

SYSC3010

Project Design

**Last Revised**: October 28, 2019

**Group**: W11

**Students**: Ahmed Sakr (101018695)

Ashfaqul Haque(101037577)

Valerie Figuracion (100965536)

Table of Contents

[Table of Figures 2](#_Toc23157642)

[Problem Overview 3](#_Toc23157643)

[System Architecture 4](#_Toc23157644)

[Plant Endpoints 5](#_Toc23157645)

[Central Processing Server (CPS) 5](#_Toc23157646)

[Android Application 5](#_Toc23157647)

[Communication Protocols 6](#_Toc23157648)

[Registering a plant endpoint or an android user with the server 6](#_Toc23157649)

[Sensory Data Collection and Transmission 7](#_Toc23157650)

[Software 8](#_Toc23157651)

[Class Diagram 8](#_Toc23157652)

[Packets 9](#_Toc23157653)

[Application 9](#_Toc23157654)

[Hardware 9](#_Toc23157655)

# Table of Figures

[Figure 1: Benefits of Indoor Plants [1] 3](#_Toc23163784)

[Figure 2: SmartGrow System Architecture Deployment Diagram 5](#_Toc23163785)

[Figure 3: Registration sequence of a plant endpoint or android user 7](#_Toc23163786)

[Figure 4: Sensory information collection and transmission 8](#_Toc23163787)

[Figure 5: Class Diagram for Smart Grow 9](#_Toc23163788)

# Problem Overview

Increased productivity, enriched health and well-being and decrease in stress are just some of the benefits of growing your own indoor plants. Indoor plants provide a significant boost to the indoor living-quality of our lives by bringing the fresh outdoor air into our homes. Research has shown that 82% of people who worked with plants and windows around them felt content or very happy. Indoor plants are also known to improve the air quality of our homes. A study conducted by Jeanna Bryner shows that rooms with indoor plants contain up to 60 percent fewer airborne molds and bacteria than rooms without plants. Listed are some additional benefits of growing indoor plants.

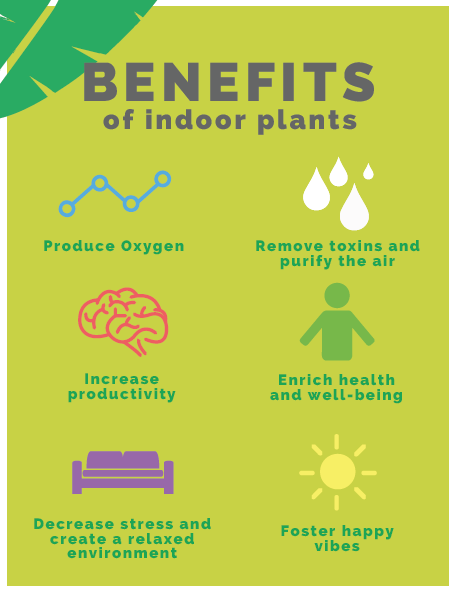


Figure 1: Benefits of Indoor Plants [1]

Maintaining and growing indoor plants is a difficult task for us to balance with our work, school and social life. It is also very difficult to monitor growing conditions of the plant such as the air humidity, soil moisture etc in order to take perfect care of our plants. The SmartGrow makes it easier for the user to grow multiple plants in our homes by:

* Automatically watering the plant by tracking the soil moisture
* Adjusting the light environment by tracking the light intensity of the room
* Providing feedback on the air temperature and air moisture of the plant
* Providing customized preferred growing conditions for a large range of plants
* Monitoring growing conditions of the plants viewable through a smartphone application

# System Architecture

The SmartGrow System is composed of three major entities: **The SmartGrow Application**, **The Central Processing Server**, and **Plant Endpoints.** Every major entity is composed of smaller subsystems that interact with each other to fulfill their necessary function. Figure 2 (shown below) provides a top-to-down breakdown of the SmartGrow system.

