LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS5053NI– Cloud Computing & IoT

“AI Enhanced Smart Parking Management System”

Assessment Type

50% System Development Report

Semester
2023 Spring
Group members

London Met ID	Student Name
22067814	Miraj Deep Bhandari
22068751	Jenish Katuwal
22067756	Krish Shrestha
22068770	Kritika Siwakoti
22067807	Rijan Paudel

Assignment Due Date: Friday, January 12, 2024

Assignment Submission Date: Thursday, January 11, 2024

Word Count: 3000

Acknowledgement

We express our sincere appreciation to our module teachers, Mr. Sugat Man Shakya, Mr. Ayush Bhakta Pradhanang, and Shishir Subedi, for the invaluable opportunity to collaborate on the "AI Enhanced Smart Parking Management System" IoT project. This effort not only deepened our research experience but also enhanced our understanding of IoT, thanks to their insightful guidance.

Our teachers shared invaluable ideas on cloud computing integration with IoT tools, resulting in the successful fusion of AI, IoT, and cloud technologies. Special thanks to our dedicated team members whose collaborative spirit and support were instrumental in bringing the project to completion, demonstrating effective teamwork.

Abstract

This AI Smart Parking project, utilizing Arduino Uno, Ultrasonic Sensors, and deep learning models, enhances parking efficiency in busy areas. The system automates parking gates, assesses space availability, and allows users to check parking status on smartphones via cloud integration.

In areas like malls, offices, colleges, and schools, our AI Enhanced Smart Parking system addresses the challenge of inefficient parking management. The automated parking gates, coupled with a deep learning model, detect available parking spaces, providing users with real-time information on their smartphones through seamless cloud integration. This innovative solution significantly reduces the need for additional staff, streamlining the monitoring of parking spaces for organizations.

Table of Contents

Table of Figures

Table of Tables

Introduction

Internet of Things (IOT) projects aim to simplify tasks and improve convenience by integrating technologies such as sensors, software, and microcontrollers. This combination of technologies collectively forms the IoT framework. In today's landscape, IOT seamlessly integrates with Cloud Computing Technologies, which, in turn, can be harmonized with Artificial Intelligence. This integration allows for the embedding of AI technologies alongside IoT devices, facilitating the exchange of information through cloud technologies.

Our project, the AI Enhanced Smart Parking Management System, embodies a practical IoT solution. Leveraging sensors, such as ultrasonic devices, for object detection at parking gates and slots, we can efficiently control servo motors and LED lights. What sets our system apart is the integration of a deep learning model, enhancing accuracy in determining parking slot availability. Users receive real-time updates on their smartphones, simplifying the process of finding vacant parking spots and elevating their overall parking experience.

1.1. Current scenario

The parking crisis in Nepal, particularly in the Kathmandu Valley, is intensified by the absence of smart parking systems. The surge in registered vehicles, from 1.2 million in 2010/11 to 2.9 million in 2017/18, has outpaced infrastructure development. Kathmandu Metropolitan City (KMC) provides only 16,000 parking spaces, significantly inadequate for the growing demand fueled by the rapid increase in vehicles. The problem is compounded by frequent parking violations, with over 180,000 instances recorded in 2022 by the Metropolitan Traffic Police Division (MTPD). The lack of smart parking systems further complicates the issue, as manual enforcement struggles to address widespread disregard for regulations, deepened traffic congestion and safety concerns. The assessment emphasizes a substantial deficit of about 1.58 million parking spaces in the Kathmandu Valley, minimizing the urgent need for adopting smart parking systems to efficiently manage and optimize available spaces, mitigating challenges posed by the escalating number of vehicles (annapurnaexpress, 2022).

1.2. Problem Statement and Project as a Solution

1.2.1 Problem Statement

Parking complexities prevails in Nepali urban hubs like cities, malls, offices, colleges, and schools due to the absence of an effective parking system. This leads to vehicle congestion, traffic jams and pedestrian challenges. Traditional parking methods, including paper tickets and manual checks, provoke the issue, slowing down processes and complicating space management. Weekdays have a high parking demand at 143.801 vehicle hours per 1000 square feet, with an efficiency of 96.35%, urging the need for more spaces. Weekends, with 59.464 vehicle hours per 1000 square feet and 39.84% efficiency, show no immediate need for additional parking areas (Sakar Pahari, 2020).

Table: Summary

Street segment	**Street Length** (feet)	**Area** (square feet)	**Maximum Reading**	**Minimum Reading**	**Parking Load** (vehicle hours per 1000 square feet)		**Parking Efficiency (η)**	
					Weekdays	**Weekend**	**Weekdays**	**Weekend**
Pako Sadak	150.6	1136.35	56	17	102.815	45.17	104.32%	45.83%
Khechapukhu Sadak I	142	1015.04	51	8	91.35	47.62	90.90%	47.39%
Khechapukhu Sadak II	93.8	768.44	65	7	148.93	32.53	88.03%	19.23%
Sundhara Marg	146.3	1008.15	66	26	136.444	77.039	104.21%	58.84%
Pyukha Marg	127.6	1263.25	56	8	81.228	18.471	91.62%	20.83%
Naya Sadak	109	857.67	64	19	143.801	59.464	96.35%	39.84%

Figure 1: Summary table of Parking load in kathmandu.

1.2.2 Project as Solution

Our solution introduces a smart parking system seamlessly integrated with AI to tackle these challenges. Utilizing sensors for vehicle detection and automatic gate control, users can access the system through a user-friendly smartphone app. This eliminates the need for additional staff, as a sophisticated deep learning model with object detection algorithms efficiently manages the allocation of free slots. The system not only aids in crowd management within parking areas but also empowers administrators to monitor and guide vehicles effortlessly through a dedicated app.

1.3 Aim

The main purpose of this project is to modernize traditional parking systems using advanced technologies to establish an automated parking system for the effective management of vehicles.

1.4 Objectives

- The system is expected to open the gate automatically once the vehicle is detected by the sensor.
- The system is build using a deep learning model which performs object detection task, for identifying vacant parking spaces.
- The system assists the user through a mobile application to check real-time parking availability.
- The system helps the administrator to look over the vehicle and guide them remotely for parking the vehicle in slots.

2. Background

2.1 System Overview

This study developed an efficient method for reducing parking problems in urban areas. In our project, we addressed this issue by creating a system that utilizes AI and IoT tools to resolve parking-related issues.

In this prototype, the project aims to develop a vehicle management system for parking. When a vehicle enters the parking area, the sensor detects it and opens the gate automatically. After the car passes, it closes automatically. Subsequently, when the car reaches the parking slot, the light glows to indicate that the space is occupied. In our system, drivers can view available parking spaces through a mobile application. This is made possible by the object detection deep learning model and cloud computing integration.

In the gate, a servo motor, along with ultrasonic sensors, is employed. In the parking slots, LEDs are used in conjunction with ultrasonic sensors. Deep learning is utilized to detect vehicles and count the free spaces, providing real-time information to both users and administrators through the mobile application.

