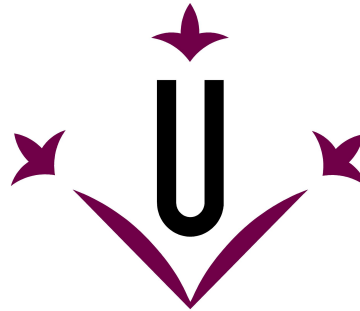


Escola Politècnica Superior

Master of Computer Engineering

ICT Project: Development and Implementation



Universitat de Lleida

Sprint 2:
Requirement definition and system analysis

Github: <https://github.com/GarduinoTeam/Garduino>

Students:

Adrià Casals Espax Joan Pau Castells Gasia
Gerard Donaire Alos Roger Truchero Visa
David Sarrat González

Date: November 20, 2019

Contents

Introduction	3
1 Product backlog + Sprint backlog	5
2 Requirements	7
2.1 Non-functional requirements	7
2.2 Functional requirements	7
2.3 List of functionalities and features	7
2.3.1 Android app	7
2.3.2 Web service	8
2.3.3 Detection plague script	8
2.4 Observations	8
3 Main use cases	9
4 General architecture	11
4.1 System architecture	11
4.2 RPi/NodeMCU architecture	12
5 Data model	13
6 Main screens desgin and Navigation	15
7 Financial Factors	16
8 Economic viability	16
8.1 Who?	16
8.2 How many?	16
8.3 Mareketing strategy	16
8.4 Monetization strategy	19
8.5 App costs	19
8.6 Analysis	20

List of Tables

1	Table of user stories Sprint 2	4
2	Product Backlog explicit table	6
3	Sprint 2 Backlog explicit table	6
4	Table of potential marketing progression	16
5	Table of economic analysis	20

List of Figures

1	Product backlog and sprint backlog issues	5
2	Application use cases diagram	9
3	System app architecture	11
4	Arduino and raspberry pi architecture	12
5	Data model UML	13
6	Inbound marketing scheme	19

Sprint 2

Introduction

This document highlights the main advances and aspects of work relevant to the second sprint of the development of the Garduino project.

This second iteration of the development of the project is mainly focused on the backend and frontend of the application, as well as the administration of users and the database.

As such, the main aspects of the progress and changes in the project, as well as the requirements of the resulting sprint, product backlog and sprint backlog will be presented.

As a result of this sprint, the general architecture of the application, a database model and a navigation scheme of the different screens and their internal relationship will be presented, both for the part of the web application and for the part of the Android application. Finally, it will offer a greater degree of deepening in financial factors and economic feasibility study.

Id	As a	I want to be able to	So that	Priority	Sprint
1	Administrator	Create, modify and delete devices	I can properly configure my irrigation system as a whole	HIGH	2
2	Administrator	Enable and disable devices	I can properly coordinate the different devices in my irrigation system at each time	MEDIUM	2
3	Administrator	Create, modify and delete rules	I can properly set up the behavior of each device	HIGH	2
4	Administrator	Enable or disable rules	I can perform punctual changes to the devices' behavior at a certain moment	MEDIUM	2
5	Administrator	Create, modify and delete conditions	I can customize and set up what each rule stands for in a high depth level	HIGH	2
6	Administrator	Enable or disable conditions	I can perform punctual changes to the rules' definition at a certain moment	MEDIUM	2
7	Administrator	Request the system to start and cancel manual irrigations for a device	I can order the system to start an immediate irrigation in a device regardless of the configuration and whether the expected conditions are being met	LOW	2
8	Administrator	see the current real-time status of my device	I can keep track of each part of my irrigation system and my garden as a whole	HIGH	2

