Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2023

PlantWays
SmartPlanter

Samuel J Ruiz
Kevin Flores
Luis Guillermo Jaen
Don Bui
Samrat Baral

Revision History

Revision	Date	Author(s)	Description
0.1	2.01.2023	KF,DB,SR,LJ,SB	document creation
0.2	2.13.2023	KF,DB,SR,LJ,SB	complete draft
1.0	5.10.2023	KF,DB,SR,LJ,SB	Final Version

CONTENTS

List of Figures

List of Tables

1 Introduction

As described in the ADS, the SmartPlanter is a product intended to ease plant maintenance by monitoring the plants environment and automating the watering process. The SmartPlanter will require a mobile application that allows users to pair their SmartPlanter device to the internet. Using the mobile application, the users can set the sensor calibration data for the specific plant they are planting. The SmartPlanter device will be able to monitor the plant's environment using the sensors and the calibration data in order to automate the watering process when the soil is too dry.

In the system requirements specification document, we described how the SmartPlanter's targeted audience are plant beginners and people who neglect plant care. The purpose of this device is for it to be used as a learning tool and to automate plant care. The mobile app's purpose is to provide an easy using monitoring system for your plant.

2 System Overview

There are 3 separate layers to our data flow diagram: Smart Planter, Phone Application, and Cloud. The phone application layer deals with the initial user's set up and inputs, and provides the user with an interface to monitor the SmartPlanter's data. This layer is then connected to the Smart Planter layer via Bluetooth, and then the WiFi connection is established. Finally, the user can send or request data, to and from the database via the cloud layer.

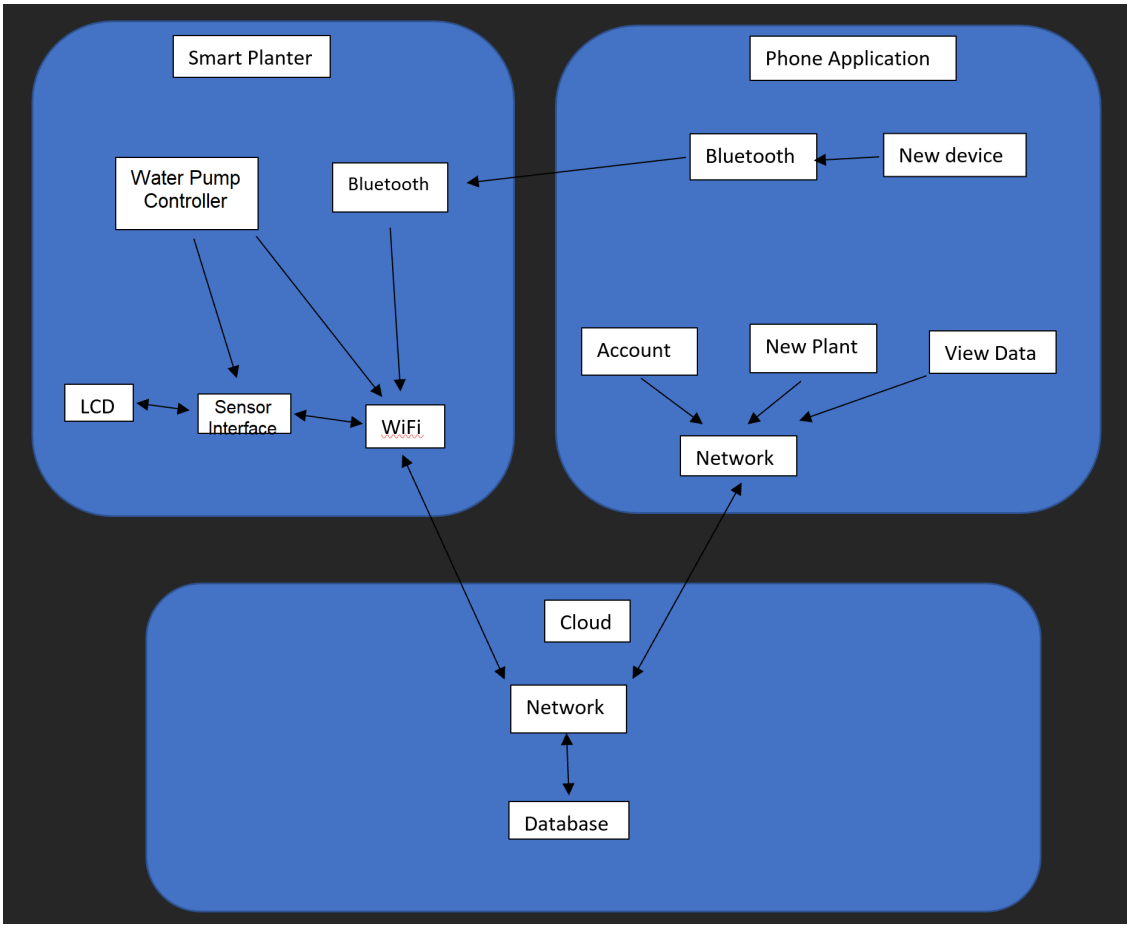

Figure 1: System architecture

3 SmartPlanter Subsystems

3.1 Layer Hardware

The SmartPlanter system is run on an ESP-32 micro-controller. What we use to make the smart planter pot, we use a software that is call Shapr3D to make the 3D model of smart planter pot and we use the Ender 3 3D printer to print out the 3D model. The smart planter pot will have LCD screen in front of the smart planter pot. We use Penpot to make the to make the graphic for the LCD screen that show the information level of the sensor.

