

1.3.3   Detection plague script

- **The pest detection algorithm will have to** be able to perform leaf trimming, and then pass it on to the convolutional neural network.

- **The pest detection algorithm will have to**, using an image of a leaf as an input, determine if this is a pest or not.

1.3.4   Arduino

- **The Arduino system will have to** allow communication with the web service.

- **The Arduino system will have to** allow images to be sent in real time to the web service.

- **The Arduino system will have to** allow the data from the humidity, temperature and soil sensors to be captured.

2 Main use cases

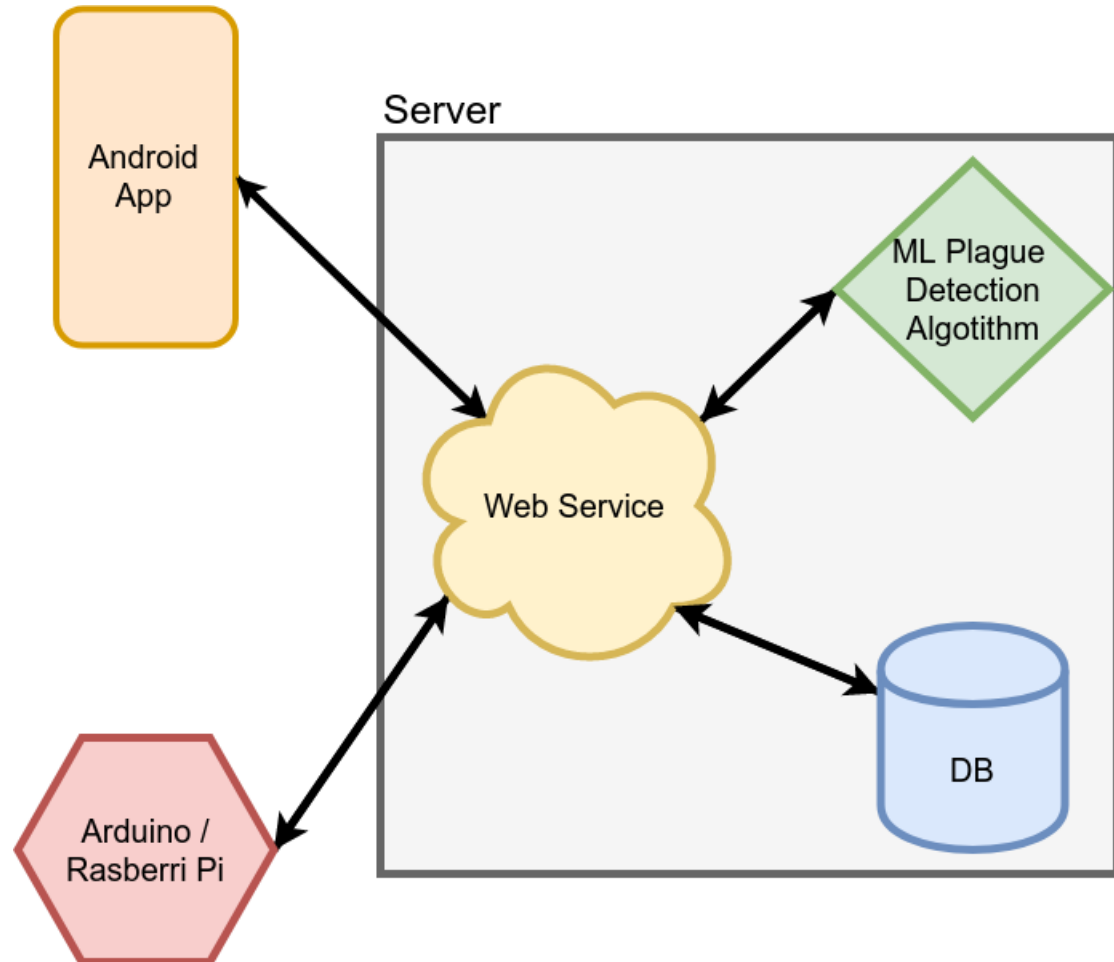## 3 General architecture

3.1 System architecture

Figure 1: System app architecture
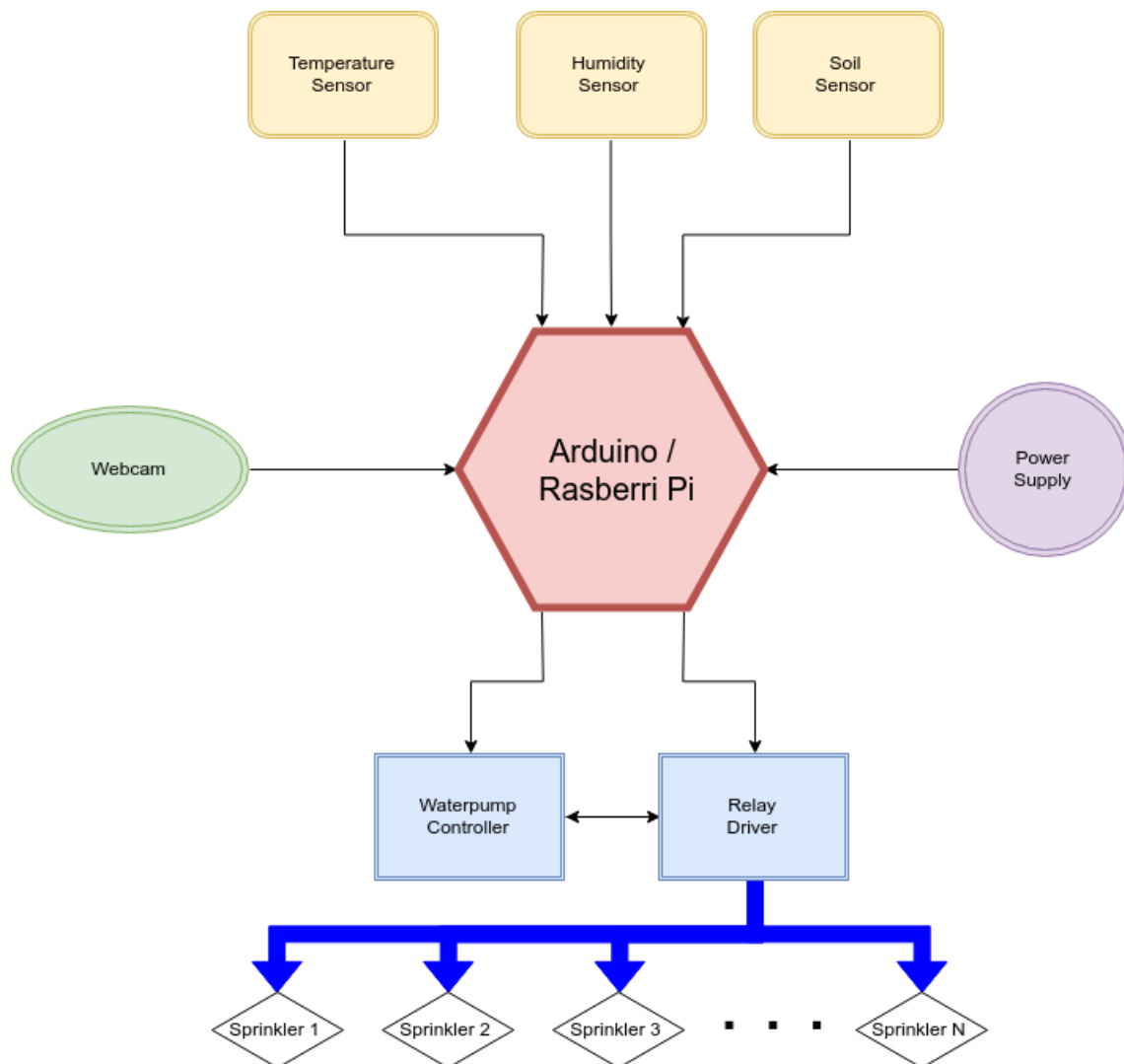
3.2 Arduino/Raspberry pi architecture