2.2 Design Diagram

2.2.1 Block Diagram

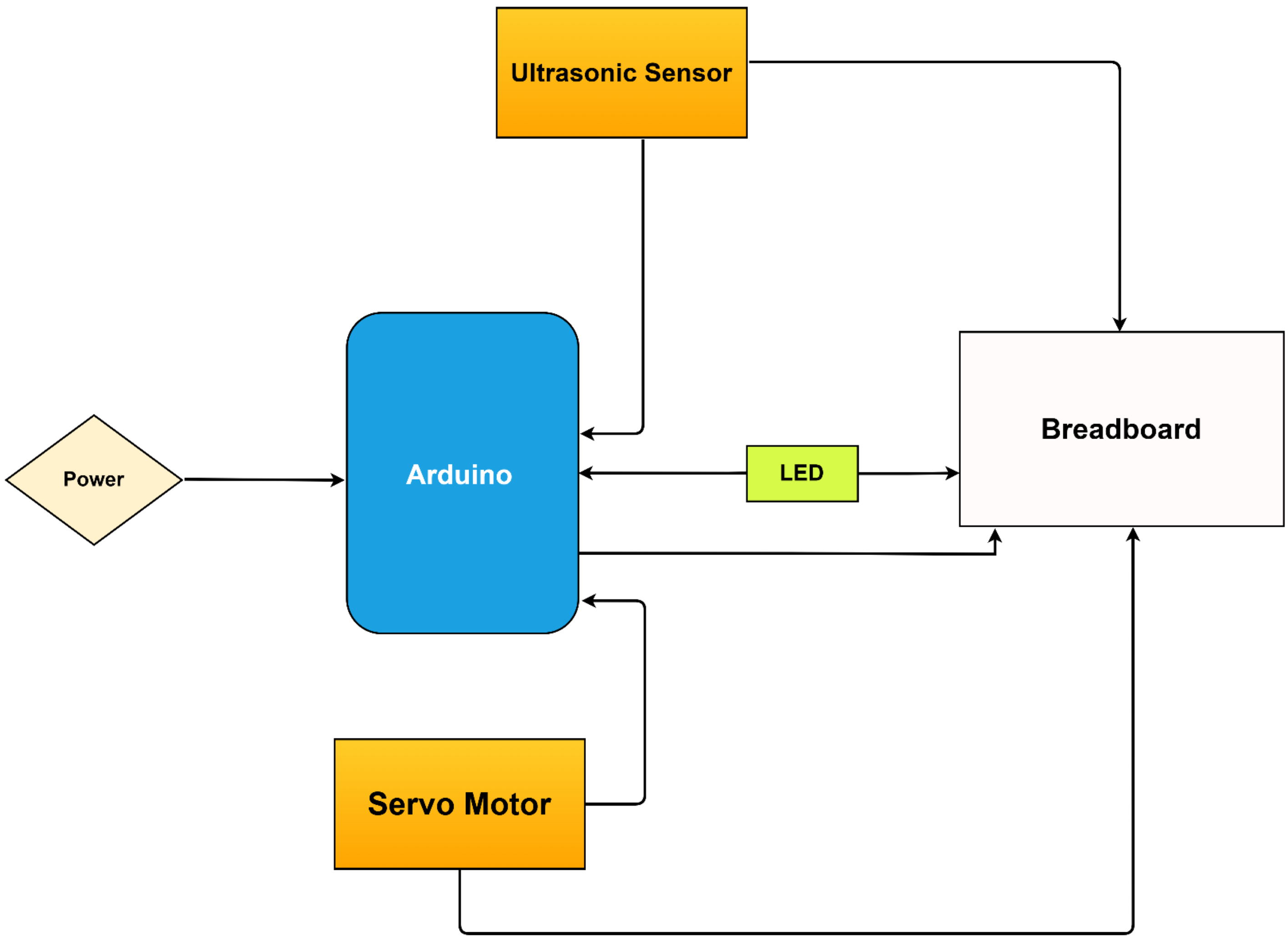

Figure 1: Block Diagram

2.2.2 System Architecture

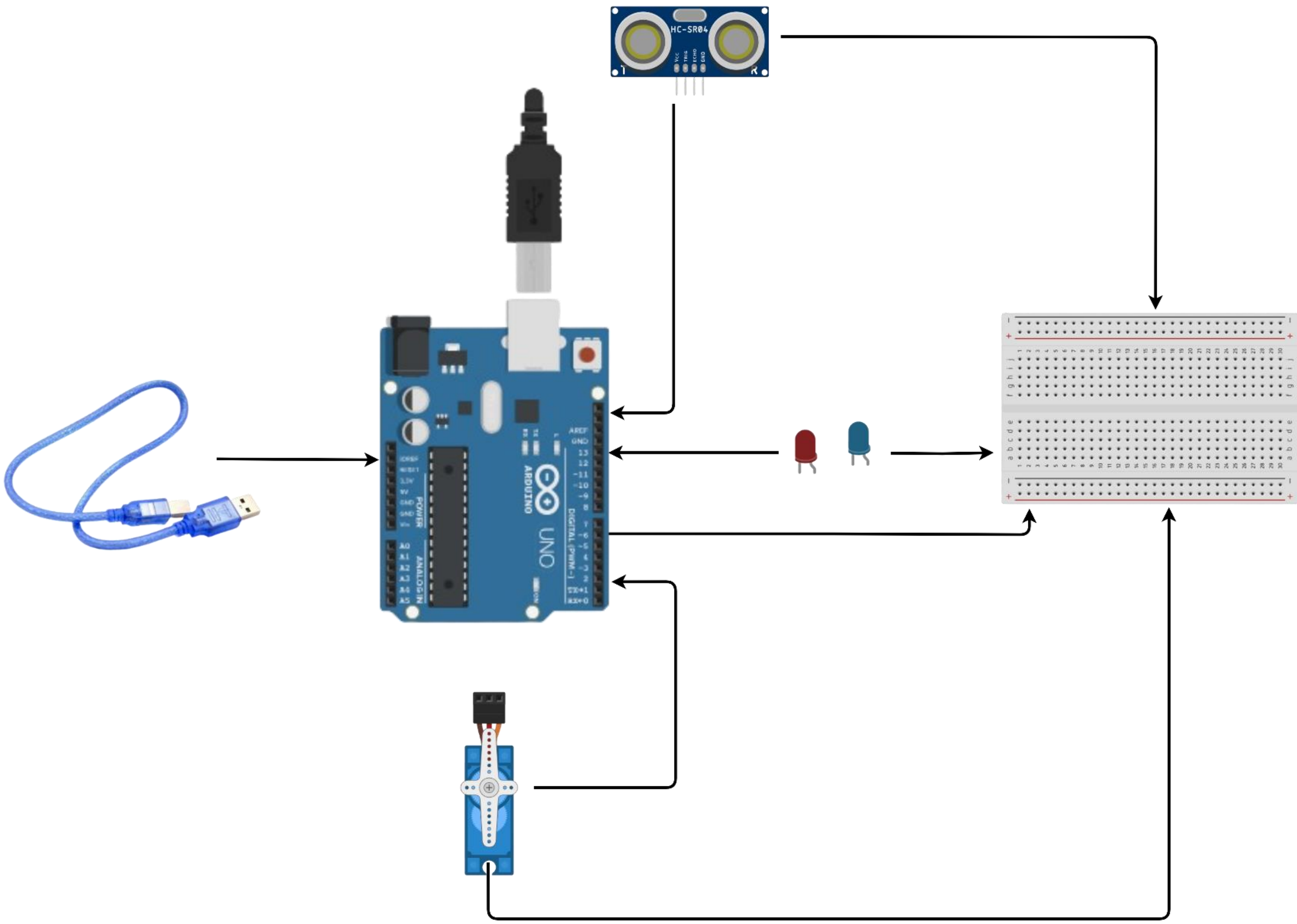

Figure 2: System Architecture

2.2.3 Circuit Diagram

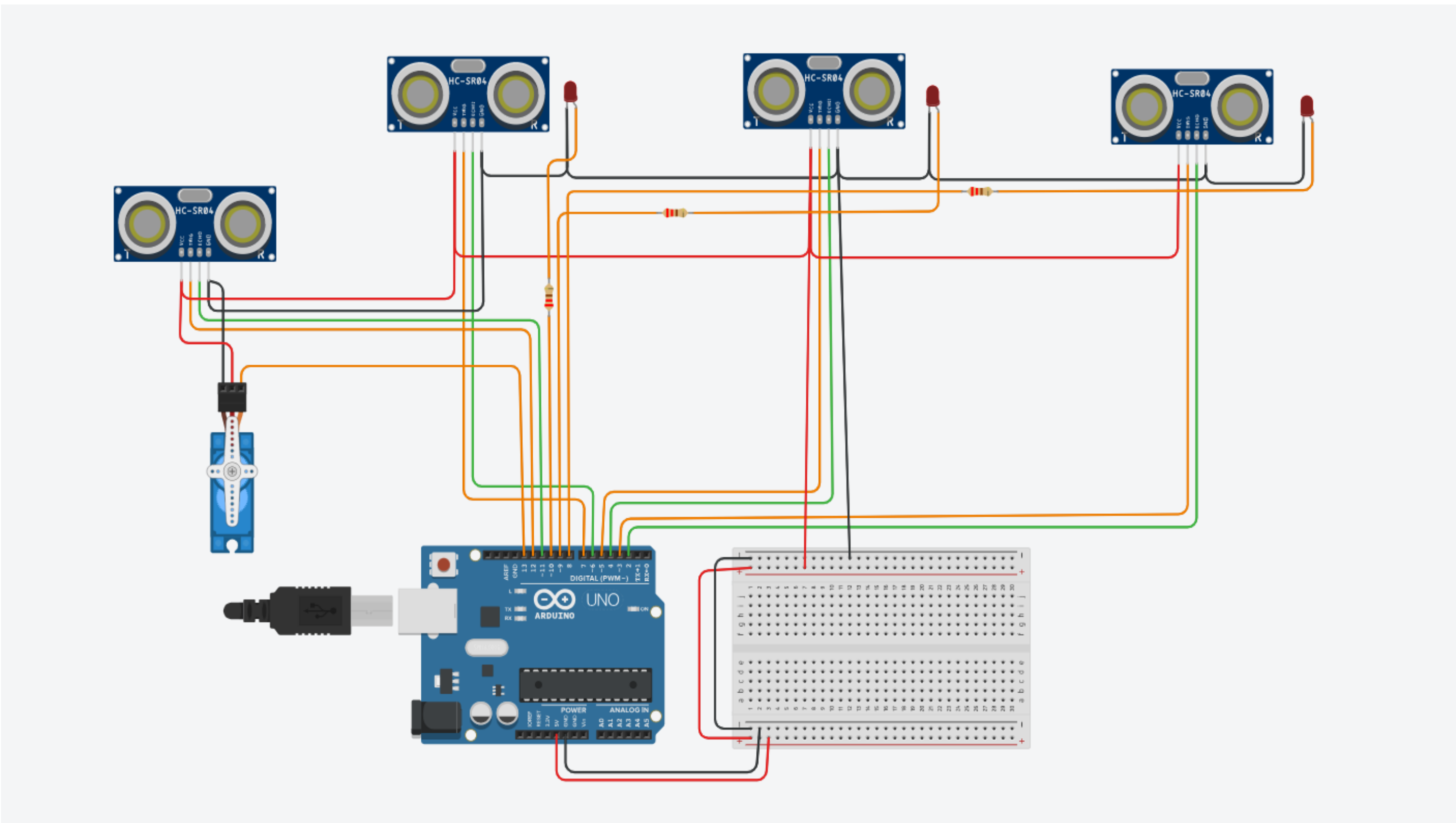

Figure 3: Circuit diagram

2.2.4 Schematics Diagram

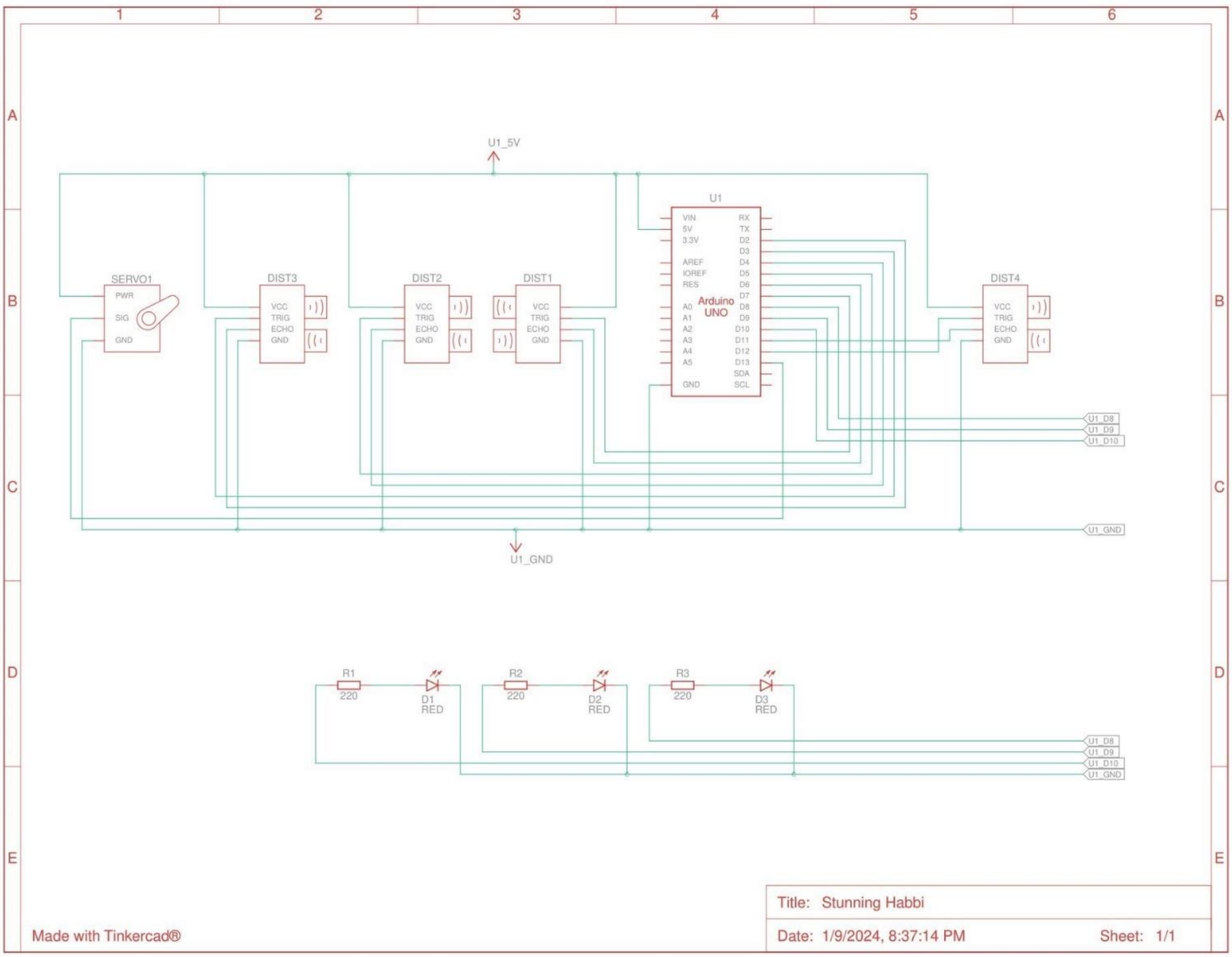

