VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

MULTIDISCIPLINARY PROJECT

Project report

Greenhouse Management System

Lecturer: - Mai Đức Trung
Student: - Phạm Mạnh Dũng 1952633
- Trần Hoài Dương 1952037
- Nguyễn Bá Minh Hưng 1952748
- Phạm Nhật Hoàng 1952703

HO CHI MINH CITY, MARCH 2022

Contents

1 Introduction

A greenhouse is a system for environmental modification and management that allows plants to be grown in climates and seasons.

The monitoring of soil moisture value to adjust the amount of water for the tree is a very important thing. However, to accurately monitor and do this for a long time requires a lot of effort. With the help of electronic devices, we can do this easily, efficiently and less effort.

Automatic moisture monitoring and control systems that help us see the soil moisture value and control the remote water pump, schedule a periodic tree, monitor the soil moisture value to automatically control the operating water pump accordingly.

2 System devices

INPUT	OUTPUT
Soil moisture	DRV power circuit
Light Sensor	A 2-color single LED
DHT11	LCD I2C / OLED
Mini-pump	Relay circuit

2.1 INPUT devices

- Soil moisture: The module can be used for automatic flower watering when there is no one to manage your garden or for similar applications such as growing plants.
- Light Sensor: The module can be used for turning on a light when it becomes dark in an light-based project.
- DHT11: This is a multi-function module that can simultaneously read temperature and humidity signals. The humidity measurement range is 20 - 90 percentages, and the temperature range is 0 - 50°C, which is a standard range in regular conditions with little variation.
- Mini-pump: This is an actuator and is usually attached to a motor circuit or a Relay.

2.2 OUTPUT devices

- DRV power circuit: ChipI - Motor DRV is a module with a motor driver that belongs to ChipFC's Chip System. The output is a 4-pin barrier that is compatible with the Chip Base Shield.

- A 2-color single LED: ChipI - 2-Color LED is a single 2-color LED module that belongs to ChipFC's Chip System. The output is a 4-pin barrier that is compatible with the Chip Base Shield.
- LCD I2C/OLED: uses the I2C communication protocol and it is compatible with 3V3 and Microbit
- Relay circuit: uses to control switch.

3 Requirements

3.1 User requirements

3.1.1 Functional requirements

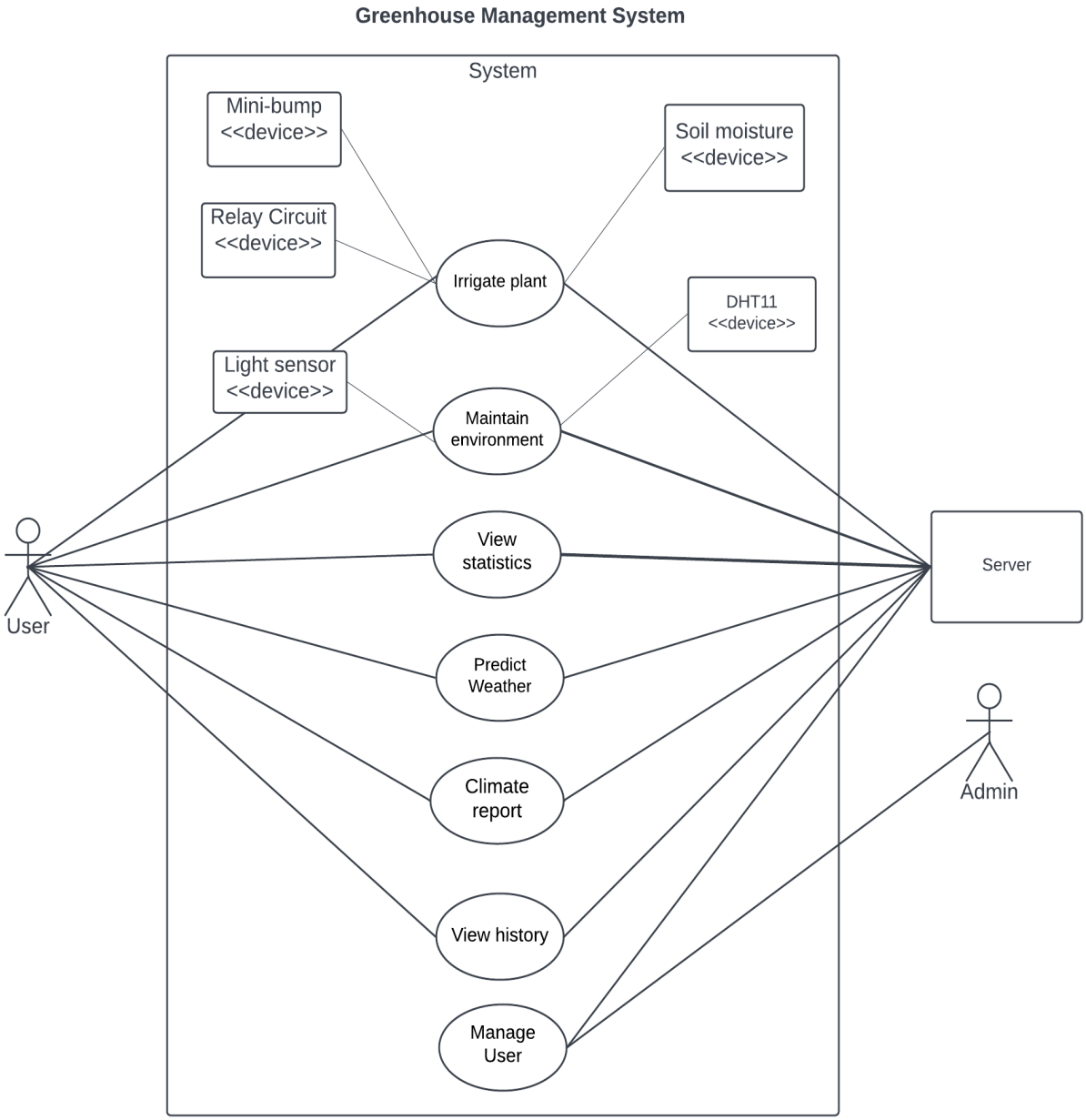
- Automatically irrigate greenhouse plants according to current soil moisture and plant type.
- Provide live climate reports to the user via some form of display (graph view and log view).
- Maintain climatic conditions according to temperature and lighting via different ways: manually, automatically or periodically by schedule.
- Predicate the next day's climate using the available model.
- Report issue : check equipment status according to data receiving from device every 5s

3.1.2 Non-functional requirements

- Run 24/7 with minimal scheduled downtime and preferably no unscheduled downtime.
- Actuator delay time should be no more than five seconds.
- Persistently log historical readings for diagnostic purposes that keep tracks of the system's status over at least 30 days.
- Simple user manuals and interactions via LCD/OLED screen or device buttons.