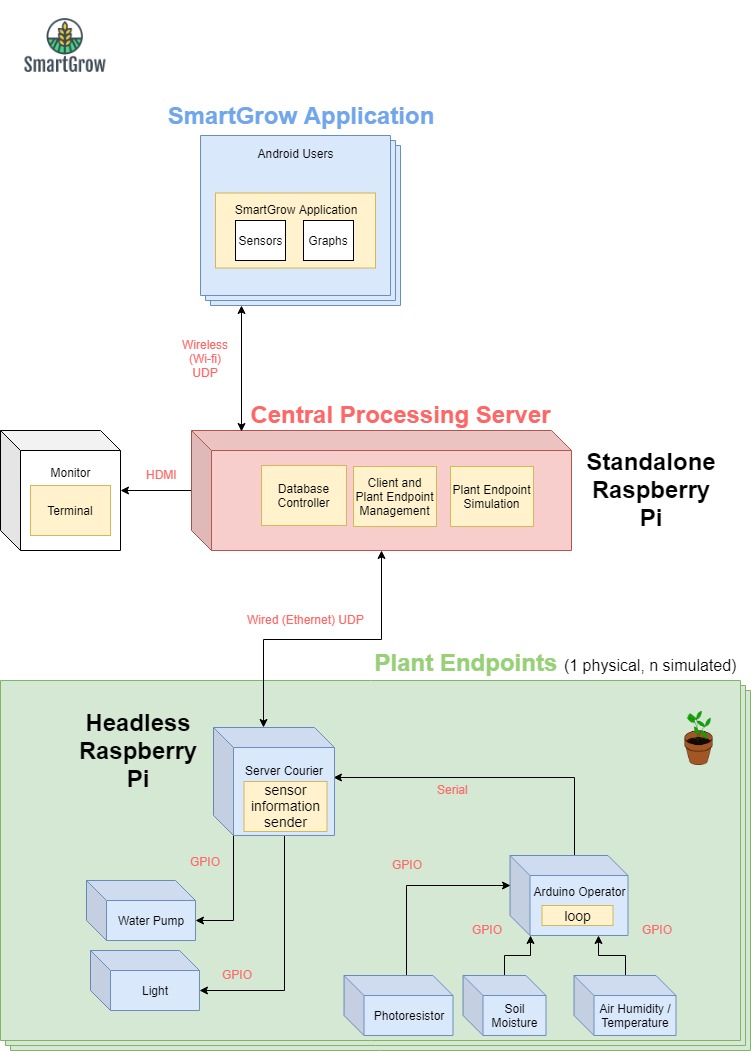


Figure 2: SmartGrow System Architecture Deployment Diagram

## Plant Endpoints

A plant endpoint is a system that is spatially located on the target plant to allow for sensory data collection. The endpoint system consists of a **Server Courier**, **Arduino Operator**, and **sensors**.

* **Server Courier**
  + Sends and receives packets to/from the central processing server.
* **Arduino Operator**
  + Polls all sensors through periodic loops and transmits the data over serial to the courier.
  + Activates the water pump if requested by the Courier
* **Sensors**
  + Environment-gathering sensors, including a Light Intensity Sensor, Soil Moisture Sensor, Air Humidity and Temperature Sensor.
  + A water pump to funnel water to the plant when needed
  + A light source to illuminate the plant when environmental illumination is low

## Central Processing Server (CPS)

The CPS enables the flow of data from end-to-end, compares sensor data to reference data (from the database), and sends data over wireless to the appropriate clients for displaying on the android application.

In order to achieve the previously mentioned functionality, the CPS will divide the work into three parallel entities:

* **Client and Plant Endpoint Management**
  + Retrieves sensor information from all plant endpoints
  + Transmits sensor information to the android users
  + Informs plant endpoints when to turn on water pumps and light sources
* **Endpoint Simulation**
  + Comprehensive simulation of a plant endpoint to demonstrate the scalability of the SmartGrow system
* **Database Controller**
  + Stores and retrieves android client profiles
  + Stores sensory information retrieved from plant endpoints

## Android Application

The android application is the client Graphical User Interface (GUI) for viewing the data flow in the system. Using the application, clients will be able to:

* Specify the plant type that is being monitored
* View real-time sensory information from the plant endpoint
* View graphs of sensor information over a specified period

# Communication Protocols

This section will provide a detailed behavioral understanding into how the previously described subsystems interact to fulfill the system functions. The behavioral understanding will be achieved through **sequence diagrams** that make clear how different actors interact with each other.

## Registering a plant endpoint or an android user with the server

Plant endpoints and android users are treated as **dynamic nodes** by the central processing server. In other words, the central processing server does not know about a node until it registers with the server when it comes up.

Figure 3 depicts the registration sequence of a plant endpoint or an android user.

* The **Leaf** class is the logical classification of a dynamic node (either a plant endpoint or an android user).
* The **DedicatedLeafServicer** class is a DatagramSocket that is created in response to the registration request, and it is responsible for fulling all communication with that leaf during its lifetime.

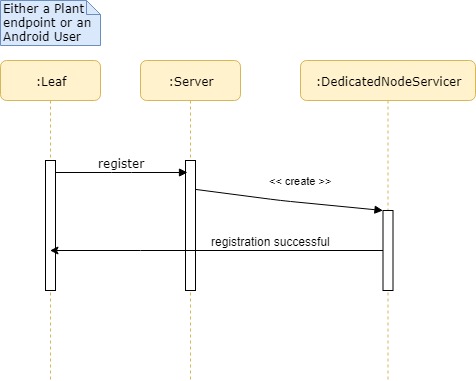


Figure 3: Registration sequence of a plant endpoint or android user

## Sensory Data Collection and Transmission

A critical function in the SmartGrow system is the ability to extract sensory information from plant endpoints and relay the data to the central processing server. As exposed by the system architecture deployment diagram, the extraction of sensory information until it is received by the server involves the interaction of a few subsystems.

Figure 4 depicts the sensory information sequence that easily portrays the involvement of all subsystems.