Table 1: Table of user stories Sprint 2

1 Product backlog + Sprint backlog

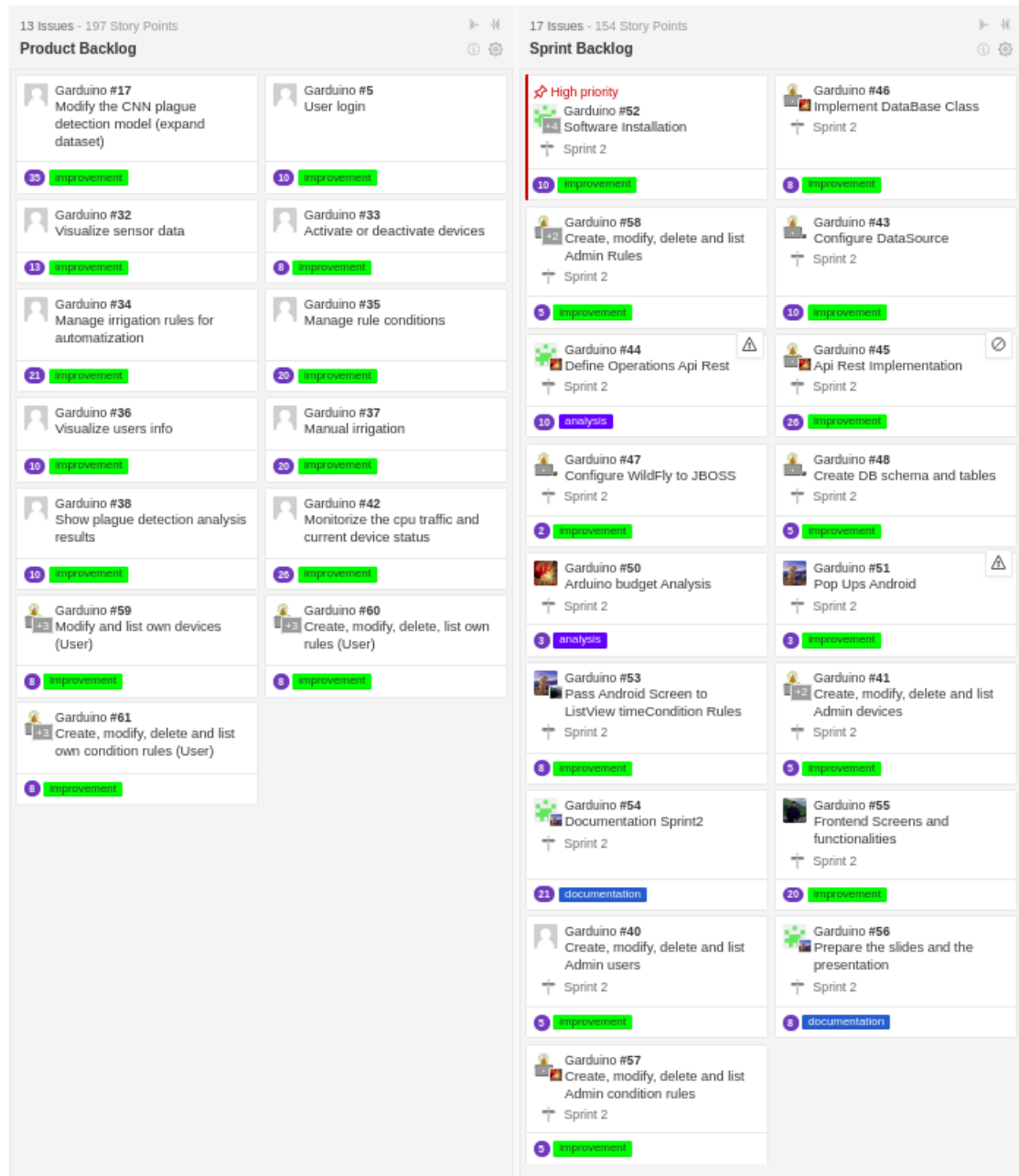


Figure 1: Product backlog and sprint backlog issues

Event	Trigger	Priority	Weight
Modify the CNN plague detection model (expand dataset)	Users	LOW	35
User Login	Users	LOW	10
Visualize sensor data	Users/Admin	MEDIUM	13
Enable or disable devices	Users	MEDIUM	8
Manage irrigation rules for automatization	Users	HIGH	21
Manage rule conditions	Users	HIGH	20
Visualize users info	Admin	LOW	10
Manual irrigation	Users	MEDIUM	20
Show played detection analysis results	Users	MEDIUM	10
Monitorize the cpu traffic and current device status	Admin	MEDIUM	26
Modify and list own devices	User	MEDIUM	8
Create, modify, delete, list own rules	User	MEDIUM	8
Create, modify, delete, list own condition rules	User	MEDIUM	8

Table 2: Product Backlog explicit table

The following issues have been added to the Product Backlog for the Sprint 3:

- Create, modify, delete, list own devices
- Create, modify, delete, list own rules
- Create, modify, delete, list own condition rules

Event	Trigger	Priority	Weight
Software Installation	Users	HIGH	10
Configure DataSource	Users	MEDIUM	10
Define Operations Api Rest	Users	LOW	10
Api Rest Implementation	Users/Admin	MEDIUM	26
Implement DataBase Class	Users	MEDIUM	8
Configure WildFly to JBOSS	Users	MEDIUM	2
Create DB Schema and Tables	Users	MEDIUM	20
Arduino Budget Analysis	Admin	LOW	10
Pop Up Screens Android	Users	MEDIUM	20
Pass Android Screen TimeConditionRules to ListView	Users	MEDIUM	10
Create, modify, delete, list users	Admin	MEDIUM	5
Create, modify, delete, list devices	Admin	MEDIUM	5
Frontend screens and functionalities	Admin	MEDIUM	5
Create, modify, delete, list rules	Admin	MEDIUM	5
Create, modify, delete, list condition rules	Admin	MEDIUM	5
Documentation Sprint 2	Team	HIGH	21
Prepare the slides and the presentation	Team	HIGH	8

Table 3: Sprint 2 Backlog explicit table

2 Requirements

2.1 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.

