**Introduction**

A plant’s needs, such as water, sunlight and temperature, must be satisfied for successful plant growth. It is not always apparent which needs are being satisfied and which are not. The purpose of this project is to use sensors to collect important data about the current growing conditions and view this data in real time on a mobile device using a mobile application from anywhere with an internet connection.

**Objective**

To develop a prototype for a remote plant monitoring system that can:
1. Collect plant environment data
2. Display environment data on a mobile device
3. Update and display data in real-time

**Design Overview**

The proposed design consists of three modules:

1. Environment Sensor Module
2. Google Firebase Module
3. Android Application Module

![Figure 1. Visual Representation of Design Modules](image)

**Module Breakdown**

**Environment Sensor Module**
- Soil Humidity Sensor
- Analog-to-Digital Converter
- Raspberry Pi 0

**Google Firebase Module**
- User Authentication
- Firebase Database
- Python script to send data to Firebase
- Integration with Android Application

**Android Application Module**
- Authenticate Users
- Display sensor data to the User
- Get latest data updates from Firebase

**Environment Sensor Hardware & Firmware**

The hardware system was designed to connect the analog moisture sensor to the Raspberry Pi using the Analog-to-Digital Converter (ADC) chip.

![Figure 2. Wiring Diagram](image) ![Figure 3. Prototype Hardware Design](image)

**Hardware Components List**
- Raspberry Pi Zero
- MCP3008 8-channel 10-bit ADC
- Assembled Pi Cobbler Plus Breakout Cable
- Capacitive moisture sensor

**How It Works**

The measurement signal from the capacitive moisture sensor must first be converted to a digital signal using the MCP3008 ADC chip before it can be readable by the Raspberry Pi. In the Pi, the 10-bit signal is converted into a percentage value before it sent it to the Firestore using an internet connection.

**Testing & Validation**

Figure 4 shows the output of the Python script running on the Raspberry Pi with the moisture sensor in the plant as shown in Figure 3 above.

![Figure 4. Raspberry Pi Output](image)

**Android Application**

Android application that enables the user to monitor plant conditions remotely in real-time

**App Architecture**

- Written in Kotlin using Model View ViewModel (MVVM) Architecture [1]
- Fragments and databinding used for responsive UI [2]

![Figure 6. Android Application User Interface](image)

**Future work**

- Design & Manufacture PCB
- Design & 3D print a case for the hardware components.
- Develop data visualization features such as graphing and plotting data in the Android app
- Write an automated testing suite for the app
- Source and integrate sensors to measure other conditions such as temperature and light

**Reference**


![Figure 7. Android App MVVM Architecture](image)