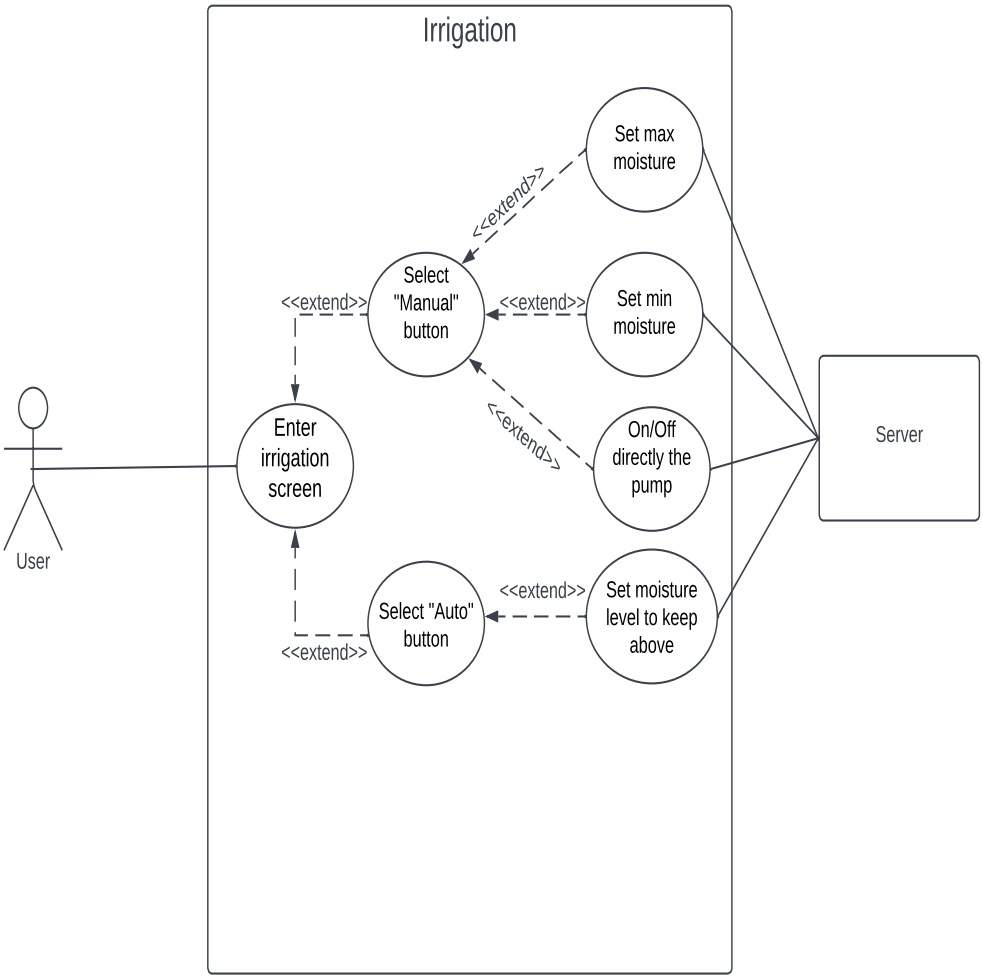
3.2 System requirements

3.2.1 Use-case diagram for whole system



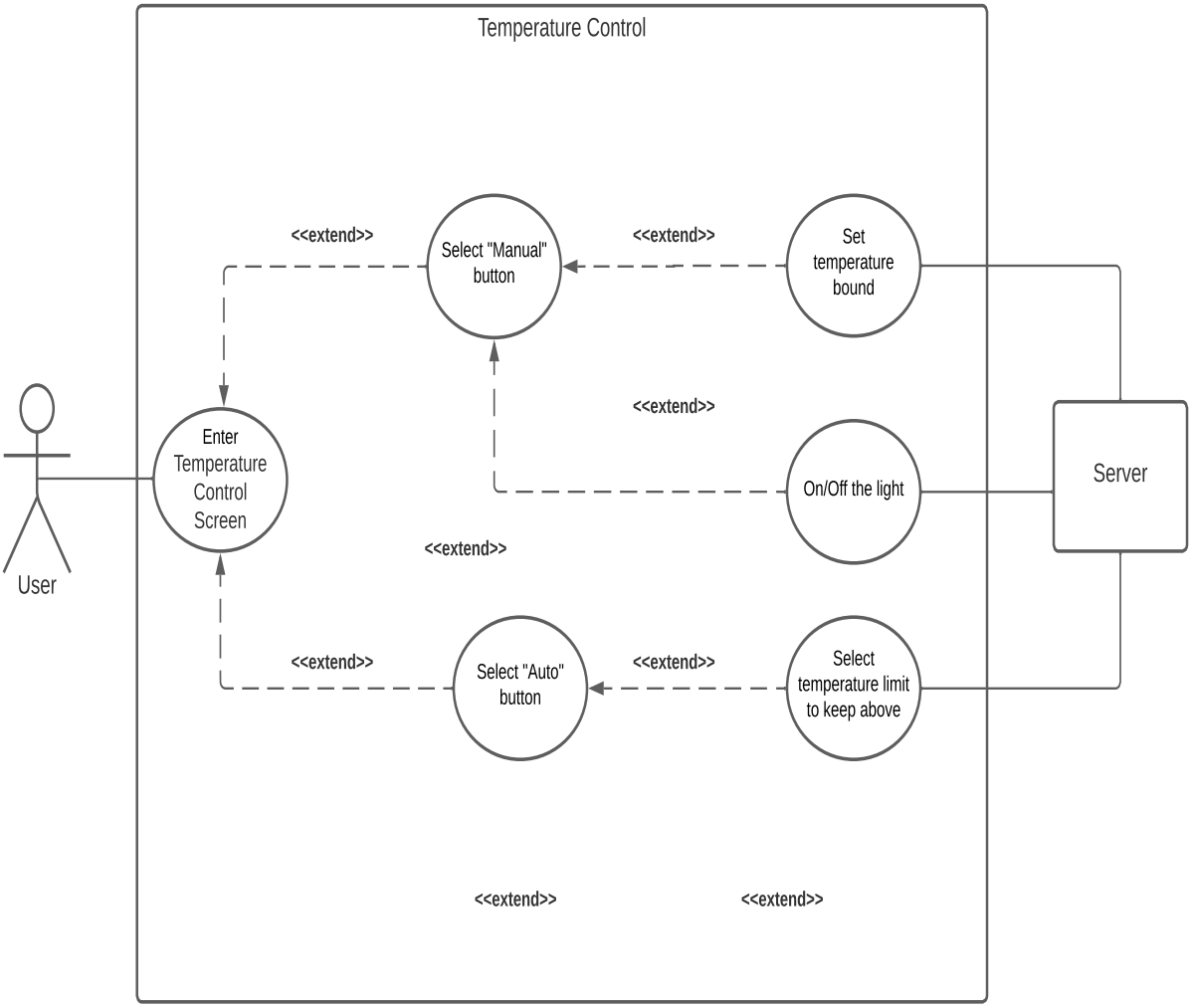
3.2.2 Use-case detail for irrigating plants

Use-case ID	No.1
Use-case name	Irrigation mode
Created by	No Hope
Actor	Users, system and devices
Description	The system provides two modes of irrigation - a fully automatic mode in which the system takes care of maintaining a preset level of soil moisture as schedule, and a fully manual mode that allows the user to explicitly start and stop the irrigation.
Trigger	None
Pre-condition	The system is running normally, and is equipped with at least a pump, a soil moisture sensor, and a water source.
Post-condition	Data is logged as often as the current plan checks soil moisture.
Normal flow	The user should delegate irrigation to the system as the intended activity. 1. The user launches the mobile app. 2. On the main page, the user selects "Irrigation" from the list of functions. 3. In the Mode section, the user selects "Automatic." 4. The user selects the desired moisture level from an on-screen table of suggested values or through a slider for bespoke settings. 5. The system examines each incoming measurement and saves it for later use.
Alternative flow	At step 3, if irrigation is to be planned, the user sets the length of the schedule (within one day, one week, or one month). The appropriate form appears. (3.2) The user specifies when and how much water to irrigate within that timetable (i.e. how long to pump for this task). If the user prefers totally manual irrigation, go to step 3: (3.1) The user determines whether the system should issue a warning in the event of potentially harmful situations, such as excessively low or high moisture levels. (3.2) If warnings are enabled, the system will check soil moisture as frequently as feasible and send out push messages if necessary.
Exception	If the system detects aberrant changes (or a lack of change) in soil moisture when irrigating at step 5, proceed to step 6. (a) While irrigating, the system detects a slower than expected change in water moisture, or no changes are observed. (b) The system provides a push notice to the smartphone application, alerting the user that the water supply (exhausted or clogged) or the pump are both malfunctioning. (c) If the fallback mechanism is activated, the system takes a more aggressive action.


