University of Hertfordshire UH

Bachelor of Engineering in Robotics and Artificial Intelligence

Team Project Module

Project Title:

AlphaMini – Robot Classroom Assistant Teacher (Rcat)

Group Progress Report 2

Name
Group Leader: Monish Naresh Kumar
Member #1: Cornelius Fulvian
Member #2: La Min Aung
Member #3: Joseph Santhosh Aaron

Project Supervisor: Sarina Tajudin

Date: 12/10/2025

Words Count: 1220

(excl. Table of content/references)

Table of Contents

Abstract

For this module we have been challenged to explore and use the AlphaMini, an intelligent humanoid robot developed by UBTECH Robotics. The AlphaMini is baked with rich artificial intelligence hardware, combined with uCode graphical programming tools and curriculum, to let students design, build, program scenarios with AI technologies, make the robot respond to human's directions and environmental changes, interact with self-built machineries, and actualize other brilliant ideas. (NIE, NTU , n.d.)

List of Figures

List of Tables

1. Introduction

1.1 Updated Aim and Objectives

Aim

To reinstate, the fundamental aim of this project is to implement the AlphaMini as a **Robot Classroom Assistant Teacher (Rcat)** designed to support the main teacher in managing and automating mundane tasks and improve efficiency of classroom discussions, lesson delivery, etc;. The general aim is for our finalized Rcat to demonstrate its capabilities to improve and enrich the efficiency of the learning environment for both students and teachers/school admin staff.

Objectives

The objectives of this project are as follows:

1) Compact attendance taking system

- The AlphaMini must be capable of identifying a QR code is presented to it
- The AlphaMini must capture an image of the QR code and send it to Remote PC
- Remote PC should decipher the QR code
- The Data from the QR code like date time and name should be logged in an external SQL / Excel Sheet.

2) Language Teaching guide

- The Alphamini should be capable of Speech recognition
- Link the Alphamini to a translation software or AI tool for language help
- Reply to the user with correct help in language

3) Interactive quiz handler

- Not finalized on execution but should be able to conduct a quiz in real time.

4) Navigation around classroom

- Basic Wandering
- Obstacle Avoidance
- Point to point movement

5) Sensing and object recognition

- Train an object recognition logic model
- Install Object Identification in Alphamini

6) Timekeeper

- Design and Build a circuit with ESP32 with a LCD / 7 segment display for a timeclock
- Program the AlphaMini to be able to start stop and set the timeclock
- Enable voice activation eg; “Start a timer for a 30 min test”

7) Peripherals

- 3D Print a Idle Hub / Home station for the Rcat to charge and sync to the wifi
- 3D Print a Teachers Stick for aesthetic and potential realism movements
- 3D Print a mock small scale classroom environment for simulation

1.2 Current State of the project

As of date of report submission, we have managed to successfully connect our AlphaMini robot to the App, Wi-Fi and the Remote PC. We were unable to program the robot using Vscode due to python issues but switching to Pycharm solved all issues. We were able to run movement test codes using the Remote PC's Keyboard and it was responsive however we came to understand that the button mapping was not intuitive and will need to figure that out for the navigation behaviour.

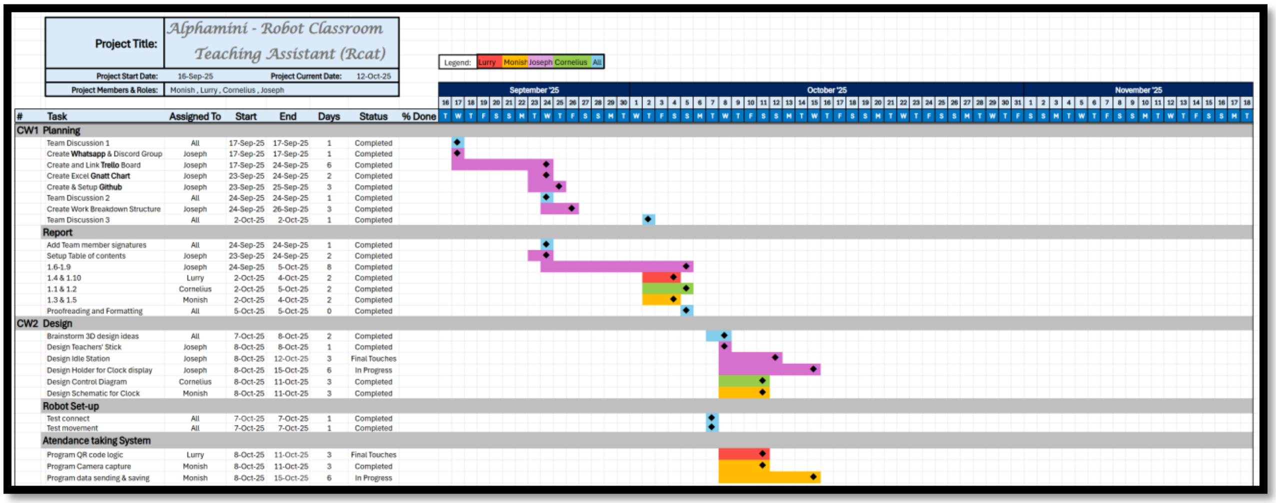
Following the milestones table that we had previously decided on, we focused on the attendance taking system and timekeeper deliverables.