Figure 4: Schematics diagram

2.2.5 Flow Chart

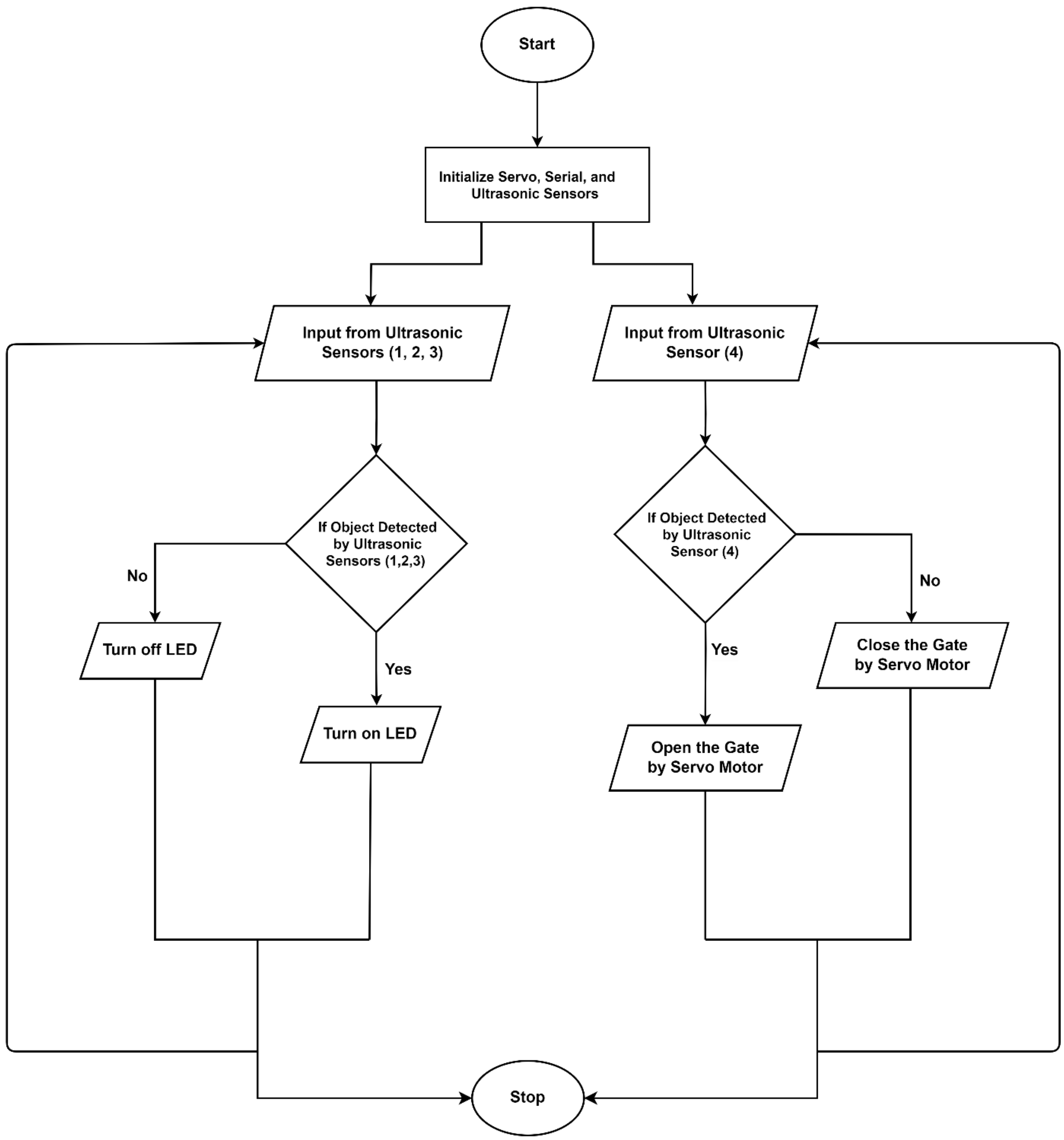

Figure 5: Flow Chart of the System

2.3 Requirement Analysis

2.3.1 Hardware Components

In creating the AI Enhanced Smart Parking Management System, we employed various hardware components which are as follows :

- ***Arduino Uno***
- ***ultrasonic sensors***
- ***servo motors***
- ***LED lights***
- ***jumper wires***
- ***resistors***

2.3.2 Software's

In creating the AI Enhanced Smart Parking Management System, we employed various software which are as follows :

- ***Arduino IDE***
- ***Tinkercad***
- ***Jupyter Notebook***
- ***Microsoft Word***

3. Development

The development section includes process of development of the IOT project.