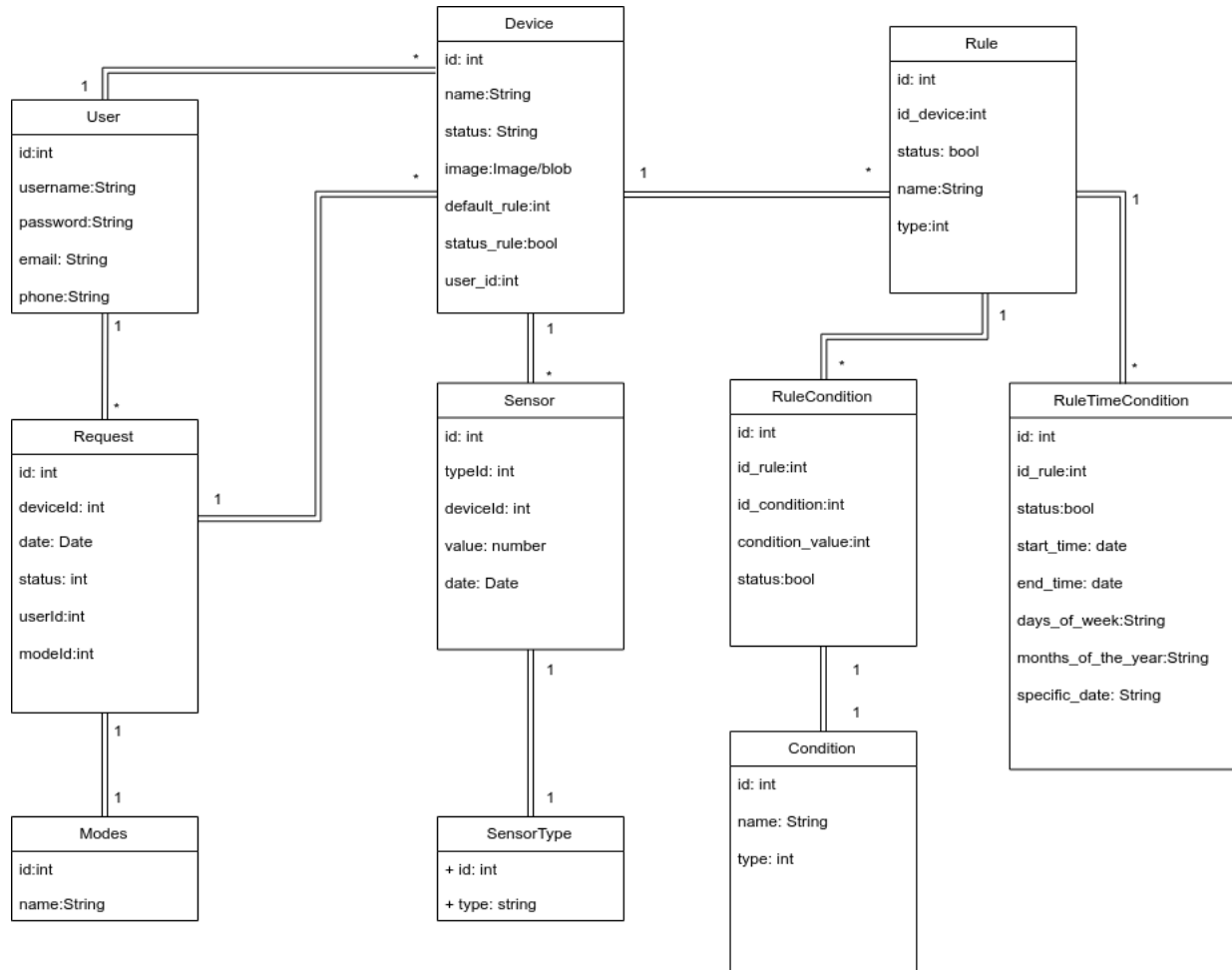Figure 2: Arduino and rapsberry pi architecture

4 Data model

Figure 3: Data model UML

5 Product backlog

6　Sprint backlog + dedication to each one

7 Main screens desgin mockup

8 Navigation schema between activities

9 Android prototype

9.1 Main screenshot files

9.2 Localization

9.3 Domain

Domain Terminology Glossary

- **Accessibility**: It is the ability to provide flexibility to accommodate each user's needs and preferences and/or limitations.

- **Concurrency**: It is a property of systems in which the processes of a computation are made simultaneously, and can interact among them.

- **Interoperability**: It is the capacity of information systems, and by extension of the procedures they support, to share data and enable the exchange of information and knowledge among them.

- **Usability**: A measure of the quality of a user's experience when interacting with a product or system.

- **User**: A person who uses the system interface.

Part II
Use case diagrams

10 Timing: Advanced iteration of the manufacturing phase

Project title: Automatic irrigation system with plague detection