2.2 Functional requirements

2.3 List of functionalities and features

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

2.3.1 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.

2.3.2 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.

2.3.3 Detection plague script

- **The pest detection algorithm will have to**, using an image of a plant as an input, determine if is contaminated or not.
- RPi(Raspberry Pi) 3B (master):
 - **The RPi will have to** allow communication with the web service.
 - **The RPi will have to** send/receive data from/to the slaves to perform operations.
- NodeMCU ESP8266 (slave):
 - **The NodeMCU will have to** be able to send the data to the RPi.
 - **The NodeMCU will have to** recolect the data from the humidity, temperature and soil sensors.
 - **The NodeMCU will have to** capture and image of the device plant.
 - **The NodeMCU will have to** control the relay to make irrigation when receives the request from the RPi.

2.4 Observations

- The leaf trimming algorithm requirement has been removed because the system will have a model that will already have the capabilities to process the whole image of the plant without the need of trimming it
- The arduino/rpi architecture has been replaced by a master/slave architecture using RPi like a master and NodeMCU slaves.

3 Main use cases

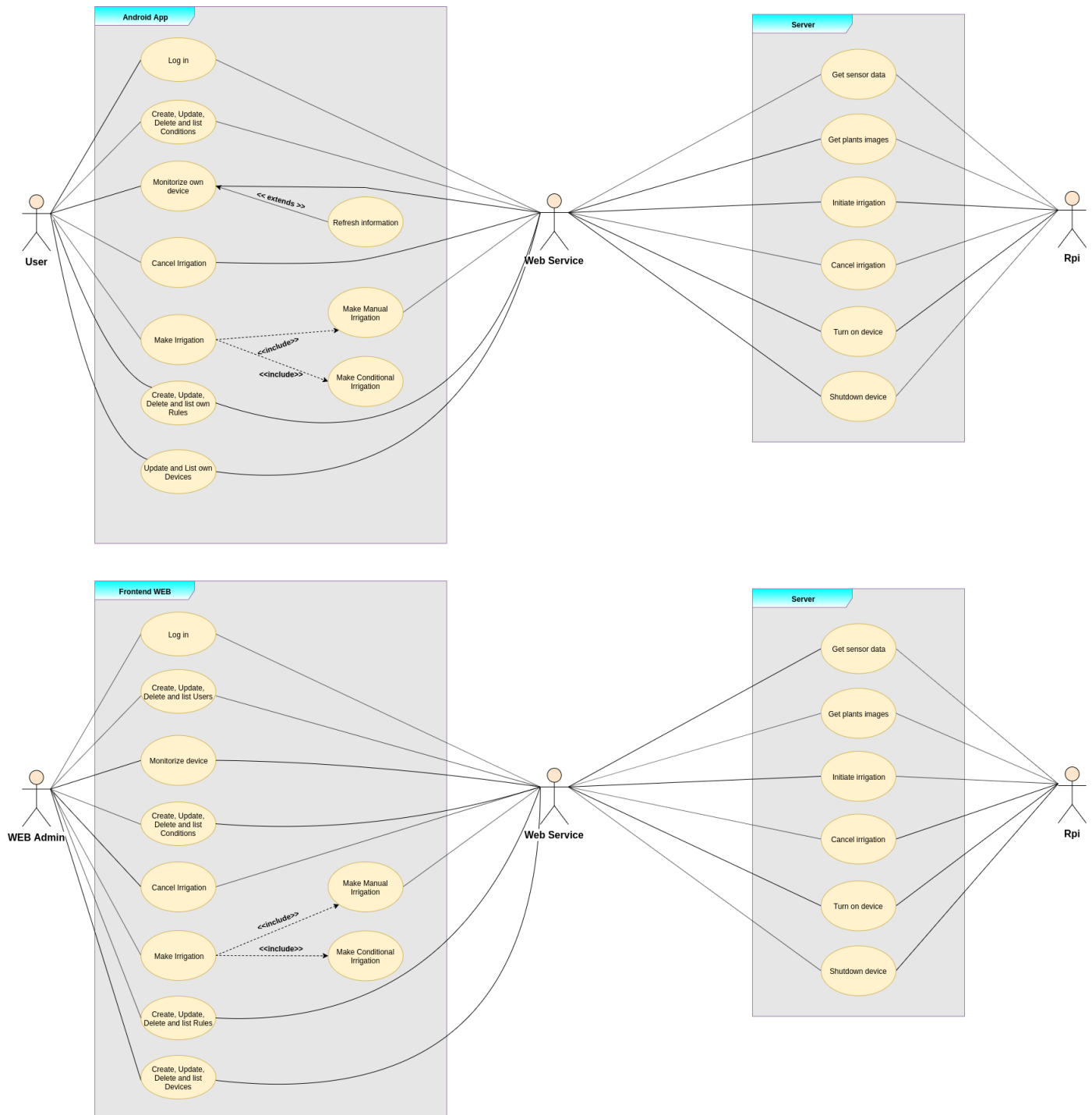


Figure 2: Application use cases diagram

En la [Figure 2](#), se muestran los siguientes actores: el usuario, el web service, el Rpi y el Admin.