Irrigation
Set max moisture
<<extend>>
Select "Manual" button
<<extend>>
Set min moisture
<<extend>>
<<extend>>
On/Off directly the pump
Enter irrigation screen
User
Server
<<extend>>
Select "Auto" button
<<extend>>
Set moisture level to keep above

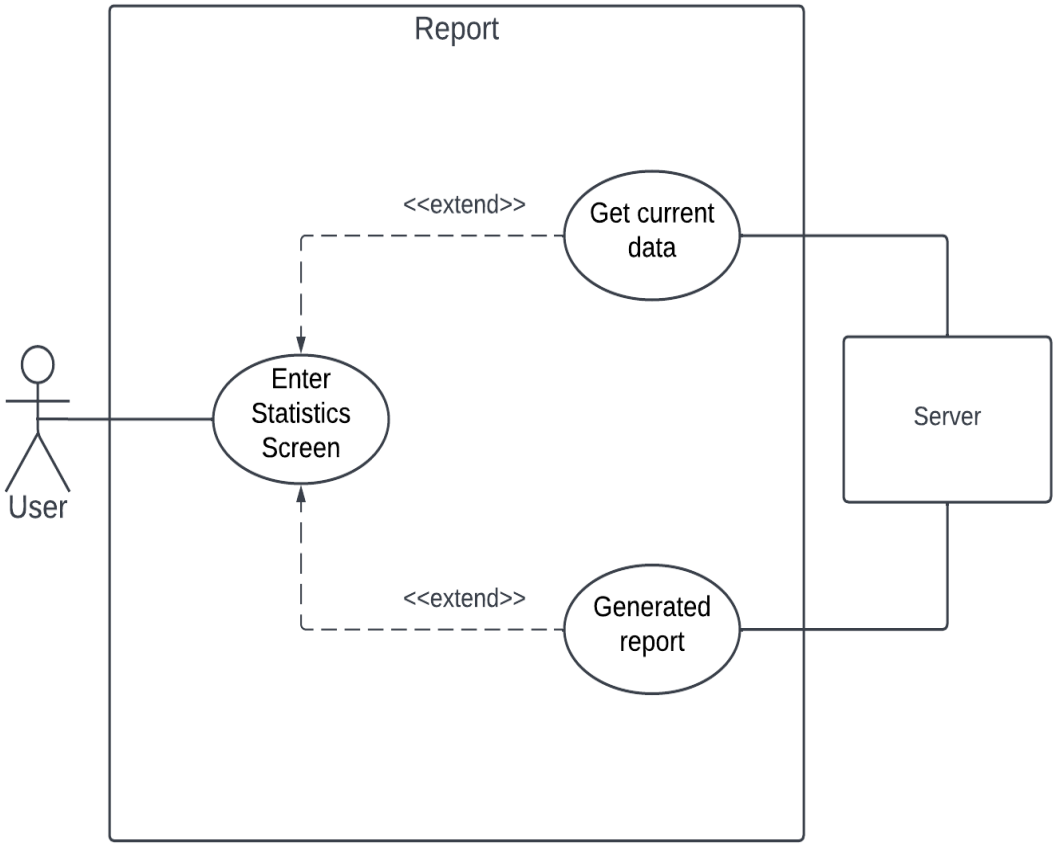
3.2.3 Use-case detail for maintaining environment

Use-case ID	No.2
Use-case name	Temperature regulation
Created by	No Hope
Actor	Users, system, and devices (DHT11, RC servo)
Description	The system also has three temperature control modes: an automatic mode that tries to keep the greenhouse at an ideal temperature, a scheduled mode that follows a user-specified schedule, and a fully manual mode
Trigger	None
Pre-condition	The system is operational and includes a DHT11 sensor, as well as sunscreens coupled to actuators if desired.
Post-condition	Temperature and sunscreen condition are reported into the system's database on a regular basis.
Normal flow	The user should delegate irrigation to the system as the intended activity. 1. The user launches the mobile app. 2. The user selects "Temperature control" from the list of functions. 3. The user chooses "Scheduled mode." 4. The user selects either "do nothing," "warn," or "warn and take action" as the safety fallback mode. 5. The system has a range-based slider that represents the 24 hours of the day. 6. The user chooses whether the sunscreens are open or closed during his or her selected time period.
Alternative flow	At step 3, if the user selects "Automatic": (3.1) If the user selects "Automatic" at step 3, the user defines the intended temperature range. (3.2) When new data is received, the system begins verifying the temperature level. At step 3, if the user selects "Manual": (3.1) The user specifies whether the system should notify of unsafe temperatures. (3.2) If warnings are enabled, the system will frequently check temperature and may issue warnings through local alerts.
Exception	None


Temperature Control
<<extend>>
Select "Manual" button
<<extend>>
Set temperature bound
<<extend>>
Enter Temperature Control Screen
On/Off the light
Server
<<extend>>
User
<<extend>>
Select "Auto" button
<<extend>>
Select temperature limit to keep above
<<extend>>
<<extend>>

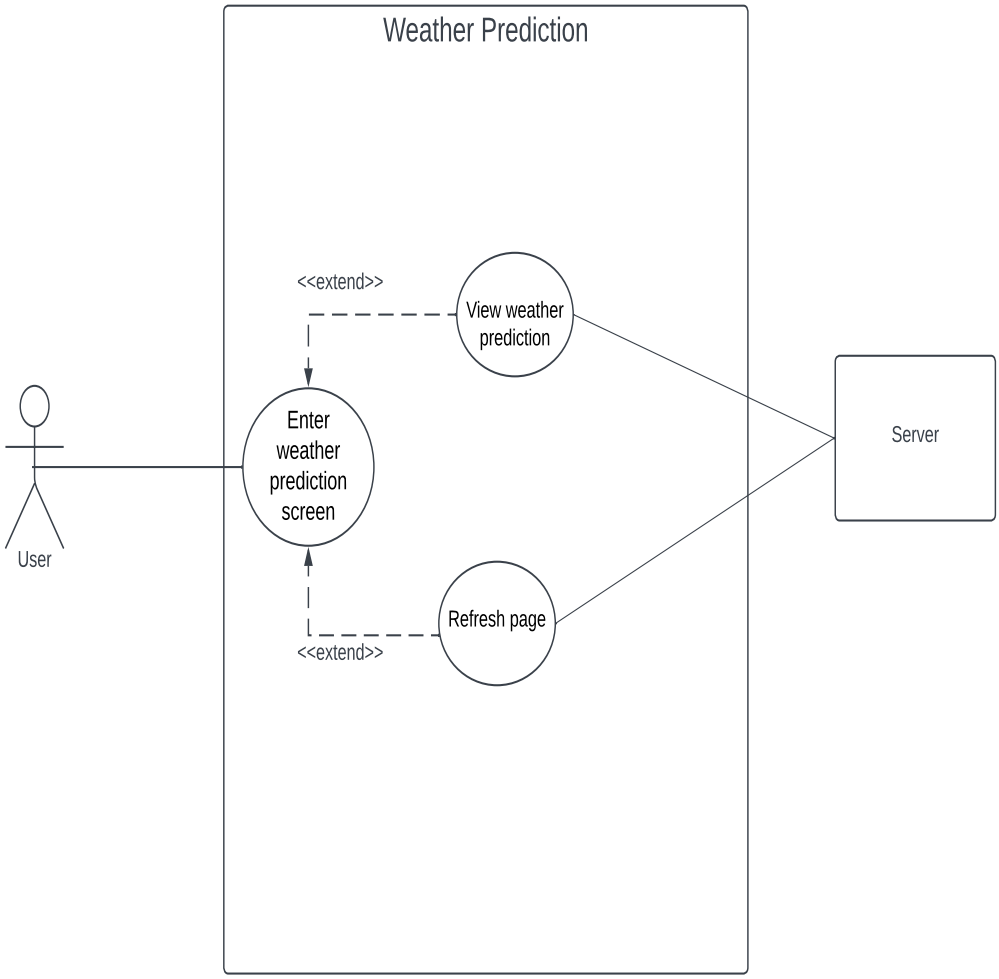
3.2.4 Use-case detail for statistics and reports