3.2 Layer Operating System

3.3 Layer Software Dependencies

The SmartPlanter code is compiled using Arduino IDE and the EspressIf Arduino packages.

3.4 Water Pump Controller

The auto water pump subsystem is in charge of watering the plant and keeping the plant at a sufficient level of moisture. It communicates with the sensors in the planter, receiving the instruction to pump water.

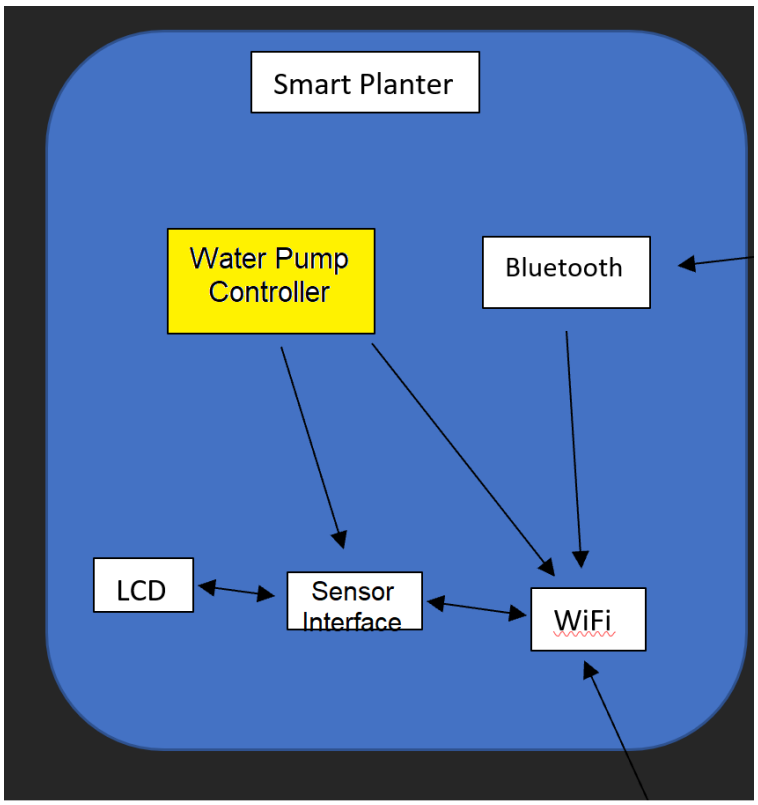

Figure 2: Automatic Water Pump Subsystem Diagram

3.4.1 Subsystem Hardware

The water pump in the SmartPlanter is an: ALAMSCN 3v - 5v micro submersible mini water pump.

3.4.2 Subsystem Operating System

3.4.3 Subsystem Software Dependencies

3.4.4 Subsystem Programming Languages

The code for the Automatic water pump was written in C using Arduino IDE.

3.4.5 Subsystem Data Structures

3.4.6 Subsystem Data Processing

3.5 Sensor Interface

The sensors in the Smart Planter are used to monitor the plant's environment by measuring the temperature, humidity, and soil moisture. When the soil moisture is low, the sensor data will notify the water pump and being the automatic watering. The water level sensor is used to assure the water pump does not run without any water.