Step 1: The design and the planning

Design and planning for the project were executed in TinkerCAD software, providing a virtual platform for project prototyping. Seven crucial hardware components were identified: Arduino Uno, Breadboard, Ultrasonic Sensors, Servo Motors, LED Lights, Jumper Wires, and Resistors. The virtual circuit design involved placing these components and establishing connections using jumper wires. To power the components, a connection was made between the ground and 5-volt pins of the breadboard and the corresponding pins on the Arduino. Servo Motors and Ultrasonic Sensors were powered accordingly. The servo's output pin was set to digital pin 13, while the sensor's input pin was set to analog pin 0. The LED lights were also connected to the Arduino and breadboard for output control. The TinkerCAD simulation offered valuable insights into the project's concept and operational mechanisms, facilitating a smooth transition to the real-life implementation and subsequent program development.

Step 2: Resource collection

To gather resources for our IoT project, we reached out to the IT resource department of the college through an application letter. They provided us with essential components like the Arduino Uno, Ultrasonic sensors, jumper wires, LED lights, Servo motors, Resistors and Breadboard. For the rest of the items, including the glues, extra jumper wires, we visited local stores and made purchases from the Arbit IT Solution. To complete our set of 7 required hardware components, we also bought some items from Arbit IT Solution. This mix of sources allowed us to collect all the necessary materials for our project efficiently.

Step 3: System Development.

Phase 1: The system development process is done same as process done to develop this project in tinker cad. Firstly, we made a connection between bread board and Arduino uno by establishing ground and a 5 voltage pins of an Arduino to the negative and positive terminals of the bread board respectively.

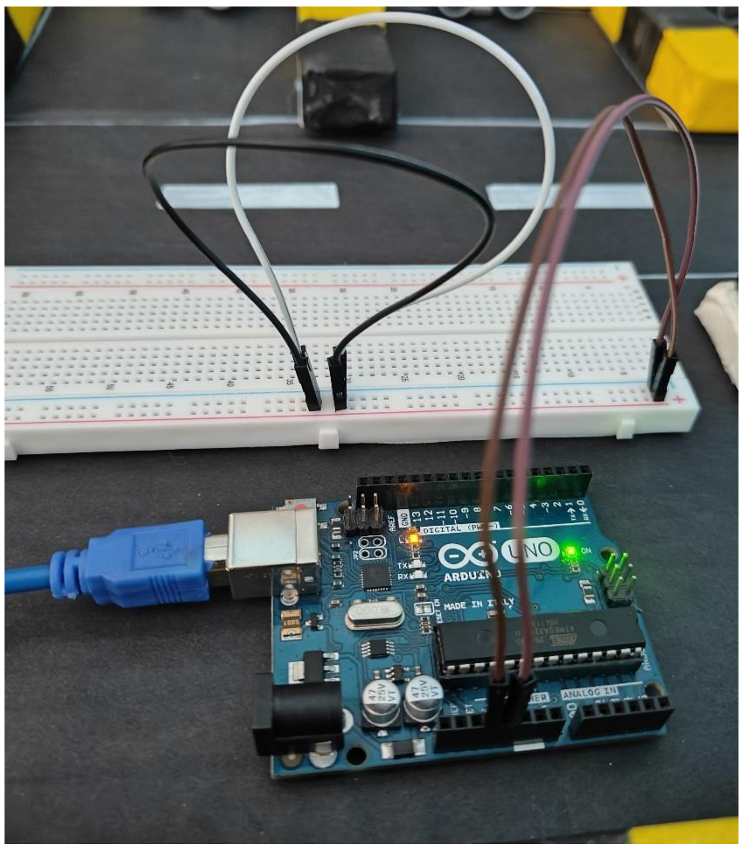

Figure 5: Picture of Phase 1 of System Development.

Phase 2: In phase 2 we connected the servo motor to the breadboard by linking its power and ground to the respective rails and its signal wire to a digital pin (13) on the Arduino . For the ultrasonic sensor, we attached VCC and GND to the power and ground rails and connect trig and echo to two digital pins (11,12) on the Arduino. This configuration enables the Arduino to control the servo motor based on ultrasonic sensor readings. In practical terms, the servo motor can be used to raise and lower a gate.

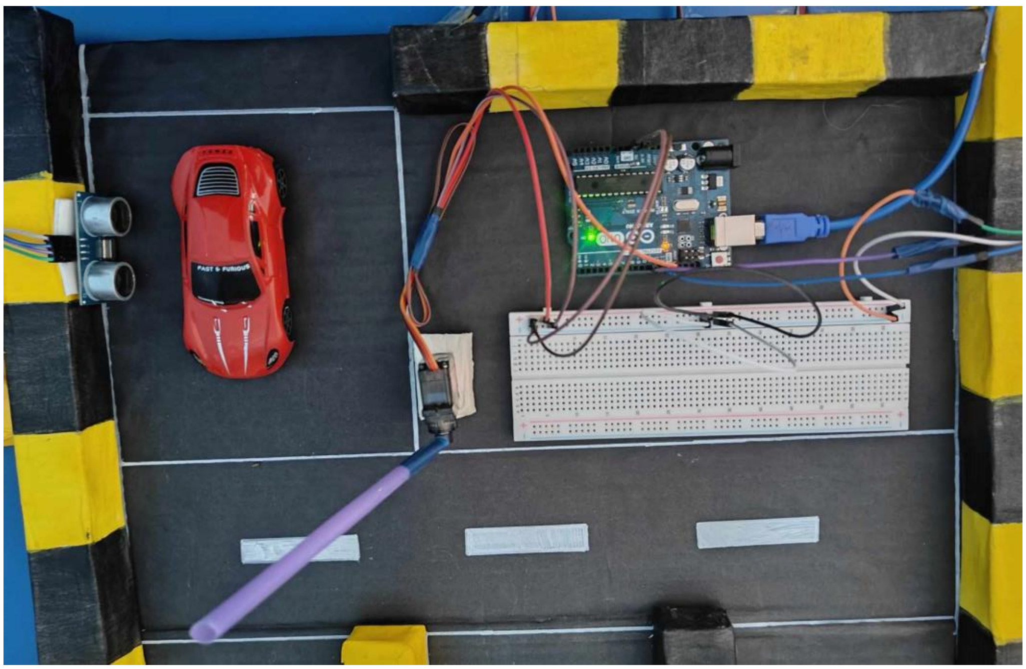

Figure 2: Picture of Phase 2 of System development.

Phase 3: In phase 3 of the system development, connect three ultrasonic sensors to the breadboard, linking their VCC and GND pins to the power and ground rails. Connect the trig and echo pins to distinct digital pins on the Arduino. This configuration enables the Arduino to communicate with the ultrasonic sensors and operate three LEDs based on their readings.