El usuario podrá loggarse, pedir que se muestren sus instalaciones y editarlas. Además, también podrá monitorizar sus propias instalaciones. Asimismo, podrá crear, modificar, eliminar y listar tanto sus reglas como condiciones de riego. Con el objetivo de regar, el usuario podrá regar sus instalaciones manualmente y con condiciones de riego. Todas estas funcionalidades son recibidas por el Web Service el cual esperará las peticiones que se hagan desde la aplicación Android y este pedirá la información necesaria a la Rpi para poder dar dicha información a la parte Android y en consecuencia mostrarla.

El admin podrá loggarse, pedir que se muestren todas las instalaciones así como crearlas, modificarlas y eliminarlas. Además, también podrá monitorizarlas. A continuación, podrá crear, modificar, eliminar y listar todas las reglas, condiciones de riego y usuarios. Con el objetivo de regar, el admin podrá regar todas las instalaciones manualmente y con condiciones de riego. Todas estas funcionalidades son recibidas por el Web Service el cual esperará las peticiones que se hagan desde el frontend de la parte web y éste pedirá la información necesaria a la Rpi para poder dar dicha información a la parte del frontend y en consecuencia mostrarla.

4 General architecture

4.1 System architecture

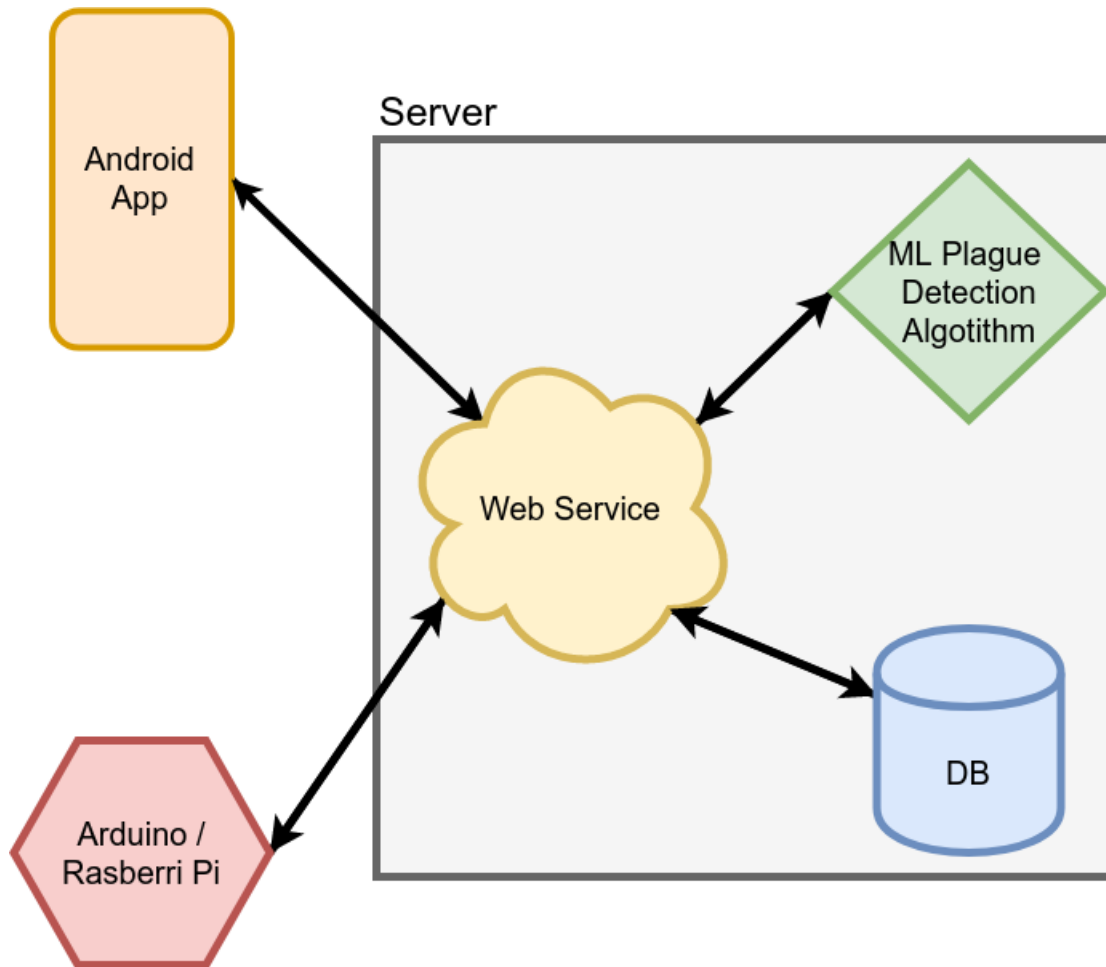


Figure 3: System app architecture

The architecture of our application is composed by:

- **Android App:** Graphical interface, in charge of collecting the data entered by the user.
- **Web Service:** Makes middleware of the application. It manages requests between the Android app and the RPi/NodeMCU, the database, ...
- **RPi/NodeMCU:** Device that will be in charge of the irrigation of each environment (gardening, flowerpot, ...).
- **Database:** Manage all the data of the application, user credentials, register all requests, device info, ...