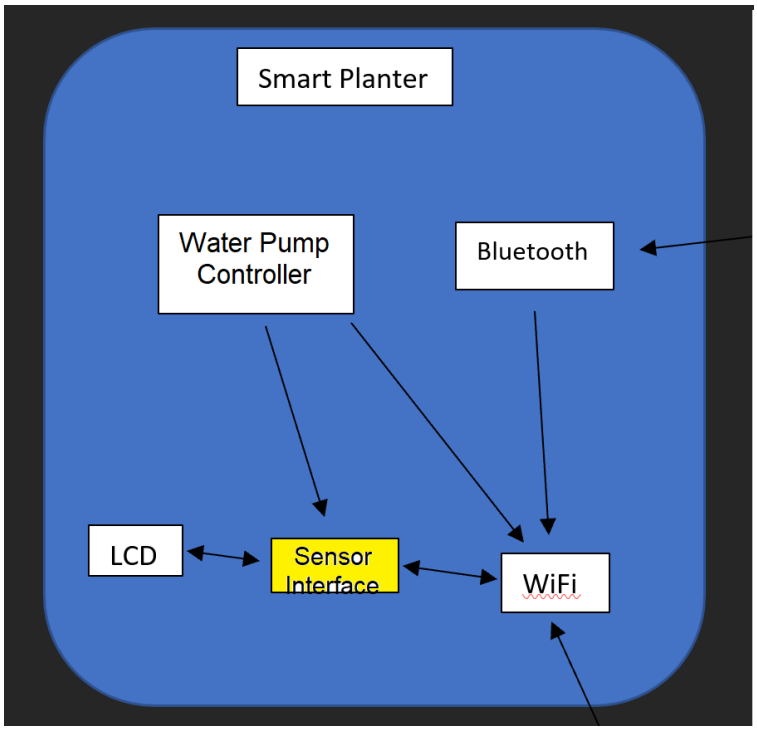

Figure 3: Sensor Subsystem Diagram

3.5.1 Subsystem Hardware

The sensors in the SmartPlanter comprise of the soil moisture is monitored using a Songhe Capacitive Soul Moisture Sensor, the water tank level is monitored using an EPTTECH Food Grade Optical Infrared Water Liquid Level Sensor, and the Temperature + Humidity is monitored using a HiLetgo DHT11 Temperature Humidity Sensor Module.

3.5.2 Subsystem Operating System

3.5.3 Subsystem Software Dependencies

The SmartPlanter DHT11 uses the Arduino DHT Library.

3.5.4 Subsystem Programming Languages

The code for the SmartPlanter Sensors is written in C.

3.5.5 Subsystem Data Structures

Sensor Interface will read in DAC units

3.5.6 Subsystem Data Processing

Data being processes to LCD screen and database will be converted into strings corresponding the sensor status. (Water level sensor DAC of 0 will process "Water Full to database and LCD screen")

3.6 WiFi

The WiFi component of our smart planter deals with receiving all of the data from the external and internal sensors, and then sending that data off to the cloud database.

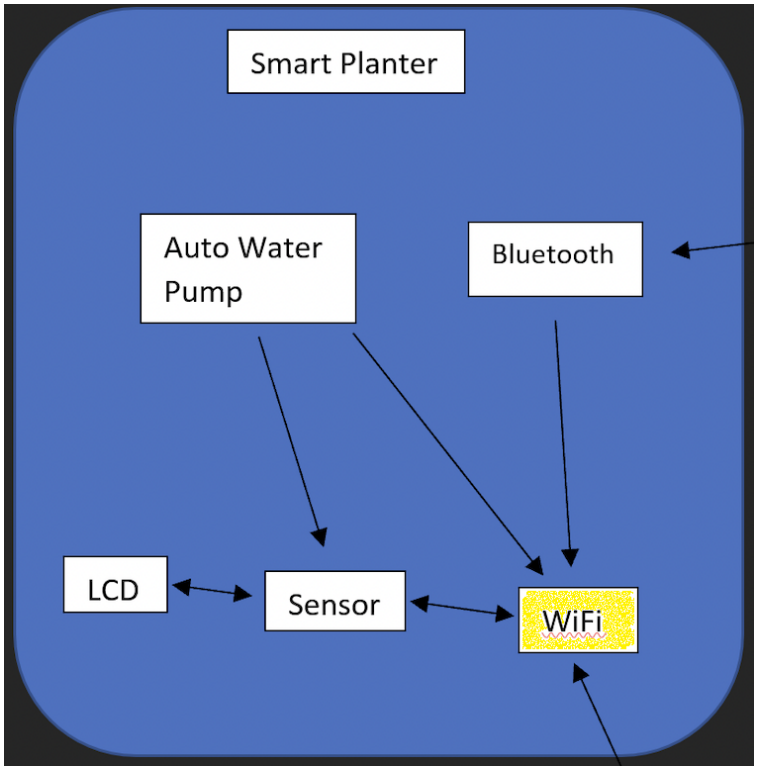

Figure 4: WiFi Subsystem diagram

3.6.1 Subsystem Hardware

The SmartPlanter is able to connect to the Wifi using the ESP-32 built in WiFi module.

3.6.2 Subsystem Operating System

3.6.3 Subsystem Software Dependencies

The set up for the WiFi uses the Arduino ESP-32 WiFi Manager library.

3.6.4 Subsystem Programming Languages

The code for the SmartPlanter WiFi was written in C.

3.6.5 Subsystem Data Structures

SmartPlanter sensor data will be automatically inserted into MangoDB which allows for the phone application to retrieve sensor data.

3.6.6 Subsystem Data Processing

Data send to the database from the ESP-32 will be in strings assigned in the sensor interface

3.7 Bluetooth

The Bluetooth system is there to establish a connection between the smart planter and the network, via your smart phone. The Bluetooth communicates with the WiFi subsystem.