Use-case ID	No.3
Use-case name	View statistics and reports
Created by	No Hope
Actor	Users and system
Description	The program allows users to examine recorded data as a summary report (over a period of time) with aggregate calculations and a graph overview of previous recordings.
Trigger	None
Pre-condition	The database is connected to the mobile app, and the system is continuing collecting data.
Post-condition	None
Normal flow	1. User launches the mobile app. 2. On the main page, the user selects "Statistics" from the list of features. 3. As the default report mode, the system harvests daily data and displays it to the user. The report provides line graphs of light levels, moisture levels, and temperature levels over time. 4. To return to the main page, the user taps on the return arrow.
Alternative flow	None
Exception	At step 3, if the time period selected by the user is not in the database system, the system displays an Error box


Report
<<extend>>
Get current data
Enter Statistics Screen
User
Server
<<extend>>
Generated report

3.2.5 Use-case detail for weather prediction

Use-case ID	No.4
Use-case name	Weather prediction
Created by	No Hope
Actor	Users and system
Description	The system provides a mode of prediction where
Trigger	None
Pre-condition	The system have been running for 5 consecutive days and have enough data for running model.
Post-condition	Weather prediction will be presented
Normal flow	The expected actions would be for the user to delegate the weather prediction to the system: 1. User opens the mobile application. 2. User selects ”Future Weather” from the list of functions on the main screen. 3.The system will show the prediction based on old data(with percentage).
Alternative flow	The system will gather data and base on the algorithm in Python, it will return the future data. After step 3 , the user will receive the report of weather can be happened in the future. If the report have some issues, the user can push the button "Run again" to start again the model.
Exception	We will set up a range of normal weather information can be happened. If the weather is out of that range, we will examine again the data by some algorithm and model again the model function.


Weather Prediction
<<extend>>
View weather prediction
Enter weather prediction screen
User
Server
Refresh page
<<extend>>

4 User interface

4.1 Sign Up Screen

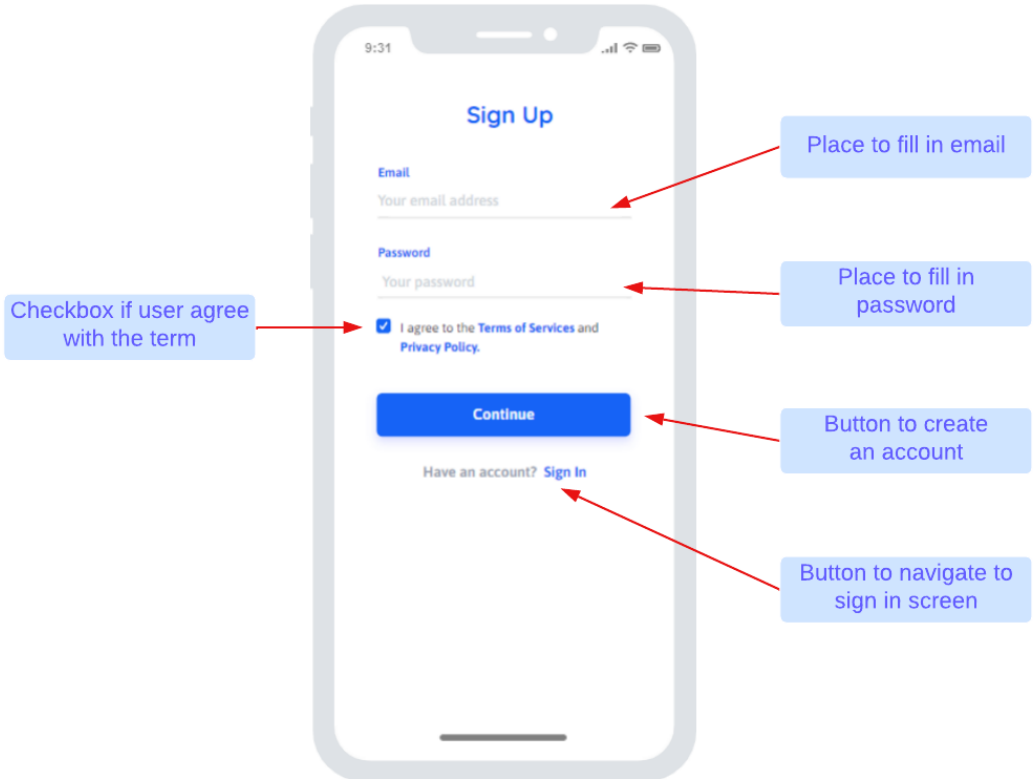


Figure 1: Sign Up

In this screen, users will create their own account by filling the form. After finishing the form, users may click 'Continue' button and it will navigate them to log in screen. If users already have an account, they can click 'Sign In' to navigate to log in screen.

4.2 Sign In Screen

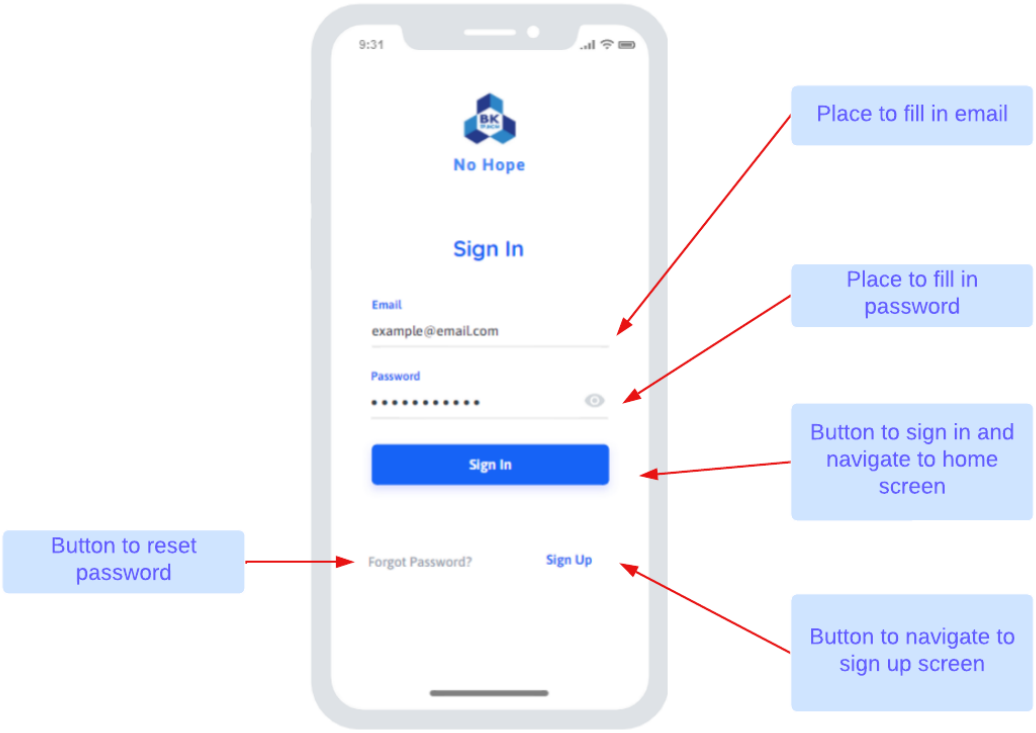


Figure 2: Sign In

In this screen, users will log in to the app by using their own account. They will press 'Sign In' button and they will be navigated to the home screen. If users do not have the account, they can navigate to sign up screen to create an account by clicking 'Sign Up'. We also provide 'Forgot Password?' feature that users can recover their password.