As of now, we have managed to set-up the camera and generated some QR codes to test its capability to capture the image. Also managed to successfully program the robot to be able to send the captured image to the Remote PC, we have yet to integrate the deciphering logic but set up a non-documentable basic test bench logging system.

For the timekeeping we have managed to trial some basic test code for the AlphaMini internal clock, but have yet to do anything with it. We instead prioritized building the physical clock.

At the same time members not working on the code worked on creating CAD designs.

1.3 Updated Schedule



Project Title:	Alphamini - Robot Classroom Teaching Assistant (Rcat)
Project Start Date:	16-Sep-25
Project Current Date:	12-Oct-25
Project Members & Roles:	Monish , Lurry , Cornelius , Joseph

Legend: 🟥 Lurry 🟧 Monish 🟪 Joseph 🟩 Cornelius 🟦 All

#	Task	Assigned To	Start	End	Days	Status	% Done
CW1	**Planning**						
	Team Discussion 1	All	17-Sep-25	17-Sep-25	1	Completed	
	Create **Whatsapp** & Discord Group	Joseph	17-Sep-25	17-Sep-25	1	Completed	
	Create and Link **Trello** Board	Joseph	17-Sep-25	24-Sep-25	6	Completed	
	Create Excel **Gnatt Chart**	Joseph	23-Sep-25	24-Sep-25	2	Completed	
	Create & Setup **Github**	Joseph	23-Sep-25	25-Sep-25	3	Completed	
	Team Discussion 2	All	24-Sep-25	24-Sep-25	1	Completed	
	Create Work Breakdown Structure	Joseph	24-Sep-25	26-Sep-25	3	Completed	
	Team Discussion 3	All	2-Oct-25	2-Oct-25	1	Completed	
	Report						
	Add Team member signatures	All	24-Sep-25	24-Sep-25	1	Completed	
	Setup Table of contents	Joseph	23-Sep-25	24-Sep-25	2	Completed	
	1.6-1.9	Joseph	24-Sep-25	5-Oct-25	8	Completed	
	1.4 & 1.10	Lurry	2-Oct-25	4-Oct-25	2	Completed	
	1.1 & 1.2	Cornelius	2-Oct-25	5-Oct-25	2	Completed	
	1.3 & 1.5	Monish	2-Oct-25	4-Oct-25	2	Completed	
	Proofreading and Formatting	All	5-Oct-25	5-Oct-25	0	Completed	
CW2	**Design**						
	Brainstorm 3D design ideas	All	7-Oct-25	8-Oct-25	2	Completed	
	Design Teachers' Stick	Joseph	8-Oct-25	8-Oct-25	1	Completed	
	Design Idle Station	Joseph	8-Oct-25	12-Oct-25	3	Final Touches	
	Design Holder for Clock display	Joseph	8-Oct-25	15-Oct-25	6	In Progress	
	Design Control Diagram	Cornelius	8-Oct-25	11-Oct-25	3	Completed	
	Design Schematic for Clock	Monish	8-Oct-25	11-Oct-25	3	Completed	
	Robot Set-up						
	Test connect	All	7-Oct-25	7-Oct-25	1	Completed	
	Test movement	All	7-Oct-25	7-Oct-25	1	Completed	
	Atendance taking System						
	Program QR code logic	Lurry	8-Oct-25	11-Oct-25	3	Final Touches	
	Program Camera capture	Monish	8-Oct-25	11-Oct-25	3	Completed	
	Program data sending & saving	Monish	8-Oct-25	15-Oct-25	6	In Progress	

Figure 1: Updated Gnatt Chart (12/10)

2. Project Updates

2.1 Individual Contributions

We decided to assign some key roles so that tasks can be handled straightforward and to members strengths so that work can be handled efficiently.