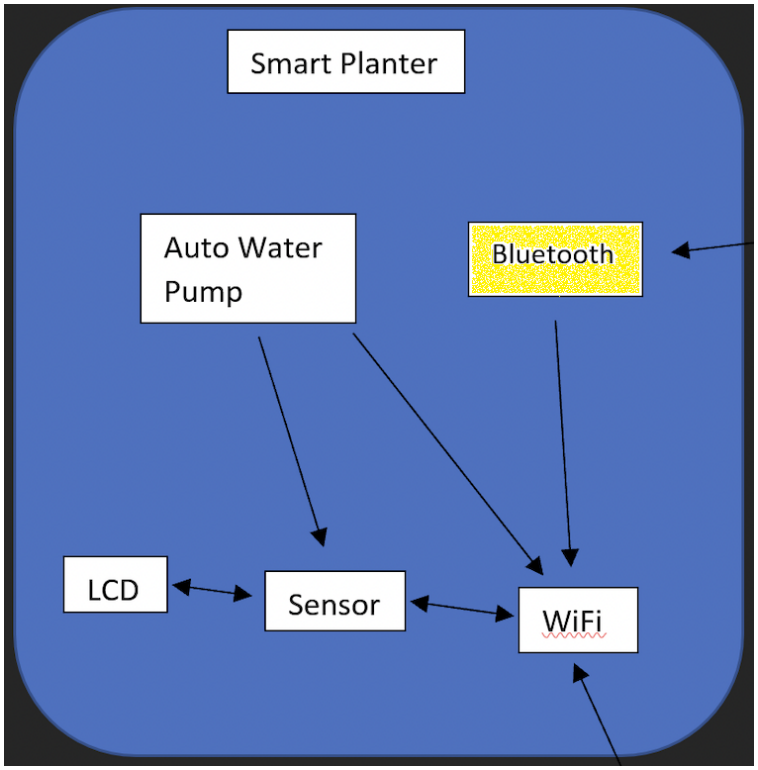

Figure 5: Bluetooth Subsystem Diagram

3.7.1 Subsystem Hardware

The SmartPlanter is able to connect to the Wifi using the ESP-32 built in Bluetooth module.

3.7.2 Subsystem Operating System

3.7.3 Subsystem Software Dependencies

The set up for the Bluetooth uses the Arduino ESP-32 Bluetooth Serial library.

3.7.4 Subsystem Programming Languages

The code for the SmartPlanter Bluetooth was written in C.

3.7.5 Subsystem Data Structures

Data will be transmitted between the phone and device using Bluetooth serial.

3.7.6 Subsystem Data Processing

Bluetooth serial data will process JSON data between the app and the ESP-32

3.8 LCD Screen

The LCD screen is used to display either the current time or the name of your plant along with different screens showcasing sensor data.

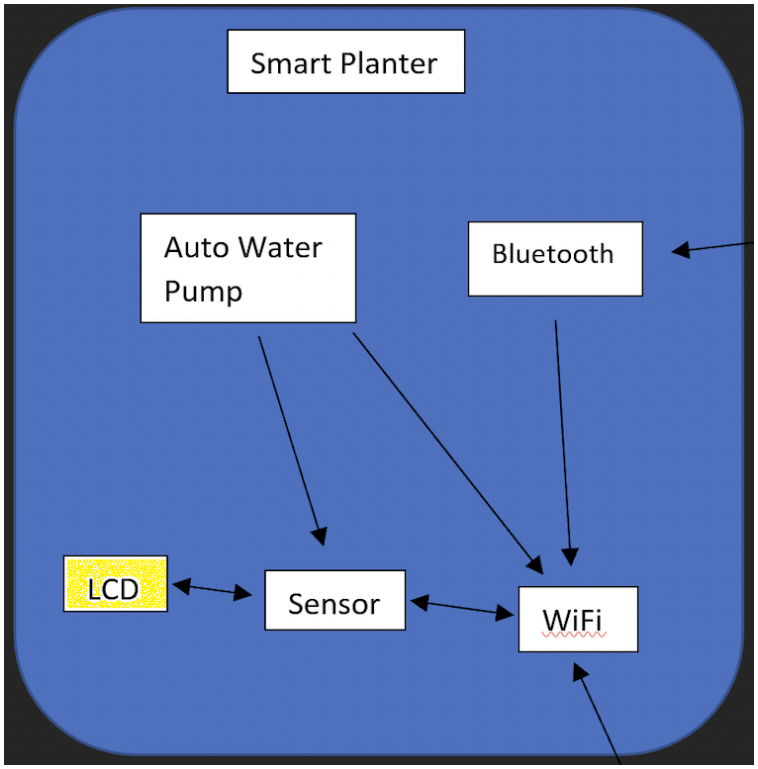

Figure 6: LCD Screen Subsystem Diagram

3.8.1 Subsystem Hardware

The LCD Screen of the SmartPlanter uses a 1.28 inch Round LCD Display Module 240x 240

3.8.2 Subsystem Operating System

3.8.3 Subsystem Software Dependencies

The LCD screen uses the Arduino Esp32 TFT library to display images and text on the screen.