4.3 Home Screen

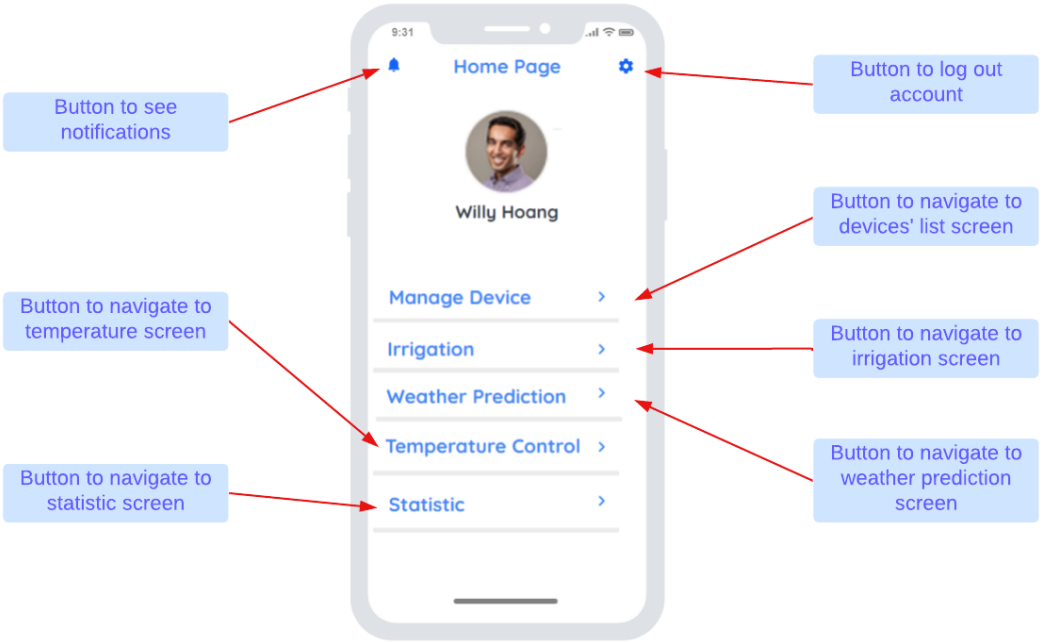


Figure 3: Home Screen

We will have a home screen that user can navigate to manage device screen , irrigation plants screen, weather prediction screen, temperature control screen and statistic screen.

4.4 Irrigation Screen

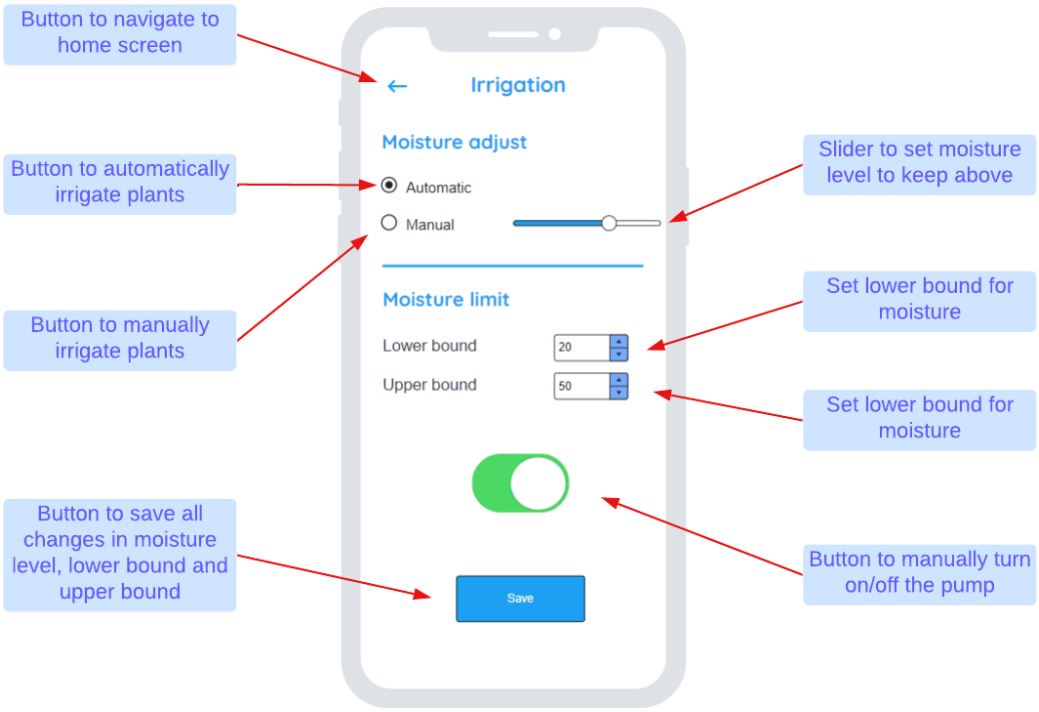


Figure 4: Irrigation page

The user can change from automatic to manual or vice versa by choosing the corresponding button. They can also selects the desired moisture level through a slider for bespoke settings if they choose the auto mode. The user can manually irrigate plants by clicking button if they choose manual mode.

4.5 Report Screen

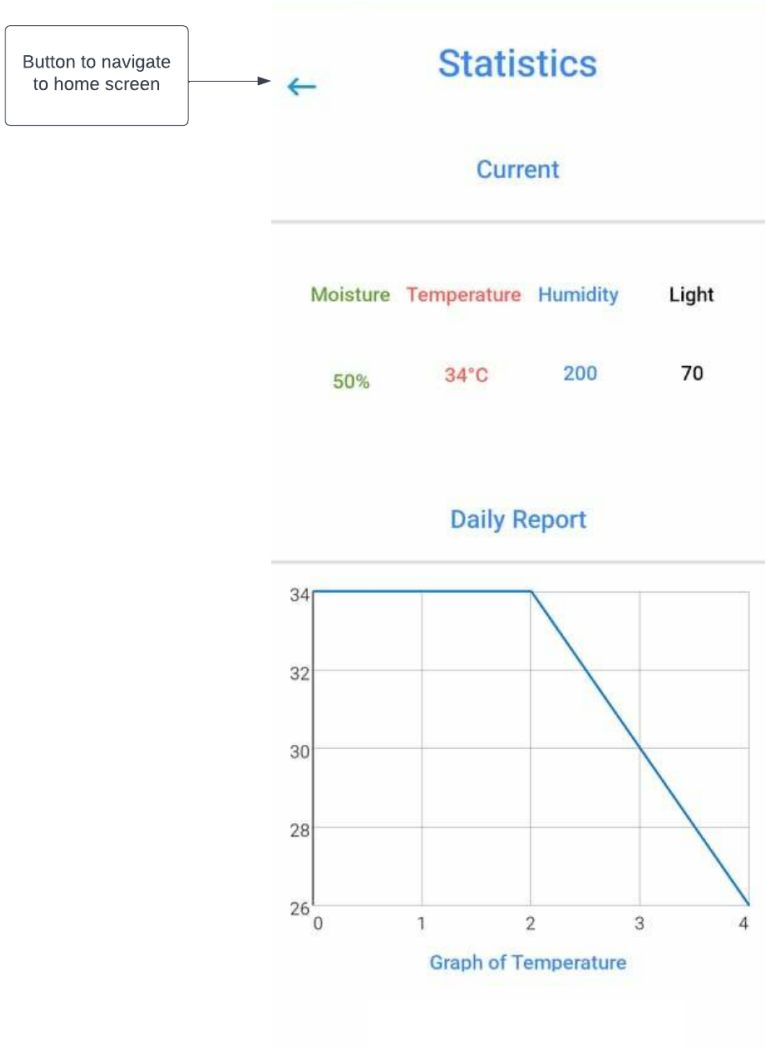


Figure 5: Statistic report

In this screen, user may view the statistic report of four components: temperature, soil moisture, humidity and light. We also provide the line chart for users to view the statistics more easily.