Assigned Primary and Secondary roles for overall project

	Monish	Cornelius	Lurry	Joseph
Primary	Coding	Research	Coding / Building	Groundwork
Secondary	Building	Coding / Building	3D Printing	3D Design and Printing

Table 1: Role Assignments

<u>Student #1: Monish Naresh Kumar</u>

I am responsible for the ideation of features (partially), creation and implementation of a physical clock for the "timekeeper" function, and coding for the attendance feature.

My contributions till the submission of this report are:

1. Research how to go about implementing the behaviours.
2. Ensure the connectivity of Alpha mini with remote desktop and start testing the camera functions for QR attendance.
3. Design and Create a circuit (virtual) for the digital clock using ESP32 and LCD display.

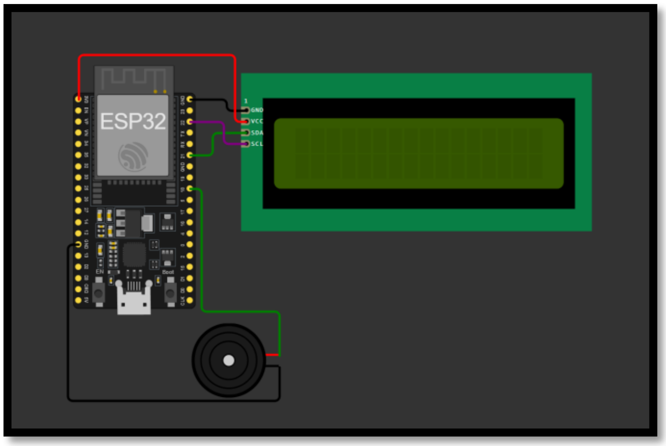
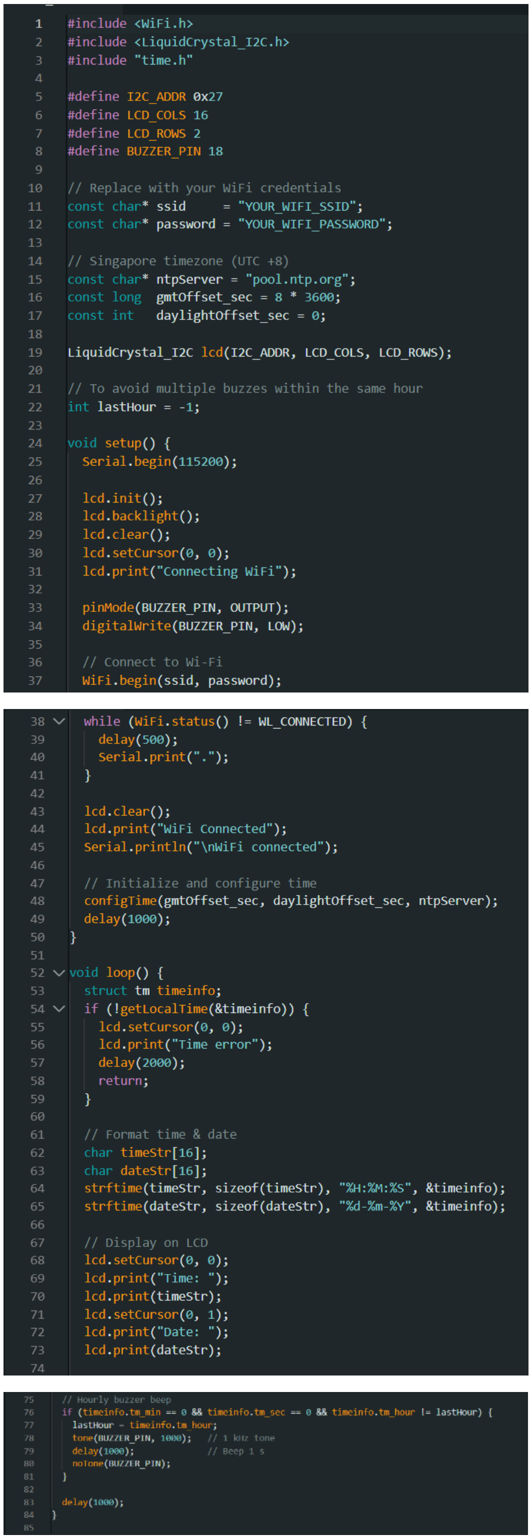
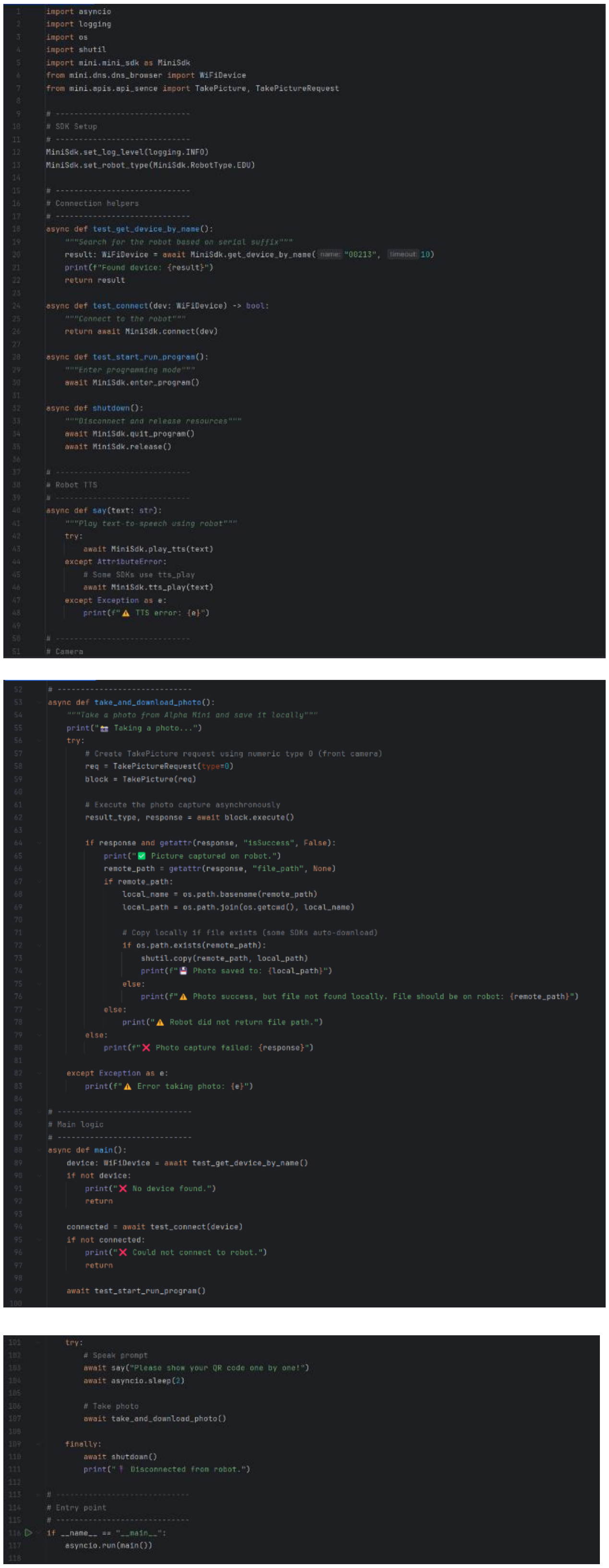