4.2 RPi/NodeMCU architecture

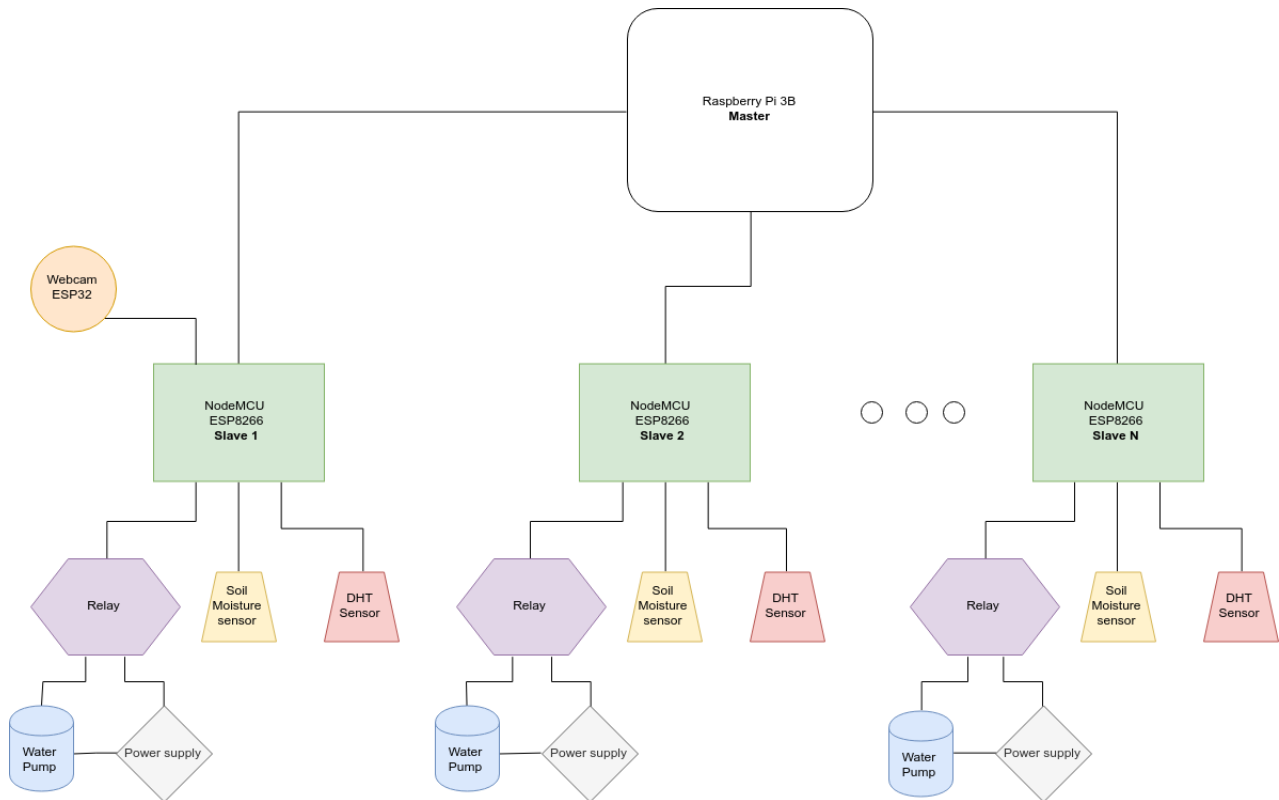


Figure 4: Arduino and raspberry pi architecture

So, we have the following components

- **Raspberry pi 3B:** We will use this microprocessor to ask the slaves for information about the sensors and the webcam, and then return the call to the web service with all the data.
- **NodeMCU ESP8266:** It will act as a device slave, in charge of collecting sensor data, activating irrigation and capturing our plant. This microcontroller will have a wifi module to maintain a wireless communication with the RPi(Raspberry Pi).
- **Webcam ESP32:** This module will be in charge of capturing the plant and sending the data to your NodeMCU.
- **Relay:** Useful to control a motor, a led strip, or any other module using arduino. The use is simple, just connect a digital output of our esp8266 to our relay module, and then we can control a power-demanding appliance with the digital signal provided by the esp8266.
- **Water pump:** He will be in charge of pumping the water in order to carry out the irrigation correctly.