3.8.4 Subsystem Programming Languages

The code for the SmartPlanter LCD Screen was written in C.

3.8.5 Subsystem Data Structures

LCD screen will receive temperature in Fahrenheit, humidity in percentage, and the rest in strings (moisture sensor, light sensor, water tank liquid sensor) from the sensor interface.

3.8.6 Subsystem Data Processing

The LCD displays string data retrieved from the sensor interface

4 User Interface Subsystems

The user interface system or the phone application system includes subsystems of information that the user is able to observe such as their account or functional subsystems where the user can perform actions such as adding a plant. The layer may seem that it may not have that many subsystems but the purpose of the application layer is for the user to just setup their pot and use the application to be able to see information about it whenever they want.

4.1 Layer Hardware

This layer is purely software the only requirement is to have a mobile device does not matter what type of operating system. Further specifics of the operating system are discussed below.

4.2 Layer Operating System

Since we are using Flutter to develop our user interface it is compatible with multiple operating systems such as android, iOS, Windows and mac OS. Since we are building a mobile app the supported platforms for this layer are android 4.1 and above and iOS 11 and above anything below that Flutter will not be able to run on the OS.

4.3 Layer Software Dependencies

This user interface layer depends entirely on the flutter framework and Dart programming language. Make connection using library of non-sequel database (MongoDB Database) mongo_dart version (0.5.1).

4.4 New Device Subsystem

The new device subsystem allows the user to connect to the smart plant pot via bluetooth. Once the connection with bluetooth is establishes, the New Device interface relays WIFI credentials through bluetooth and receives the mac address of the plant.

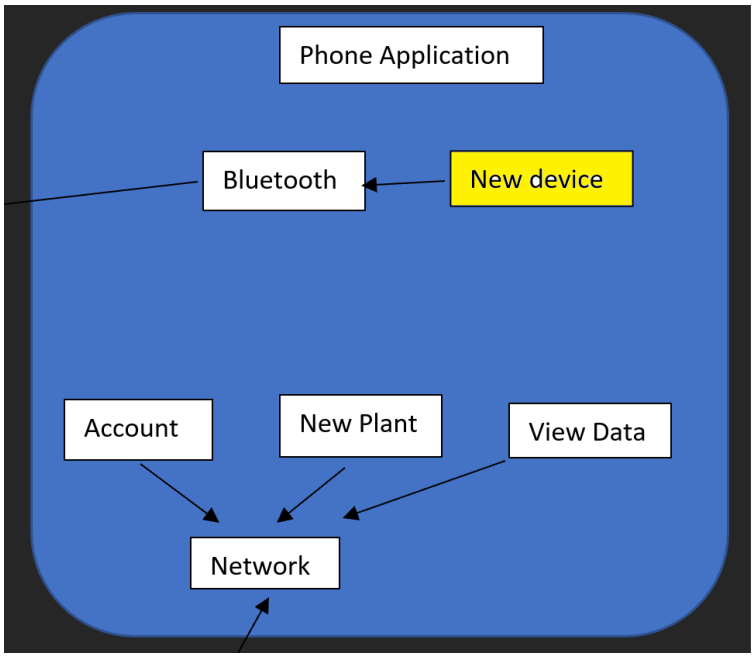

Figure 7: New Device Subsystem Diagram

4.4.1 Subsystem Software Dependencies

Since this subsystem is used to connect the device to the network it requires the flutter_blue version 0.8.0 package.

4.4.2 Subsystem Data Structures

Data sent through bluetooth from the new device interface consist of wifi credentials encoded in JSON. The MAC address returning from bluetooth will also be in JSON.

4.4.3 Subsystem Data Processing

Data sent and received from the bluetooth interface will be serialized/unserialized in JSON in the new device interface.

4.5 Bluetooth Subsystem

The bluetooth subsystem works as a bridge by establishing a connection used to connect the phone application's new device interface and the smart planter hardware bluetoot.