Figure 2: Clock Circuit Diagram



```cpp
#include <WiFi.h>
#include <LiquidCrystal_I2C.h>
#include "time.h"

#define I2C_ADDR 0x27
#define LCD_COLS 16
#define LCD_ROWS 2
#define BUZZER_PIN 18

// Replace with your WiFi credentials
const char* ssid     = "YOUR_WIFI_SSID";
const char* password = "YOUR_WIFI_PASSWORD";

// Singapore timezone (UTC +8)
const char* ntpServer = "pool.ntp.org";
const long  gmtOffset_sec = 8 * 3600;
const int   daylightOffset_sec = 0;

LiquidCrystal_I2C lcd(I2C_ADDR, LCD_COLS, LCD_ROWS);

// To avoid multiple buzzes within the same hour
int lastHour = -1;

void setup() {
  Serial.begin(115200);

  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Connecting WiFi");

  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  lcd.clear();
  lcd.print("WiFi Connected");
  Serial.println("\nWiFi connected");

  // Initialize and configure time
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  delay(1000);
}

void loop() {
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) {
    lcd.setCursor(0, 0);
    lcd.print("Time error");
    delay(2000);
    return;
  }

  // Format time & date
  char timeStr[16];
  char dateStr[16];
  strftime(timeStr, sizeof(timeStr), "%H:%M:%S", &timeinfo);
  strftime(dateStr, sizeof(dateStr), "%d-%m-%Y", &timeinfo);

  // Display on LCD
  lcd.setCursor(0, 0);
  lcd.print("Time: ");
  lcd.print(timeStr);
  lcd.setCursor(0, 1);
  lcd.print("Date: ");
  lcd.print(dateStr);

  // Hourly buzzer beep
  if (timeinfo.tm_min == 0 && timeinfo.tm_sec == 0 && timeinfo.tm_hour != lastHour) {
    lastHour = timeinfo.tm_hour;
    tone(BUZZER_PIN, 1000);   // 1 kHz tone
    delay(1000);              // Beep 1 s
    noTone(BUZZER_PIN);
  }

  delay(1000);
}
```

Figure 3: Clock Circuit Code



```python
import asyncio
import logging
import os
import shutil
import mini.mini_sdk as MiniSdk
from mini.dns.dns_browser import WiFiDevice
from mini.apis.api_sence import TakePicture, TakePictureRequest

# ----------------------------
# SDK Setup
# ----------------------------
MiniSdk.set_log_level(logging.INFO)
MiniSdk.set_robot_type(MiniSdk.RobotType.EDU)

# ----------------------------
# Connection helpers
# ----------------------------
async def test_get_device_by_name():
    """Search for the robot based on serial suffix"""
    result: WiFiDevice = await MiniSdk.get_device_by_name( name: "00213",  timeout: 10)
    print(f"Found device: {result}")
    return result

async def test_connect(dev: WiFiDevice) -> bool:
    """Connect to the robot"""
    return await MiniSdk.connect(dev)

async def test_start_run_program():
    """Enter programming mode"""
    await MiniSdk.enter_program()

async def shutdown():
    """Disconnect and release resources"""
    await MiniSdk.quit_program()
    await MiniSdk.release()

# ----------------------------
# Robot TTS
# ----------------------------
async def say(text: str):
    """Play text-to-speech using robot"""
    try:
        await MiniSdk.play_tts(text)
    except AttributeError:
        # Some SDKs use tts_play
        await MiniSdk.tts_play(text)
    except Exception as e:
        print(f"⚠️ TTS error: {e}")

# ----------------------------
# Camera
# ----------------------------
async def take_and_download_photo():
    """Take a photo from Alpha Mini and save it locally"""
    print("📷 Taking a photo...")
    try:
        # Create TakePicture request using numeric type 0 (front camera)
        req = TakePictureRequest(type=0)
        block = TakePicture(req)

        # Execute the photo capture asynchronously
        result_type, response = await block.execute()

        if response and getattr(response, "isSuccess", False):
            print("✅ Picture captured on robot.")
            remote_path = getattr(response, "file_path", None)
            if remote_path:
                local_name = os.path.basename(remote_path)
                local_path = os.path.join(os.getcwd(), local_name)

                # Copy locally if file exists (some SDKs auto-download)
                if os.path.exists(remote_path):
                    shutil.copy(remote_path, local_path)
                    print(f"💾 Photo saved to: {local_path}")
                else:
                    print(f"⚠️ Photo success, but file not found locally. File should be on robot: {remote_path}")
            else:
                print("⚠️ Robot did not return file path.")
        else:
            print(f"❌ Photo capture failed: {response}")

    except Exception as e:
        print(f"⚠️ Error taking photo: {e}")

# ----------------------------
# Main logic
# ----------------------------
async def main():
    device: WiFiDevice = await test_get_device_by_name()
    if not device:
        print("❌ No device found.")
        return

    connected = await test_connect(device)
    if not connected:
        print("❌ Could not connect to robot.")
        return

    await test_start_run_program()

    try:
        # Speak prompt
        await say("Please show your QR code one by one!")
        await asyncio.sleep(2)

        # Take photo
        await take_and_download_photo()

    finally:
        await shutdown()
        print("🔌 Disconnected from robot.")

# ----------------------------
# Entry point
# ----------------------------
if __name__ == "__main__":
    asyncio.run(main())
```

