# Final Project: Smart Waste Management System

|  |  |  |  |
| --- | --- | --- | --- |
| **PROGRAM:** | 420.B0 Computer Science Technology | | |
| **COURSE:** | Internet of Things I | | |
| **COURSE CODE:** | 420-420-LE | | |
| **WEIGHT:** | 20% of the final score | | |
| **SEMESTER:** | Winter 2024 | | |
| **TEACHER:** | Gabriel Astudillo  [gaastudillo@crcmail.net](mailto:gaastudillo@crcmail.net) | Office C-241 |  |

## Project Introduction

To finish the course Internet of Things 1 you will develop an application that will use different sensors(inputs) and actuators(outputs) connected to your Raspberry Pi.

This is a group assignment (max 2 students per group). You can also work individually

## Project Description

In this project, students will design and implement a smart waste management system using Python and IoT hardware components. The system will utilize sensors to monitor waste levels in trash containers or dumpsters, visually represent the data via a GUI, and include a security feature to manage access to the container.

## Requirements:

* **Sensing Waste Levels:** Use an ultrasonic sensor to determine how full the trash container is. Be sure that the measurements are accurate. (This depends on the depth of the container)
* **Data Visualization:** Create a GUI using Tkinter to display the waste level in real-time and allow users to set a threshold for when the container should be considered full (Ex:80%).
* **Visual Alert System:** Implement an LED bar outside the container to indicate the waste level visually. This LED bar should change according to the waste level detected by the sensors. Ex:
  + **0%-**20% 1 Bar
  + **21% to 40% 2 Bars**
  + **41% to 60% 3 Bars and so on…**
* **Security System:** Develop a security system for the container using a red and blue button. The system requires a specific button sequence to be entered to unlock it (BLUE-RED-BLUE) in a 4 seconds interval. Use LEDs to indicate the status: green for correct sequence and red for incorrect.
* **Alarm System:** Incorporate a passive buzzer that is activated if the container is opened when locked, and it should stop once the container is securely closed. This event should also trigger a visual alert on the GUI. (Hint: you can use the same distance sensor to detect that the container is open)

## Project Deliverables:

### Deliverable 0 - Deadline: April 26th (1 point)

***Repository:*** Each student or group needs to create a repository for the project and add the teacher as collaborator since day one – My GitHub user is: gabrielastudillo.

### Deliverable 1 – Deadline: April 27th (5 points)

***System Design Document:*** Outline the design and architecture of the system, including:

* A paragraph with the most important points of your idea, the more details you put in, the easier it will be to prepare the demo (1)
* A list of hardware to use with images of each element (1)
* A block diagram of how to interconnect all the elements. Identify the GPIO pins to use (2)
* A mockup of the graphical interface to be developed (1)

All these elements need to be done in the readme of your repository; you can find inspiration here: <https://github.com/Aditya-Alexander/Pet-Feeder>

### Deliverable 2 – Deadline: May 3rd (5 points)

First draft: In this first draft, you will validate that all your sensors and actuators work **at least** individually (you can use the examples on the repo) and also a first basic GUI is presented:

* Connect and test your hardware components one at the time to be sure that everything works correctly. You can use a basic Python code to test. (2)
  + Hint: to test the distance sensor you can use a black piece of paper (matte)
* Program the GUI, at this point you can use some dummy functions to test. (2)
* Update your repo with ALL the code produced on this first iteration (be sure to comment in-line). (1)

### Deliverable 3 (Final presentation) – Deadline May 10th (9 points)

* Code Implementation: Complete Python code for the sensor data acquisition, GUI, and control logic for the security and alert systems. (4)
* Final Presentation: A presentation explaining the design, challenges faced, and demonstrations of the system functionality. (Be sure that you have fully tested your application in advance) (3)
* User Manual: A guide describing how to operate the smart waste management system, including how to set thresholds and enter the security sequence (in your repo). (2)