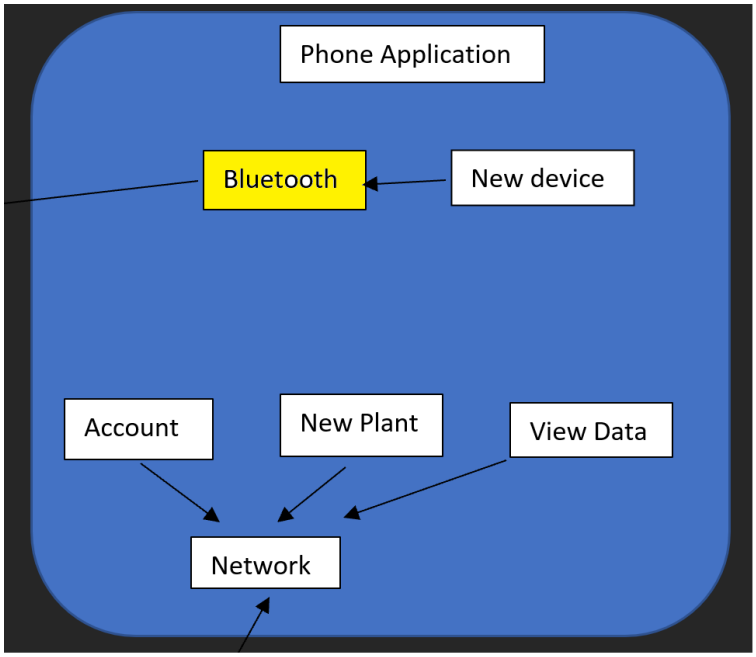

Figure 8: Bluetooth Subsystem Diagram

4.5.1 Subsystem Operating System

For both iOS and Android we need permissions to use Bluetooth and access location

4.5.2 Subsystem Software Dependencies

The package flutter_blue version 0.8.0 should be imported in order to implement Bluetooth within the application. flutter_blue is only compatible from version 19 of android SDK.

4.5.3 Subsystem Data Structures

Data structures passed through the Bluetooth serial interface will be serialized JSON data.

4.5.4 Subsystem Data Processing

No data processing

4.6 View Data Subsystem

The purpose of the view data subsystem is only for the user to observe sensor readings with pots it has, also has additional functionality such as deleting a pot. The sensor readings will be read from the database in time intervals and display them to the user.

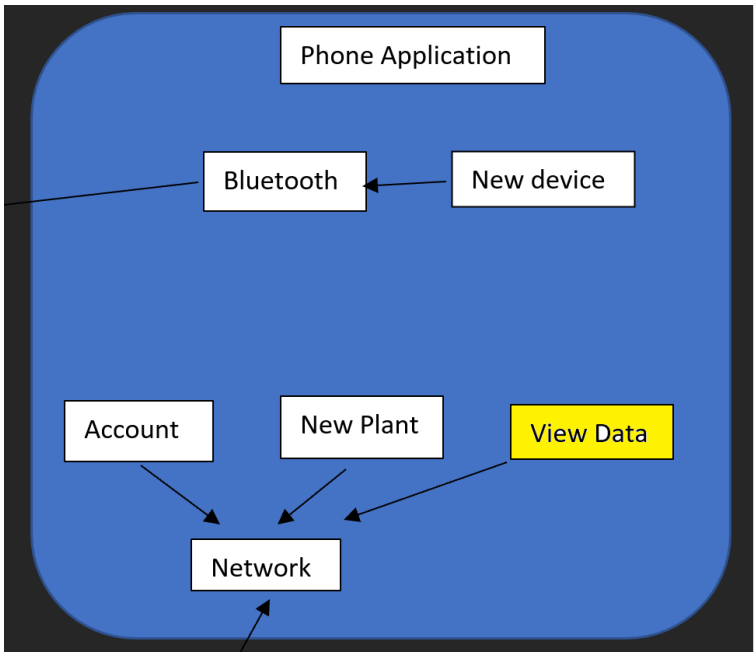

Figure 9: View Data Subsystem Diagram

4.6.1 Subsystem Software Dependencies

The package flutter/material.dart is a dependency in order to create the UI for the view data page. View data also depends on database connection which is done by interacting with the network to pull data.

4.6.2 Subsystem Programming Languages

The user interface is being done with the flutter framework.

4.6.3 Subsystem Data Structures

View data interface receives data from the database through a network connection. Data received as strings.

4.6.4 Subsystem Data Processing

String data from the network is displayed on the view data user interface.

4.7 New Plant Subsystem

The new plant subsystem is a component that will allow the user to edit the plant that is assigned to the smart planter instance. This works by retrieving plant classes from the database through the network connection and displaying them on the UI. When the user selects add new plant to database, the class they retrieved is pushed on to their database account in the section for their corresponding plant pot.