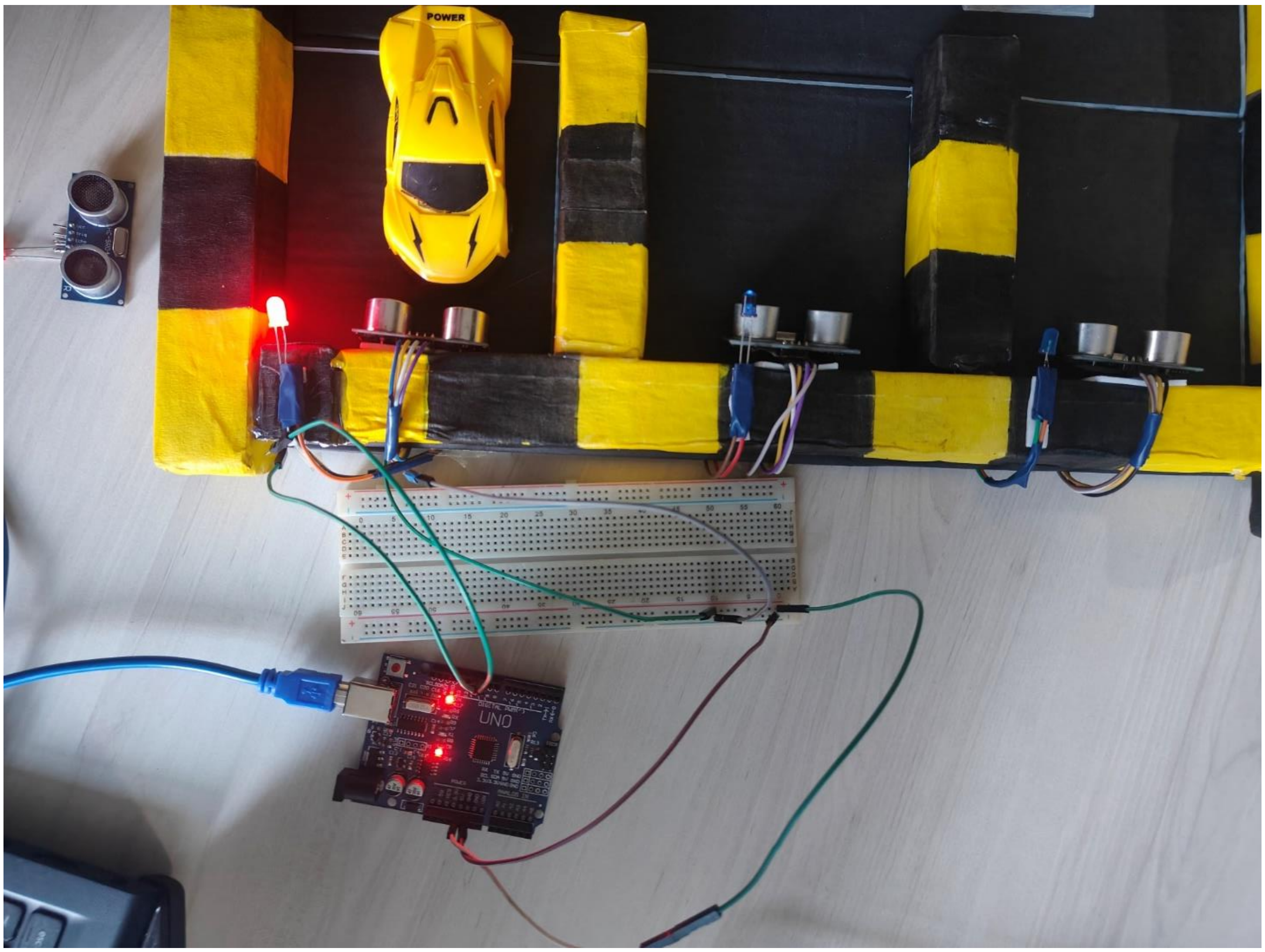

Figure 3: Picture of Phase 3 of System Development.

4. Results and Findings

4.1 Results

Following the comprehensive development process, the Smart Parking Management System stands as a robust solution for optimizing parking efficiency. By using ultrasonic sensors and Arduino Uno, the system precisely detects vehicles, allowing seamless gate operations and precise identification of available parking spaces. LED lights intuitively guide users to vacant slots. Firstly, the vehicle is detected through ultrasonic sensors which leds servo motor to raise the gate and after vehicle enters from gate the servo closes the gate. After the vehicle enters the available parking LED light gets turn on through ultrasonic sensors. The system not only streamlines parking but also demonstrates the potential for smart city solutions, contributing to efficient traffic and resource management.

4.2 Findings

Testing

Test 1: Successful Code Execution in Arduino IDE

Test	1
Objective:	To confirm the successful execution of the code in the Arduino IDE.
Activity:	Uploading the project code to the Arduino Uno.
Expected Result	The code should upload without errors, and the Arduino Uno should be ready for operation.
Actual Result	The code uploads successfully, and no errors are reported.
Conclusion	The test is successful.

Table 1: Table of Test 1

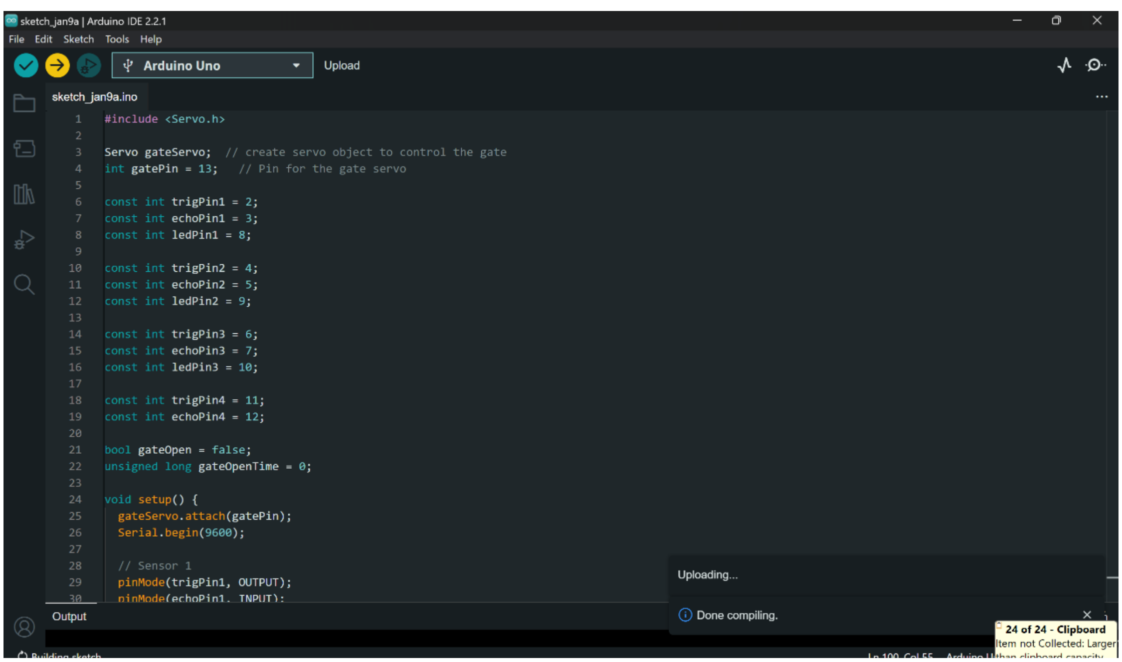

Figure 4: Picture of test 1.