Figure 4: QR capture and PC code

<u>Student #2: Cornelius Fulvian</u>

I am responsible for research and building side of the project.

My contributions till the submission of this report are:

1. Research and Evaluate the existing problem and substantiate the need for a solution.
2. Come up with a Business proposition.
3. Create and Design a Control Diagram for the Project.

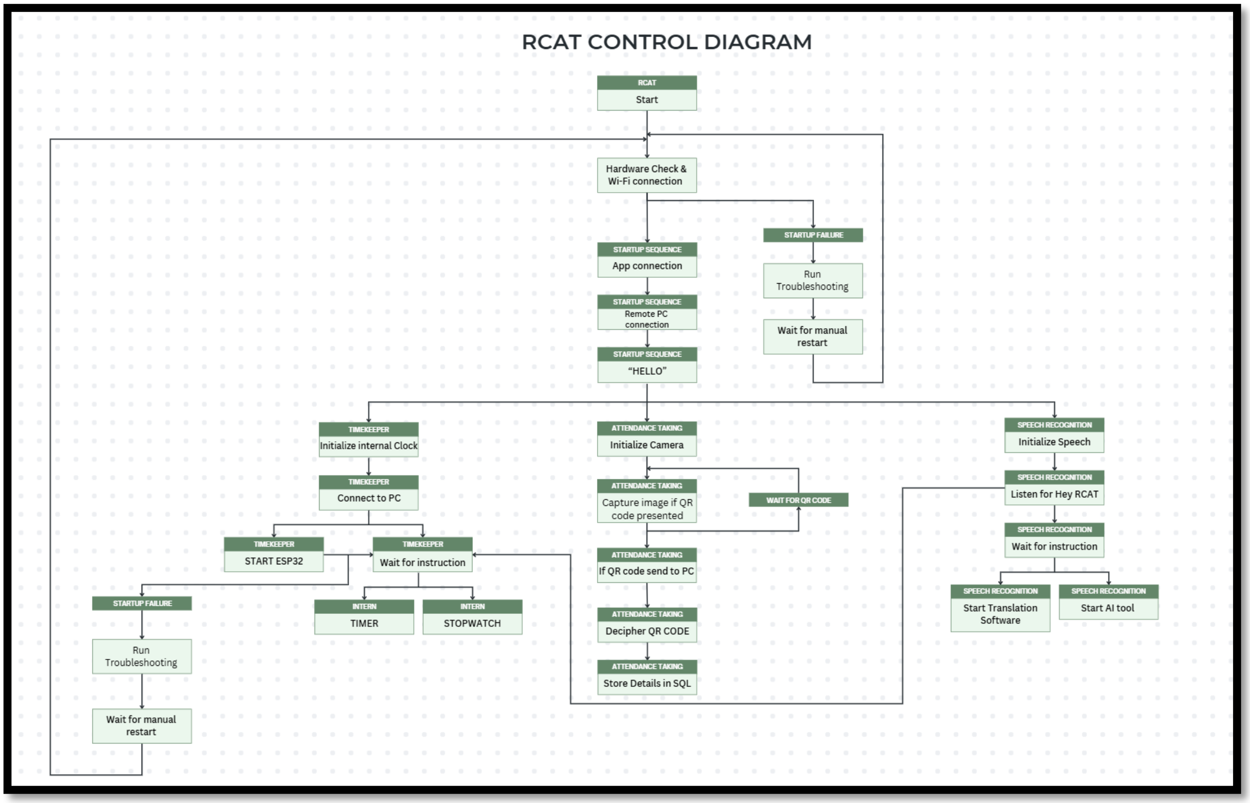


Figure 55: Control Diagram

<u>Student #3: La Ming Aung</u>

I am responsible for the ideation of features (partially), handling Remote PC side of coding for the attendance feature.