- **Power supply:** Our devices will need a source of energy, in this case we have chosen a power supply of 12V to batteries. In the future it could be exchanged for a solar panel or even make a hybrid between the two depending on the climate.
- **Soil moisture sensor:** Measures the volumetric water content in soil. Since the direct gravimetric measurement of free soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content.

- **DHT sensor:** This sensor will allow us to simultaneously measure temperature and humidity. We have two types of DHT sensors, DHT11 and DHT22.

The DHT11 is a very limited sensor that we can use for training, testing, or in projects that don't really require accurate measurement.

The DHT22 has acceptable features for use in real monitoring or logging projects that require medium accuracy.

In this project we will start implementing the DHT11 version due to its low cost (70 cents).

5 Data model

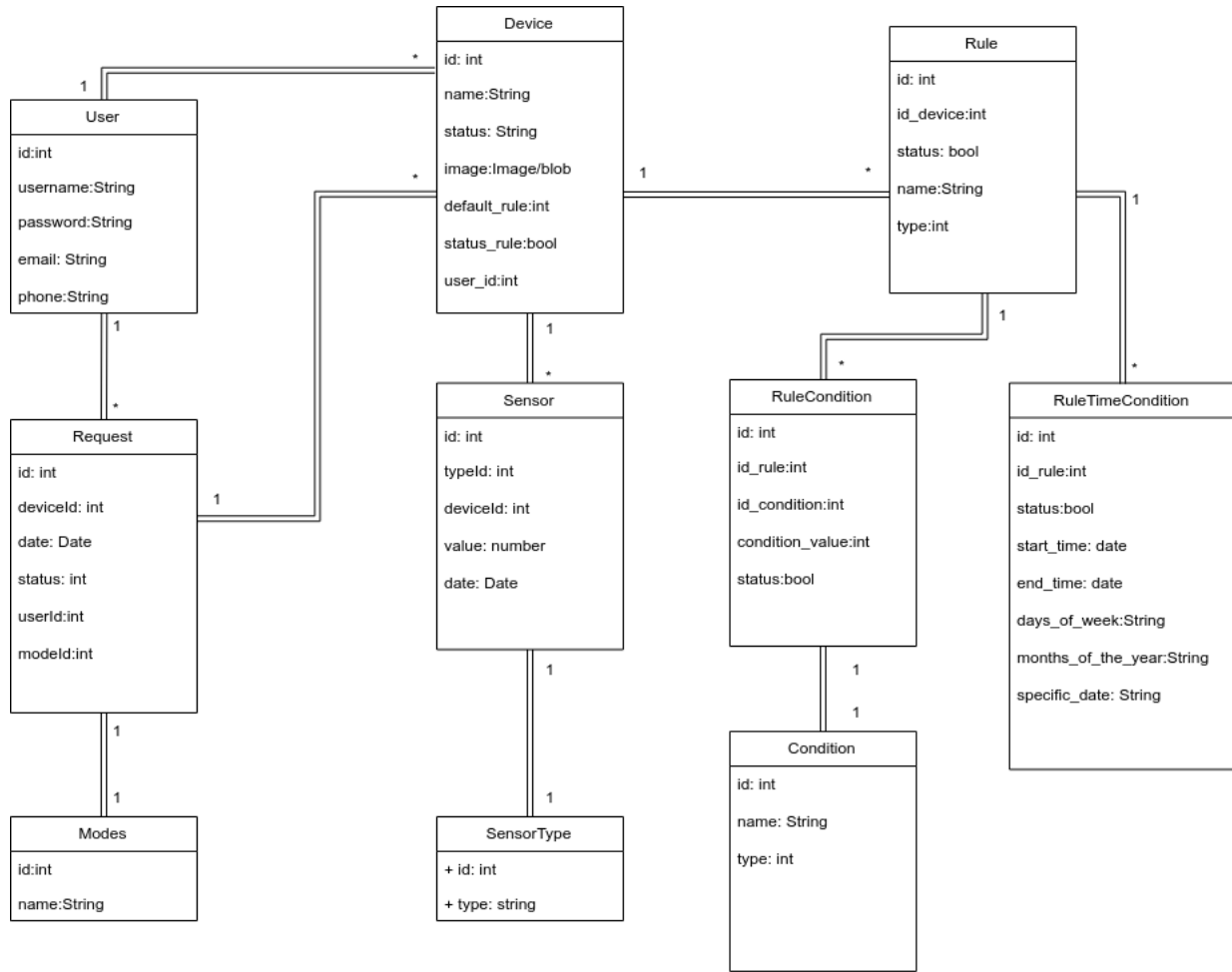


Figure 5: Data model UML

In this data model diagram we can see how our data is structured and related with the different datatables. We explain the target of the different tables:

- **User**: It stores all the data related to our users.
- **Request**: It is used to store the requests of the users to a specific device and to check if these requests have been completed.
- **Mode**: It stores the different kind of requests that a user can do like manual irrigations, conditional irrigations, the obtainment of the sensor data.
- **Device**: It stores the information of a device which corresponds to a user. It can have a default rule, so the user has not need to create its own.
- **Sensor**: It stores the value of a sensor of a specific type which correspond to a device.
- **SensorType**: It stores the different type of sensors which are: Humidity, Temperature, Soil.
- **Rule**: It stores all the configurations which correspond to a device. It can have RuleConditions and TimeConditions.
- **RuleCondition**: It stores the conditions with its respective value. It corresponds to a rule and a condition. If the sensors data of a device accomplish these conditions then an irrigation is produced.

- **RuleTimeCondition:** It corresponds to the conditions of a rule which a user decides when an irrigation has to be produced. With the status we determine if this condition has to be considered.
- **Condition:** It stores the different conditions which are: Temperature is lower than, Temperature is greater than, Humidity is lower than, Humidity is greater than, Soil is lower than, Soil is greater than.

6 Main screens design and Navigation

- Link Android Design:

[Click here: Android Screens Navigation](#)

- Brief explanation of the different screens in the Android application:

On screen 1, a list of the different devices available in the user's installation is shown. You can see the last photograph taken by the system, as well as the last date and time of irrigation. As such, the user can activate and deactivate the system for that particular device by means of a switch, or enter the panel of that particular device by pressing on the rest of the box pertinent to that device.

Screen 2 shows the control panel for a particular device. The last image obtained by the system, temperature, humidity and soil can be consulted. You can see the last time and date the data was updated. A manual irrigation can be started and the configuration of the device can be accessed.

Screen 3 shows the pop-up relevant to the operation of a manual irrigation. You can enter a numerical amount on how long the irrigation will last, and you are given the option of initializing the irrigation or canceling it.

Screen 4 shows the control panel of a device when there is a manual irrigation in progress. It indicates how much irrigation time is left, and offers the option of cancelling it instantaneously.

Screen 5 shows the configuration menu of a device. It offers the option to change its name, change the frequency of updating information and access to a panel to modify the irrigation rules. All changes will be saved when you press the save button.

Screen 6 shows the configuration panel of a device's irrigation ruleset. It shows the list of rules, which can be activated and deactivated by means of a switch, and these can be modified by clicking on them. It shows the option to add new rules and the option to save changes.

Screen 7 shows the irrigation rule creation screen. You can choose its name, confirm its creation and cancel it. This screen will be converted to a pop-up in the near future of the development of the project.

Screen 8 shows the configuration panel of the irrigation rules of a device after a new irrigation rule has been created. The internal configuration of this can now be accessed and modified by clicking on its name.

Screen 9 shows the configuration panel of an irrigation rule. This will consist of various conditions, which can be activated and deactivated through the use of a switch. It offers the option to modify the parameters of the conditions, as well as to create a new one and save the changes in that irrigation rule.

Screen 10 shows the screen to add a new condition. It shows the different types of conditions that are available.

Screen 11 shows the configuration of a time condition. It shows the various options for which the condition will be active: start and end time of the days it will be active, configuration by days of the week, by months in which it will be active, and specific dates in which the time condition will be active.

Screen 12 shows the modification of one of the parameters related to the time of a time condition (start time and end time). The specific times can be selected by specifying them in a clock.

Screen 13 shows the modification of the parameters relative to the specific dates in which the time condition will operate. The specific days can be chosen by means of a calendar.

As an improvement of the previous prototype, some screens have been changed by the pop-ups corresponding to these in order to streamline navigation and maintain an awareness of the location between the various screens on which the user navigates at all times.

- Link Web Design:

7 Financial Factors

8 Economic viability

8.1 Who?

The customer profile should be the one interested in controlling the irrigation and the health of the plants in your house in a comfortable, efficient and automated way. The population segment defined as target audience is broad and determined by these very simple conditions.

It can be focused both on private gardening and on a larger scale. Volume limits are set by the customer's business model or personal use.

Likewise, the company bets on the provision to distributors and garden shops, as well as the direct provision to individuals.