Test 2: Servo Motor Operation - Gate Opening and Closing

Test	2
Objective:	To validate that the servo motor successfully raises the barrier upon detecting a car with the ultrasonic sensor and closes it after a delay.
Activity:	To move a car towards the parking gate, triggering the ultrasonic sensor. Observe the gate opening and closing.
Expected Result	The servo motor should raise the gate barrier upon car detection and close it after a delay.
Actual Result	The servo motor successfully raises the gate barrier upon car detection and closes it as expected after a delay.
Conclusion	The test is successful.

Table 2: Table of Test 2.

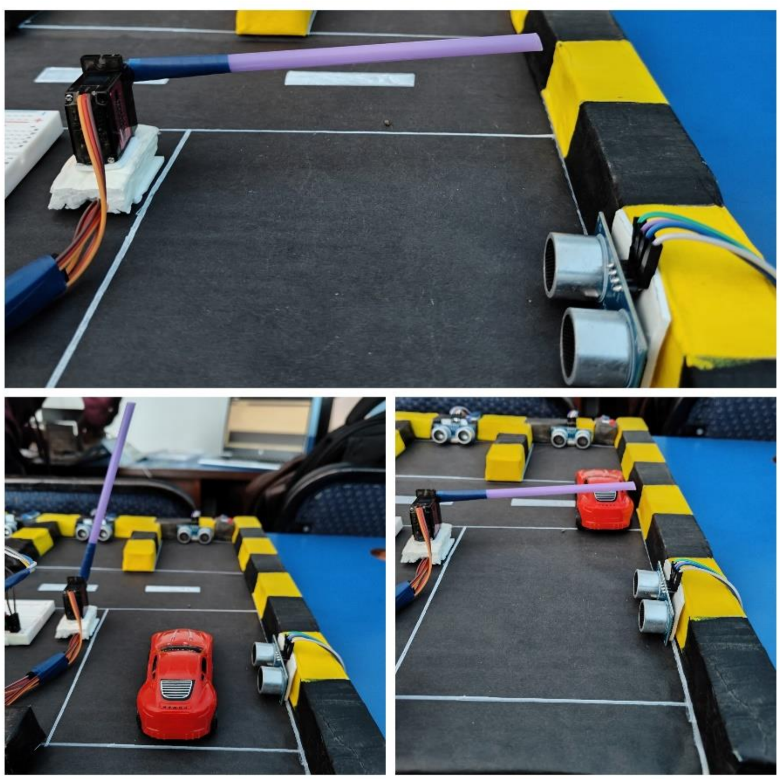

Figure 5: Picture of Test 1.

Test 3: LED Light Activation - Car Detection

Test	3
Objective:	To confirm that the all 3 LED lights activate when the ultrasonic sensor detects a car.
Activity:	To move a car to parking slot, within the detection range of the ultrasonic sensor.
Expected Result	The all 3 LED lights should turn on upon car detection.
Actual Result	The LED lights accurately activate when the ultrasonic sensor detects the car.
Conclusion	The test is successful.

Table 3: Table of test 3.

Figure 6: Diagram of Test 3.

5. Future Works

In future developments, the AI Enhanced Smart Parking Management System could incorporate a user-friendly display for parking space booking, real-time updates, optimize machine learning for better object detection, ensure scalability for larger parking areas, enhance mobile app features, integrate with smart city initiatives, implement energy-efficient measures, strengthen security, introduce a user feedback mechanism, stay abreast of IoT advancements, and engage with the local community for valuable insights. These enhancements aim to make the system more versatile, sustainable, and responsive to evolving user needs and urban challenges.

6. Conclusion

In conclusion, our "AI Enhanced Smart Parking Management System" successfully addresses the challenges of inefficient parking management in urban areas. By integrating Arduino Uno, ultrasonic sensors, and deep learning models, we automated parking gates, accurately assessed space availability, and provided real-time parking status updates through cloud integration and a user-friendly smartphone app. The system not only optimizes parking processes but also contributes to effective traffic and resource management. Our collaborative effort, guided by insightful input from our module teachers and dedicated teamwork, resulted in the successful development of an innovative solution that enhances parking efficiency and user experience.

7. References

- annapurnaexpress, 2022. *annapurnaexpress.* [Online]
  Available at: https://theannapurnaexpress.com/news/kathmandus-parking-problems-bad-to-worse-30134/
  [Accessed 11 12 2023].

- Sakar Pahari, S. K. U. K., 2020. On-Street Parking in inner Kathmandu. ON-STREET PARKING IN INNER KATHMANDU: DETERMINING ON-STREET PARKING LOAD AND PARKING EFFICIENCY OF MOTORCYCLES IN THE STREETS OF NEW ROAD IN KATHMANDU METROPOLITAN CITY, 7(8), p. 15.

- Seager, C. (2001). Microcontroller developments. *Electronics Education*, 2001(1), pp.11–14. doi: https://doi.org/10.1049/ee.2001.0008.

- Nakamura, K. (2012). Ultrasonic transducers : materials and design for sensors, actuators and medical applications. Cambridge, UK: Woodhead Publishing.

- Kusuda, Y. (1999). New generation servomotors. Assembly Automation, 19(3), pp.190–196. doi:https://doi.org/10.1108/01445159910280001.

- Lee, J.-W. and Kwon, H.-H. (2015). A Study on Energy Savings When replacing Performing Halogen Lights with LED Lights in Seoul Community Hall. The Journal of The Institute of Internet, Broadcasting and Communication, 15(5), pp.213–217. doi:https://doi.org/10.7236/jiibc.2015.15.5.213.

- Electronics Notes (n.d.). What is a Resistor: Types, Uses, Details» Electronics Notes. [online] www.electronics-notes.com. Available at: https://www.electronics-notes.com/articles/electronic_components/resistors/resistor-types.php [Accessed 14 Dec. 2023].

- all3dp (2022). What Is Tinkercad? – Simply Explained. All3DP. [online] 14 Nov. Available at: https://all3dp.com/2/what-is-tinkercad-simply-explained/ [Accessed 14 Dec. 2023].

- Silaparasetty, N. (2020). Machine learning concepts with Python and the Jupyter Notebook environment : using Tensorflow 2.0. Berkeley, Ca: Apress.

- Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [online] doi:https://doi.org/10.1109/cvpr.2016.91.