4.7.1 Subsystem Software Dependencies

The package flutter/material.dart is a dependency in order to create the UI for the new plant page. New plant also depends on a database connection to the cloud in order to edit plant settings

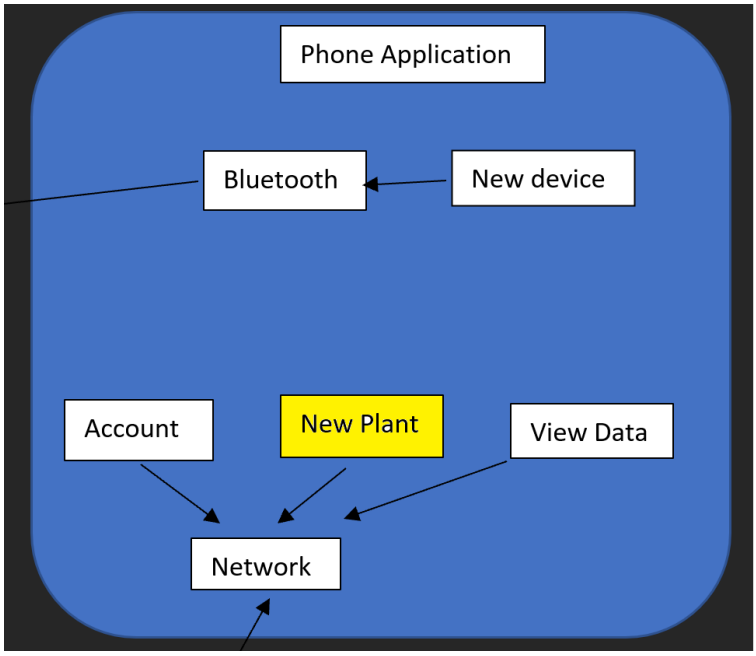

Figure 10: New Plant Subsystem Diagram

4.7.2 Subsystem Programming Languages

The user interface is being done with the flutter framework.

4.7.3 Subsystem Data Structures

Plants information send and received as classes

4.7.4 Subsystem Data Processing

Data sent and received through the network as classes

4.8 Account Subsystem

The account subsystem allows the user to change any settings they are allowed and also view any account information.

4.8.1 Subsystem Programming Languages

The user interface is being done with the flutter framework.

4.8.2 Subsystem Software Dependencies

The package flutter/material.dart is a dependency in order to create the UI for the new plant page. Depends on database connection.

4.8.3 Subsystem Data Structures

Data send and received in flutter string

4.8.4 Subsystem Data Processing

Data retrieved from the database is displayed on the UI as strings, and data sent to the database when the user changes the information is saved as a database entry string

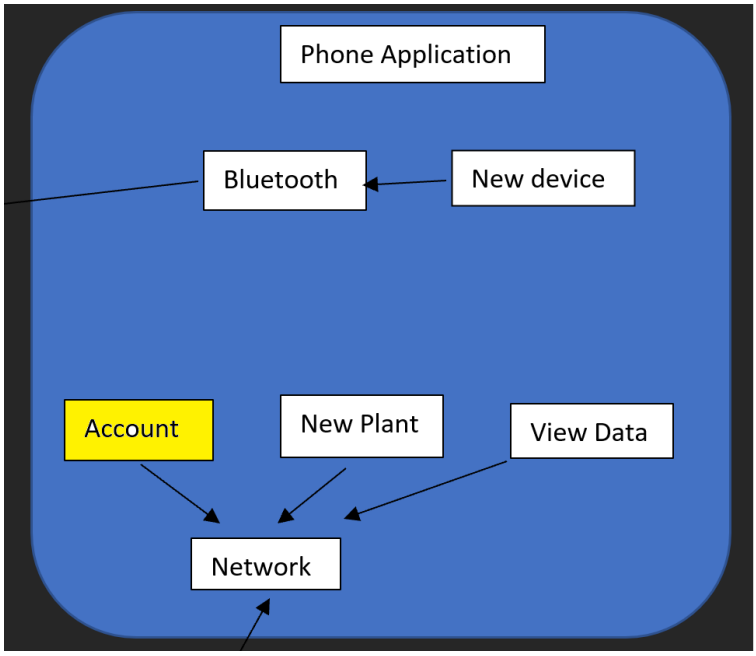

Figure 11: Account Subsystem Diagram

4.9 Network Subsystem

The network is in charge of the communication between the non-SQL database(MongoDB Database), SmartPlanter application(Mobile Application) and the SmartPlanter system(Smart Planter Pot). This layer is automated system which means that its a virtual representation of the network connection between mobile application and smart planter using their WiFi and Bluetooth connection that connects to internet and update the database.

4.9.1 Subsystem Programming Languages

Realm is implemented in dart

4.9.2 Subsystem Software Dependencies

MongoDB with Atlas Device Sync and Realm

4.9.3 Subsystem Data Structures

Have to create Object Schemas in dart to let Realm know the structure of the data being sent.

4.9.4 Subsystem Data Processing

Realm uses Atlas Device Sync to synchronize data between MongoDB and the phone application.