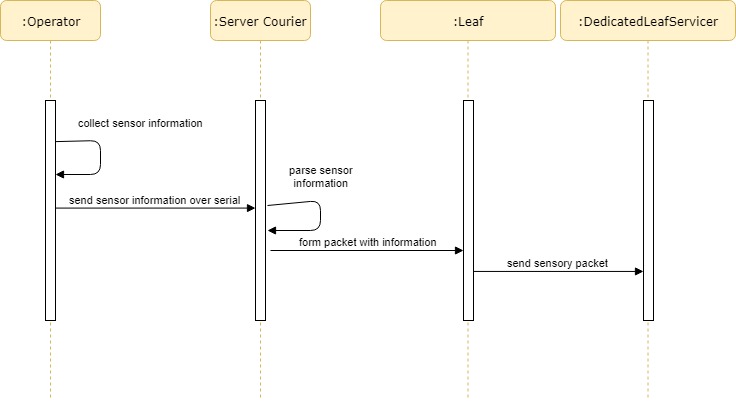


Figure 4: Sensory information collection and transmission

Table 1 summarizes the interactions between different actors in terms of data exchange.

Table 1: Itemized details of message passing for sensory information collection and transmission

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor 1** | **Actor 2** | **Message** | **Data** |
| Operator | Operator | Collect sensor information | Digital values from sensors through GPIO |
| Operator | Server Courier | Sensor information over serial | A byte array of sensors information |
| Server Courier | Server Courier | Parse sensor information | Extract information from byte array |
| Server Courier | Leaf | Form packet | UDP Datagram with sensor information |
| Leaf | DedicatedLeafServicer | Send sensory packet | Finalized packet |

# Database

A screenshot of a cell phone

Description automatically generated

Figure 5:Database schema for SmartGrow

# Software

Below is an overview of the diagram of the classes for SmartGrow. Each classes can be easily grouped together in terms of their functionality.

A screenshot of text

Description automatically generated

Figure 6: Class Diagram for Smart Grow

## Network

The SmartGrow network consists of three major components **stem, branch,** and **leaf**.

* A **leaf** is a process that listens to a specific port. It is responsible for sending data from the port to other leaves and receiving data from other leaves to extract.
* The **branch** groups different but similar leaves together. This allows a message meant for a specific set of leaves to be broadcasted to said group.
* The **stem** connects all the components of the SmartGrow system. This includes but is not limited to the Central Processing Server and the Plant Endpoints. It is the layer that allows the transport of information from one leaf to another leaf.

The leaf and stem extends the transport method for ease of communication.

## Packets

Packets are the protocol for communication between leaves that are in a fixed size of **512 bytes**. The packets start with an opcode which takes up one byte. The opcodes are contained in the OpCodes class. The packet end with a four-byte Cyclic Redundancy Check (CRC) to detect errors. Such errors include incompatible packet size, corrupt packet data, and incorrect or unrecognized opcodes.

### Errors

Each packet will be checked by custom exceptions. CRCVerificationException will check if the CRC is what is expected as described previously. OpCodeNotRecognizedException will check if the first byte is what is expected. Both are extended by CorruptPacketException. TransportInterruptedException will check to see if the data in transit has been blocked or rejected due to the socket being closed.

## Application

A picture containing text

Description automatically generated

Figure 7: Flowchart of actions for the application

# Hardware

The main component of the SmartGrow is an embedded system consisting an Arduino Uno and 3 sensors. The soil moisture sensor will be inserted in the plant soil, relatively close to the water pump, in order to be able to tell the soil moisture of the plant. The soil moisture will be connected to the 5V power supply of the Arduino and will get input though the analog (A0) port of the Arduino. The air quality sensor will be responsible for obtaining the air temperature and the air humidity of the surrounding environment of the plant. The air quality sensor will also get power from the 5V power supply of the Arduino and will get its input thought the digital pin (Port 4) of the Arduino. The Light Dependent Resistor (LDR) is responsible for obtaining the light intensity of the environment surrounding the plant. It is connected to a 10K Ohm pull-up resistor to avoid any interference from external signals. It will also get power from the 5V Arduino power supply and will be connected to the analog (A1) port.

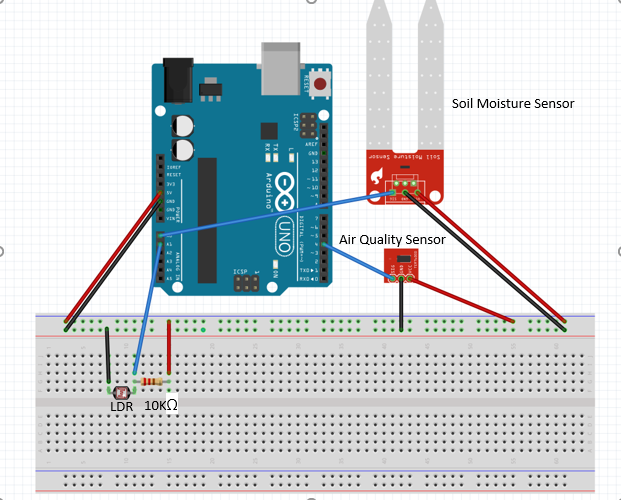


Figure 8: Sensors attached to the Arduino