My contributions till the submission of this report are:

1. Research how to go about implementing the behaviours.
2. Identify the Limitations and hone the Scope of the project.
3. Setup a SQL for attendance data logging. (not documented - random test data)

Student #4: Joseph Santhosh Aaron

I am responsible for the ideation of features (partially), designing aspects of the project and providing support for coding if necessary.

My contributions till the submission of this report are:

1. Create and setup all the necessary groundwork tools (Trello, GitHub, Excel Gnatt Chart, Work Breakdown Structure, Communication channels)
2. Design some 3D designs for peripherals for the robot and its simulation environment. Test prints for getting measurements of arm size correct.
3. Update the Trello and Gnatt Chart Regularly.

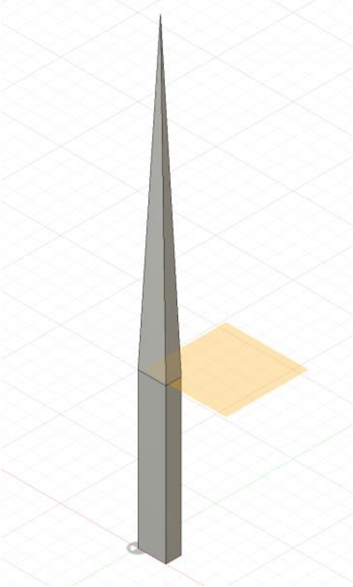
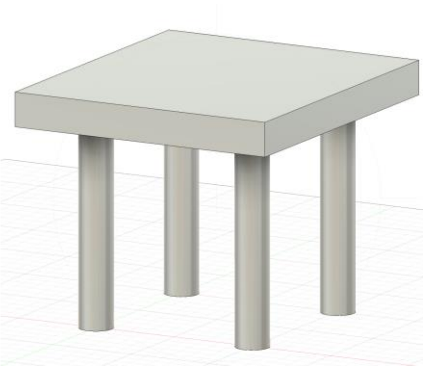
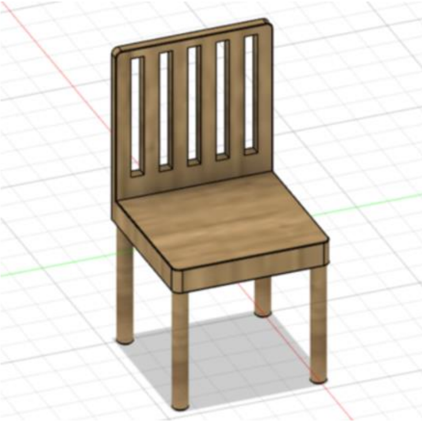
		
Teachers' Stick v0.2	*Class Table*	*Chair (Hasan, 2022)*

Figure 66: 3D Designs

2.2 Remaining Work in terms of Design

Physical work

After designing the physical Circuitry for the timeclock we need time to source the necessary parts (primarily the ESP32 microcontroller board) and build and test the full circuit.

CAD work

Since the working circuit for the timeclock has yet to be fully built we have made a unscaled prototype design for the LCD display holder and the rest of the components (ESP32, battery, wiring holes). A general idea for the Idle Hub has also been conceived and is being worked on however, it is merely in budding stage and will remain a template as changes are almost guaranteed.

Control Diagram

Since most of the other behaviours logic flows haven't been created, the control Diagram still remains unfinished and will be worked on concurrently with progress.

3. Conclusion

If we were to follow the milestone table to the letter T, then we are off track in terms of completion dates. However, since ideation for behaviour implementation happens concurrently, it is logical for tasks to be delayed (procurement of parts), furthermore, some tasks are overlapping with others and are ahead of time.

4. All team member signatures

Name	Signature
Monish Naresh Kumar	
La Min Aung	
Cornelius Fulvian	
Joseph Santhosh Aaron	

Table 2: Members' Signatures

References

Hasan, A. (2022). chair. https://grabcad.com/library/chair-898.

NIE, NTU . (n.d.). From libguides : https://libguides.nie.edu.sg/c.php?g=965796&p=7017302