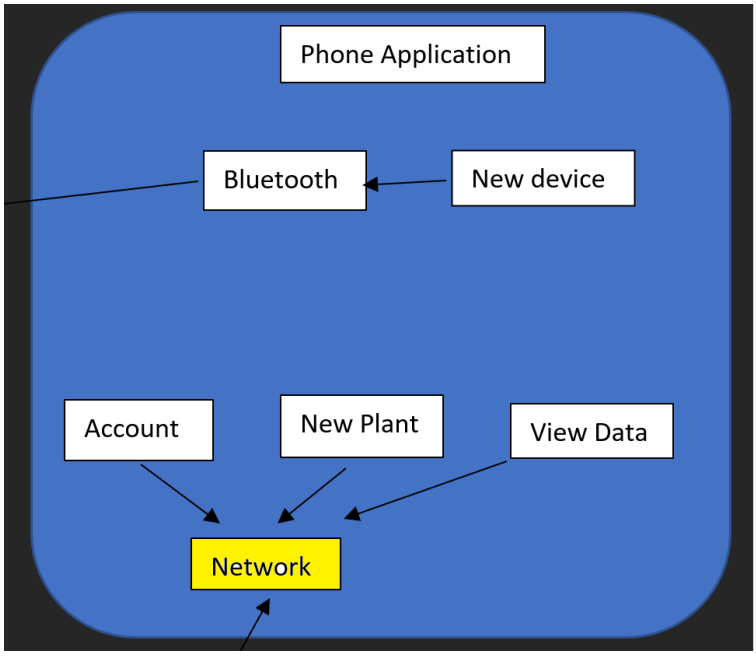

Figure 12: Network Subsystem Diagram

5 Cloud Subsystems

The smart planter layers is made up with network and the database, the could will communicating between the database throw the network connecting to the smart planter pot system. The smart planter pot will send data from the sensor throw the network connected to the database to be store.

5.1 Network

The network is in charge of the communication between the non-SQL database(MongoDB Database), SmartPlanter application(Mobile Application) and the SmartPlanter system(Smart Planter Pot). This layer is automated system which means that its a virtual representation of the network connection between mobile application and smart planter using their WiFi and Bluetooth connection that connects to internet and update the database.

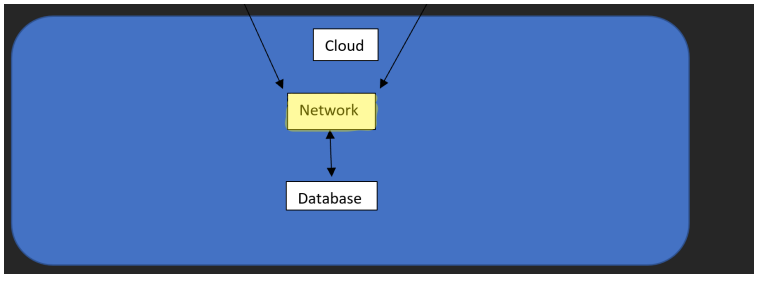

Figure 13: Network Subsystem Diagram

5.1.1 Subsystem Programming Languages

Realm is implemented in dart

5.1.2 Subsystem Software Dependencies

MongoDB with Atlas Device Sync and Realm

5.1.3 Subsystem Data Structures

Have to create Object Schemas in dart to let Realm know the structure of the data being sent MongoDB will also create the same schemas to update data in the collections with that same format.

5.1.4 Subsystem Data Processing

ealm uses Atlas Device Sync to synchronize data between MongoDB and the phone application.

5.2 Database

The cloud database is server-less. It needs no version control. Cloud database stores user information and plant information.

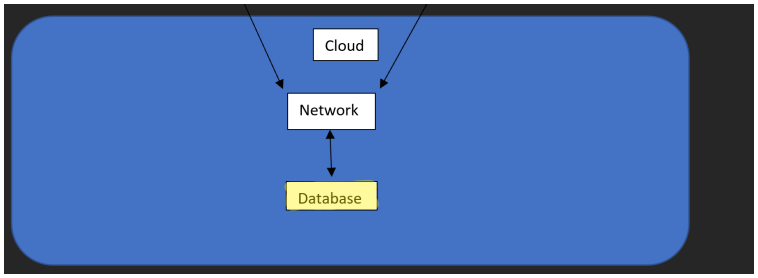

Figure 14: Database Subsystem Diagram

5.2.1 Subsystem Programming Languages

MongoDB is an Open Source database written in C++. But the database is automated and supports multiple programming languages. To implement realm you can use the dart programming language.

5.2.2 Subsystem Software Dependencies

MongoDB and Realm are needed in this subsystem in order to transfer data.

5.2.3 Subsystem Data Structures

MongoDB offers a Document oriented storage, It is support dart programming languages and its library which can help storing data in collections even without defining the structure you can create a collections. MongoDB supports neither joins nor transactions. However, MongoDB does provide atomic operations on a single document.

5.2.4 Subsystem Data Processing

MongoDB supports rich query to fetch data from the database, replacing an entire document (database) or some specific fields with it's update() command, replacing an entire document (database) or some specific fields with it's update() command.

6 Appendix A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

References