It will begin marketing the product in the area of Catalonia, where it will be made known to local distributors and will provide support in installation, configuration and maintenance to customers.

8.2 How many?

An incremental and exponential progress of the user base is estimated as a function of the increase in the company's resources as a result of the successful marketing and expansion of the product. As such, it is estimated that some representative figures of success in the first year are:

- To have marketed and installed and to have in maintenance 100 devices in particular.
- To have established supply agreements with 2 distributors.

A significant potential progression would be:

	Year 1	Year 2	Year 3	Year 4	Year 5
Particularities	100	250	500	1000	2000
Distributors	2	4	6	8	10

Table 4: Table of potential marketing progression

8.3 Marketing strategy

The main communication Garduino objectives are:

- Promote the notoriety of the brand and/or product marketed among the target audience.

- Promote knowledge of product attributes, characteristics and performance, as well as competitive differentiation with respect to the same product market.
- Establish and strengthen a connection between company and target audience, carrying out different strategies to familiarize and bring the user closer to the brand and product in marketing.
- To constitute, with regard to the potential client, one of the alternatives of the market to contemplate when acquiring a product of the nature of the article in question.

The public of interest is understood to be the group of publics or audiences targeted by a marketing campaign. It should be noted that we can differentiate between primary and secondary audiences, depending on whether or not they are involved as potential end consumers.

As such, it has established a profile of primary interest audience of the characteristics described in the section "Who"?

The communication techniques used for the promotion and diffusion of brand and product will be the following:

- **Advertising in conventional media:** transmission of the advertising message through advertisements and programmed messages in media such as television, radio...
- **Sales promotion:** alterations in the sales lot or in the price of the product in order to attract consumers.
- **Fairs and exhibitions:** Participation and presence in fairs and exhibitions to make oneself known or to collaborate in the transmission of a message.
- **Sponsorship:** Association with other initiatives, businesses, causes and figures in order to carry out an exhibition of the brand in certain contexts, as well as to carry out an exploitation of the image associated with the activity in question, and to try to carry out a commercial exploitation of activities derived from the event.
- **Personal sale:** Presential presentation and sale of the commercial to the client of the product in a physical space. This technique is usually carried out in commercial spaces, by means of informative and/or exhibition stands where the product is presented to the client and it is a question of negotiating a closing of sales with the same one.
- **Website:** Use of an informative web space where relevant company information can be announced, as well as support for online sales and subscription to other types of digital communications.
- **Blogs:** Use of web spaces structured through a system of publication of articles and news over time, which can allow different degrees of participation by users, and through which it is intended to capture an interest in the user and redirect traffic to the main website or digital content that presents information on products that may be of interest to the potential consumer.
- **Social networks:** Use of social network platforms to maintain a system of communication and promotion with the user, in addition to being a powerful tool for the approach and contact with it.
- **E-mailing:** newsletter system that transmits information and advertising relevant to the brand and product through e-mail.

The communication model presented by Garduino is the multiphase model. This communication model not only involves commercials and experts or opinion leaders as influential and diffusion figures, but it will be the target public itself that will assume a role of active participation within the communicative process. These will act as "ambassadors" of a product, brand, good, service...

This model manifests itself strongly in the growing digital and social network environments, and the search for the implementation of this kind of model by companies is evident in the incitement for the user to participate in one of the dissemination campaigns through these environments. It is also considered that this strategy will be based on the principles of inbound marketing:



Figure 6: Inbound marketing scheme

8.4 Monetization strategy

The main sources of revenue for the business model will be:

- Sale of the software to individuals
- Sale of the software to distributors
- Sale of the devices to private individuals
- Sale of devices to suppliers
- Installation service
- Maintenance service
- Special customization service

8.5 App costs

- **App type:** app for the management and monitoring of irrigation and pest control installations.
- **App platform:** Android (potentially IOS, in the future) Coding: The start-up partners will act as developers of the project software.
- **Features:** Configuration of the installed devices, live monitoring, notification system for events such as detection of anomalies/pests...
- **UI Design & Graphics:** The interface design of the app must be intuitive and user-friendly, while covering a high degree of customization options for irrigation systems.
- **Marketing:** The stipulated communication campaign will be followed, whose investment will be proportional to the level of income generated at each moment of the life phase of the product.
- **Operational costs:** Costs for the Arduino parts of the prototypes, electricity costs and maintenance of the computers with which the project is programmed.

8.6 Analysis

	Year 1	Year 2	Year 3	Year 4	Year 5
Income	20.000€	50.000€	100.000€	200.000€	400.000€
Expenses	35.000€	65.000€	65.000€	115.000€	215.000€
Benefit	-15.000€	-15.000€	35.000€	85.000€	185.000€

Table 5: Table of economic analysis