4.6 Device Management Screen

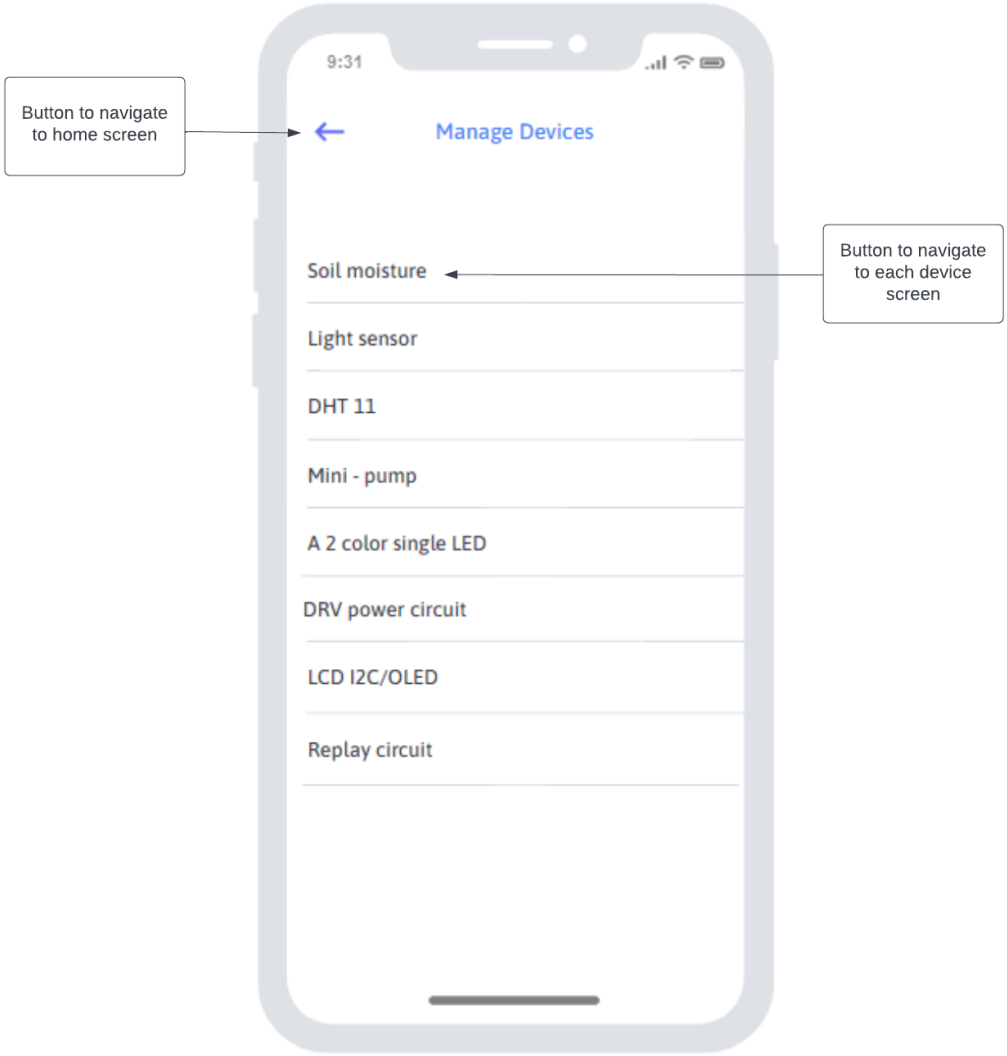


Figure 6: Device Management

We will provide users the device management screen which users can view the current status of each device in the list. If user want find more information about each device, they can tap on the name on the device to be navigated to the corresponding screen.

4.7 Weather Prediction screen

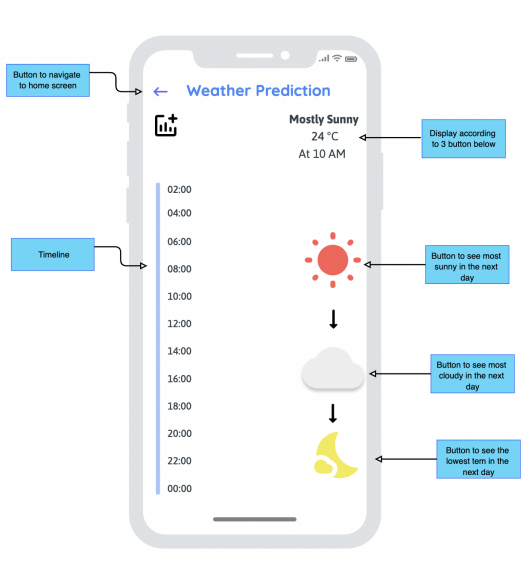


Figure 7: Weather Prediction

From the home screen select Weather Prediction. In this screen,it have the arrow button to go back the home screen and the system will predict the weather of the next day for the users. As can be seen that there is the timeline for the weather ,and they can click on specific time to see what the weather will be like.
On the other hand, people can see more data about our prediction by clicking the icon above the timeline.

4.8 Temperature Control Screen

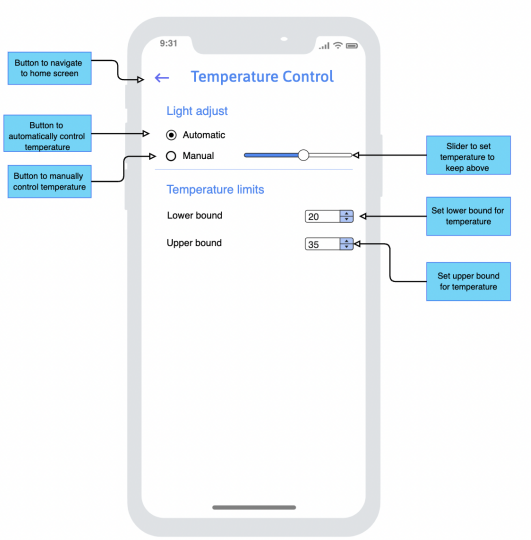


Figure 8: Temperature Control

From the home screen select Temperature Control. In the Temperature Control screen, the layout is similar to some previous screen. It have the arrow button to go back the home screen, light adjust function with automatic mode and manual mode. When user choose Automatic mode the temperature limits will disappear and there is an bar to set temperature level to keep above. If user select Manual mode the sliding bar will be hidden and the temperature limits will appear again, there are two bound for user to input temperature level they wanted.

5 Database Design

5.1 Firebase introduction

In this project, our group decide to use Google Firebase to design database and authentication. Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytic, reporting and fixing app crashes, creating marketing and product experiment.

To develop back-end of our application we use Firebase Real-time Database. The Firebase Real-time Database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our Apple platforms, Android, and JavaScript SDKs, all of your clients share one Real-time Database instance and automatically receive updates with the newest data.

Figure 9: Firebase Real-time Database

5.2 Database Design

Multidisciplinary Project

```
{
    Device: {
        moist_level: integer
        moist_max: integer
        moist_min: integer
        status: string
        temp_level: integer
        temp_max: integer
        temp_min: integer
        temp_mode: bool
        moist_mode: bool
    }
    History: [
        Date: {
            light: integer
            humidity:integer
            temperature: integer
            moisture: integer
            led: boolean
            motor: boolean
        }
    ]
    Users: [
        UserId: {
            Profile: {
                job: string
                name: string
                phone: integer
            }
        }
    ]
    Future: [
        Temp_at_time1: integer
        Temp_at_time2: integer
        Temp_at_time2: integer
    ]
}
```

6 Design Pattern

The Model View Controller (MVC) design pattern specifies that an application consist of a data model, presentation information, and control information. The pattern requires that each of these be separated into different objects.

MVC is more of an architectural pattern, but not for complete application. MVC mostly relates to the UI / interaction layer of an application. You're still going to need business logic layer, maybe some service layer and data access layer.

• The Model contains only the pure application data, it contains no logic describing how to present the data to a user.

• The View presents the model's data to the user. The view knows how to access the model's data, but it does not know what this data means or what the user can do to manipulate it.

• The Controller exists between the view and the model. It listens to events triggered by the view (or another external source) and executes the appropriate reaction to these events. In most cases, the reaction is to call a method on the model. Since the view and the model are connected through a notification mechanism, the result of this action is then automatically reflected in the view.

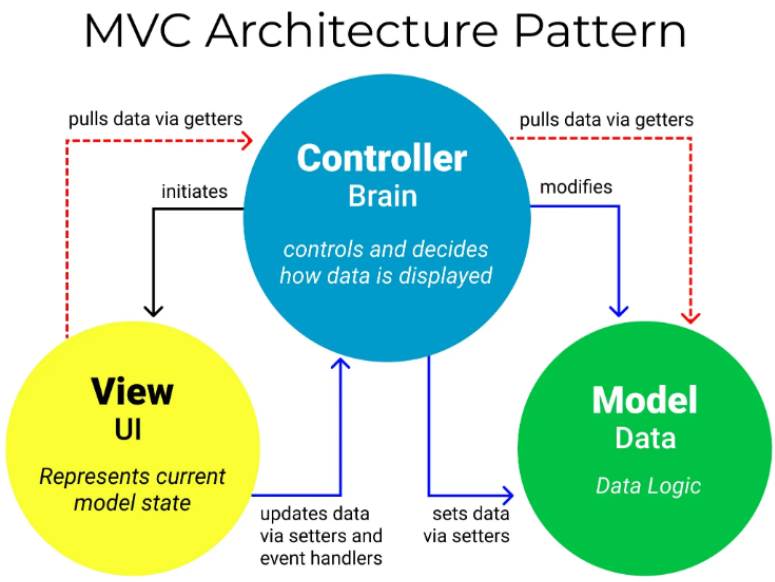


Figure 10: MVC Design Pattern

7 Final result

7.1 Temperature Control

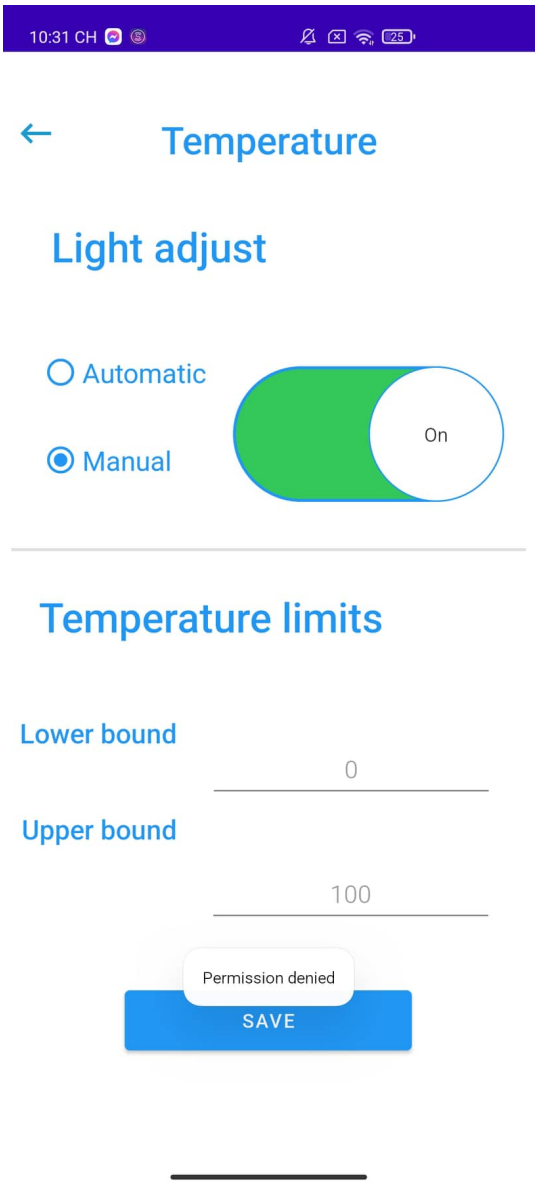


Figure 11: Automatic temperature control

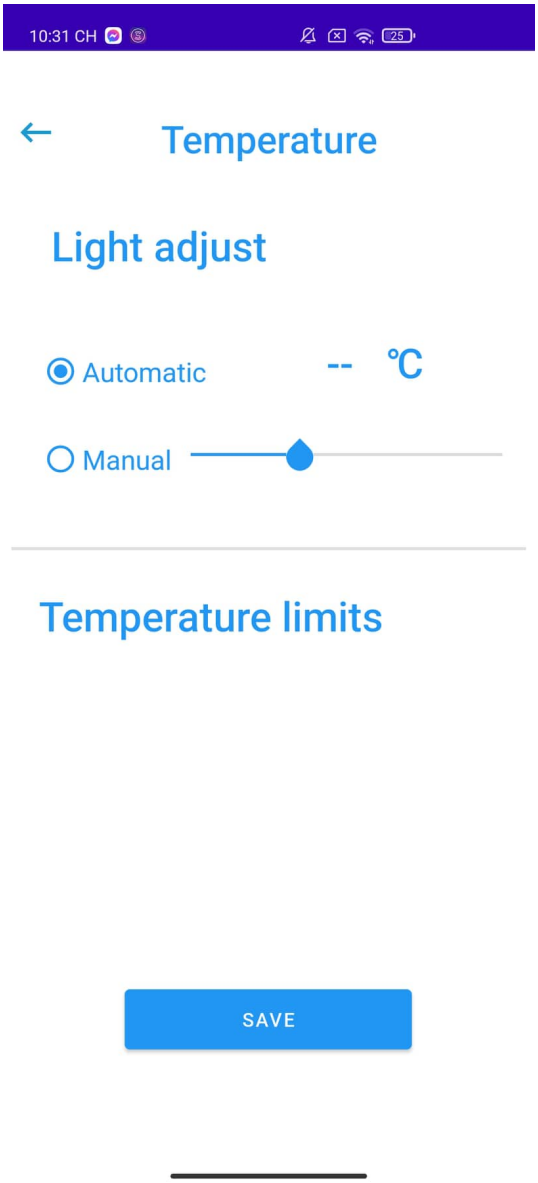


Figure 12: Manual temperature control

7.2 Irrigation Control

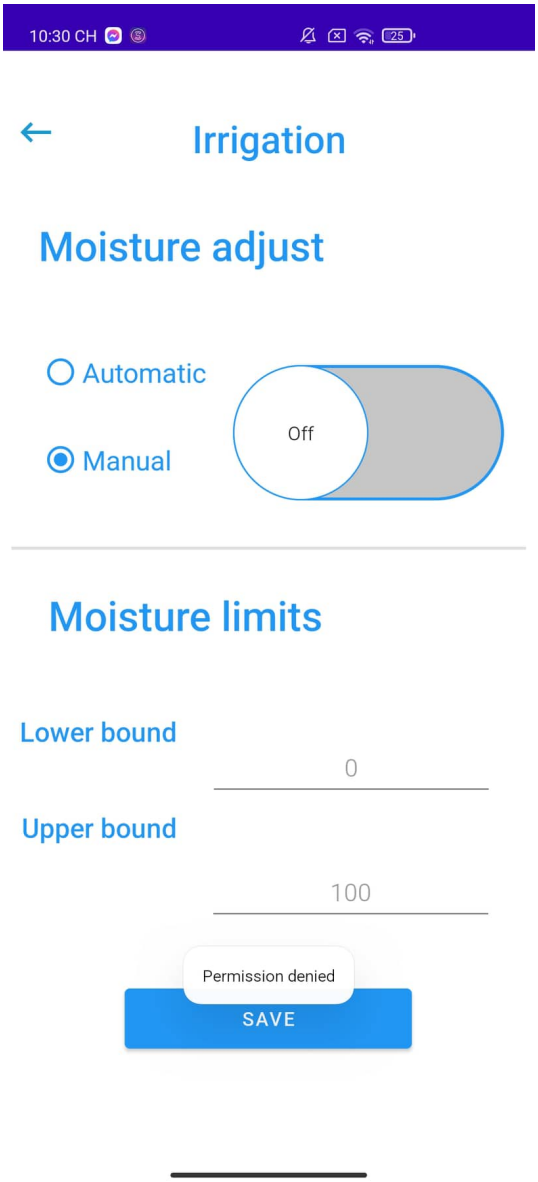


Figure 13: Automatic irrigation control

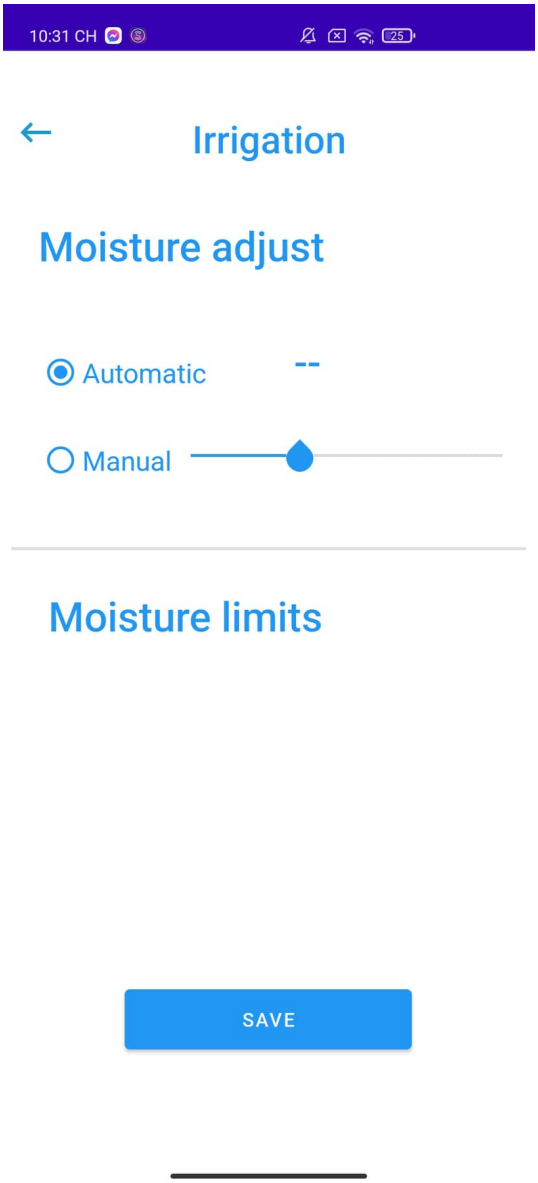


Figure 14: Manual irrigation control

7.3 Statistic and Report

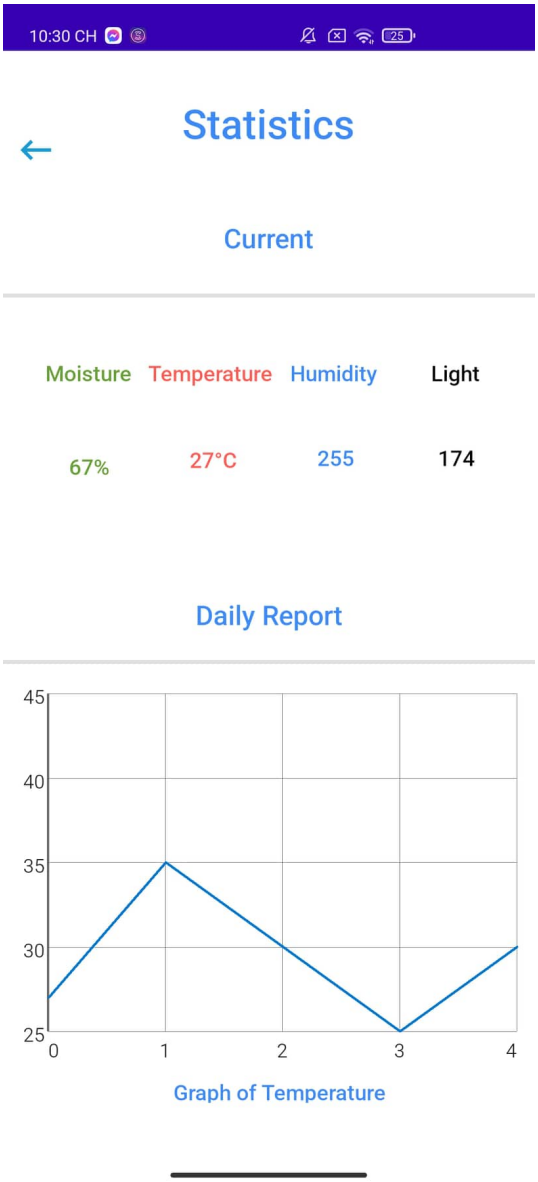


Figure 15: Statistic and Report

7.4 Weather Prediction

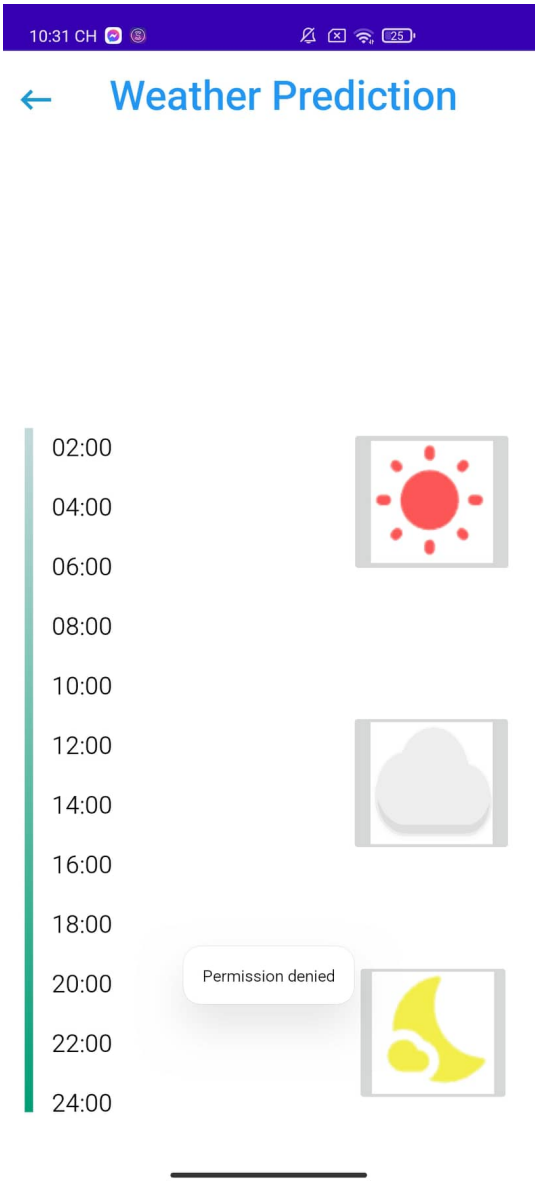


Figure 16: Weather prediction

7.5 Device Management

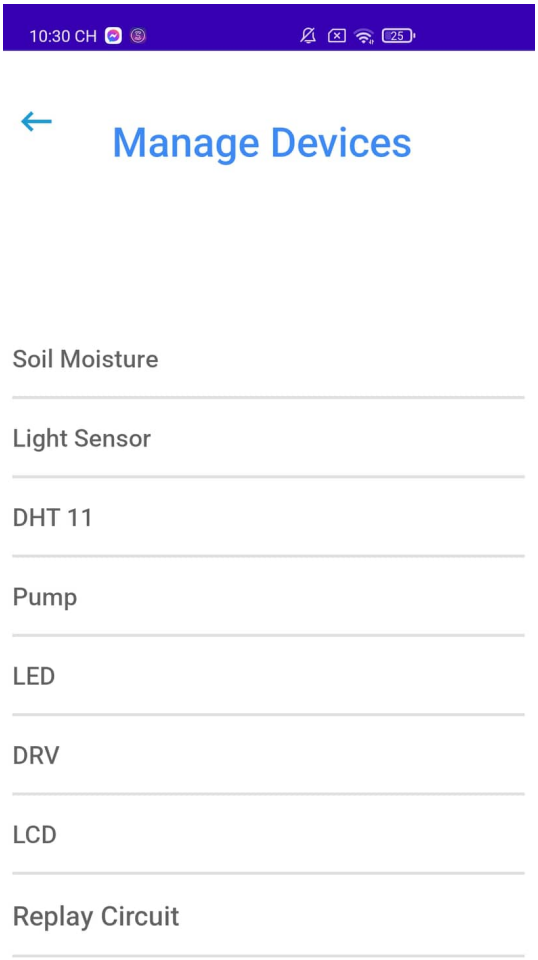


Figure 17: Device management