8. Appendix

8.1 Source code

```
#include <Servo.h>

Servo gateServo;  // create servo object to control the gate
int gatePin = 13;   // Pin for the gate servo

const int trigPin1 = 2;
const int echoPin1 = 3;
const int ledPin1 = 8;

const int trigPin2 = 4;
const int echoPin2 = 5;
const int ledPin2 = 9;

const int trigPin3 = 6;
const int echoPin3 = 7;
const int ledPin3 = 10;

const int echoPin4 = 12;

bool gateOpen = false;
unsigned long gateOpenTime = 0;

void setup() {
  gateServo.attach(gatePin);
```

```
  Serial.begin(9600);

  // Sensor 1
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(ledPin1, OUTPUT);

  // Sensor 2
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  pinMode(ledPin2, OUTPUT);

  // Sensor 3
  pinMode(trigPin3, OUTPUT);
  pinMode(echoPin3, INPUT);
  pinMode(ledPin3, OUTPUT);

  // Sensor 4
  pinMode(trigPin4, OUTPUT);
  pinMode(echoPin4, INPUT);

  gateServo.write(162);
}

void loop() {
  // Check ultrasonic sensors and control LEDs
  measureAndControl(trigPin1, echoPin1, ledPin1);
  measureAndControl(trigPin2, echoPin2, ledPin2);
```

```
  measureAndControl(trigPin3, echoPin3, ledPin3);
  measureAndControl(trigPin4, echoPin4, 0);  // Use 0 as the LED pin for the fourth sensor

  // Check if object is detected by the fourth sensor, open the gate
  if (gateOpen) {
    if (millis() - gateOpenTime >= 2000) {
      closeGate();
    }
  }
}

void openGate() {
  gateServo.write(90);  // Open the gate
  gateOpen = true;       // Set the flag
  gateOpenTime = millis();  // Record the time
}

void closeGate() {
  gateServo.write(162);    // Close the gate
  gateOpen = false;       // Reset the flag
}

void measureAndControl(int trigPin, int echoPin, int ledPin) {
  long duration, distance;

  // Send ultrasonic pulse
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
```

```
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Measure the echo duration
  duration = pulseIn(echoPin, HIGH);

  // Calculate distance in centimeters
  distance = (duration * 0.0343) / 2;

  // Check distance and control the LED (for the fourth sensor)
  if (ledPin == 0 && distance < 10) {
    openGate();  // Open the gate when an object is detected by the fourth sensor
  } else {
    // Control the LEDs for the first three sensors
    if (distance < 10) {
      digitalWrite(ledPin, HIGH);  // Turn on the LED immediately
    } else {
      digitalWrite(ledPin, LOW);   // Turn off the LED
    }
  }
}
```

8.2 Screenshots of the system

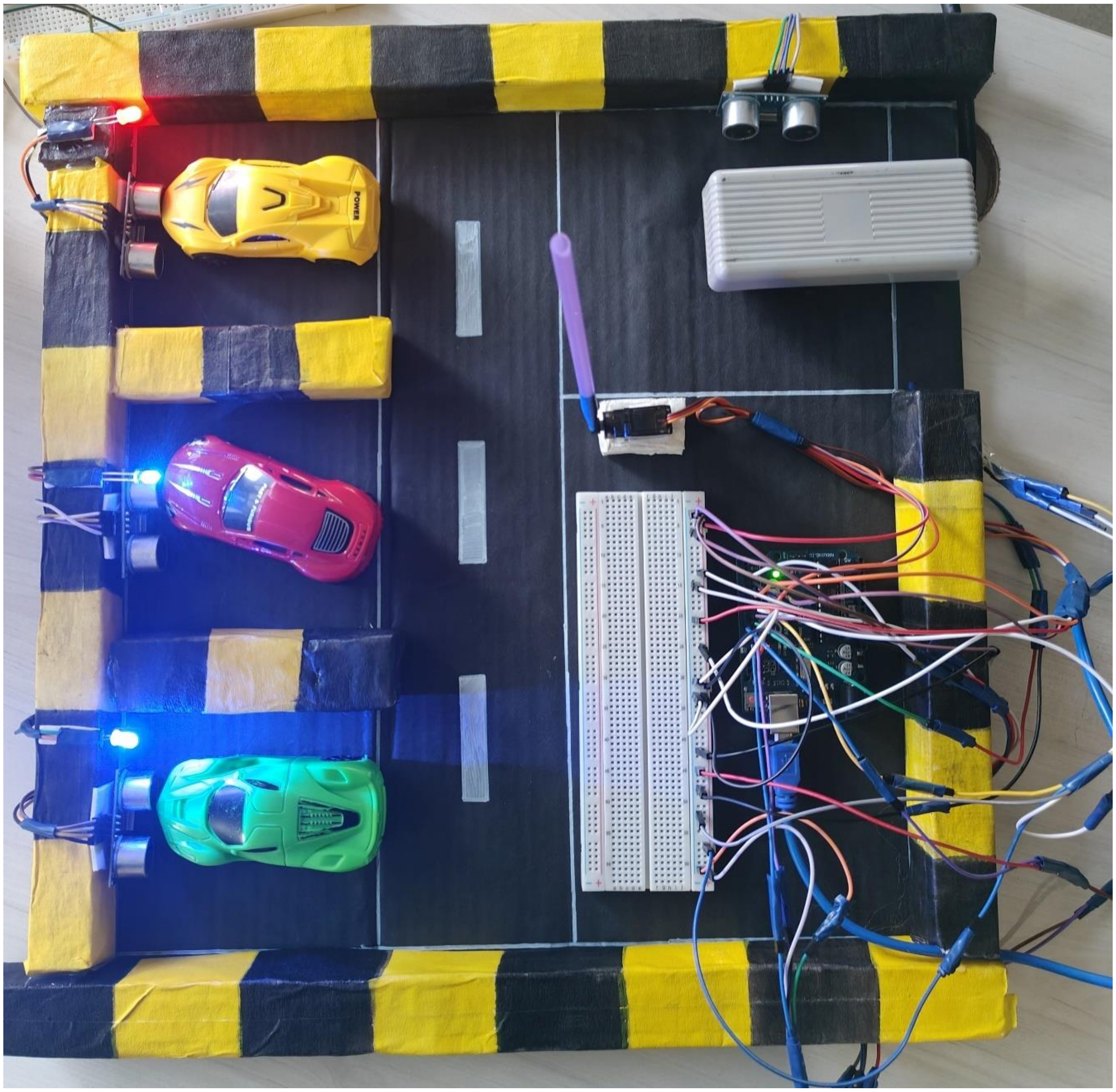

Figure 7: Screenshot of entire system.

8.3 Design Diagrams

The below diagram shows our system incorporates Cloud and Artificial Intelligence Technology. In our system, parking spaces are identified using cameras, and the data is transmitted to a deep learning model responsible for detecting vehicles in parking slots. The deep learning model communicates the details of available spaces to the cloud. Users can access real-time parking availability information on their smartphones through a dedicated application connected to the cloud.

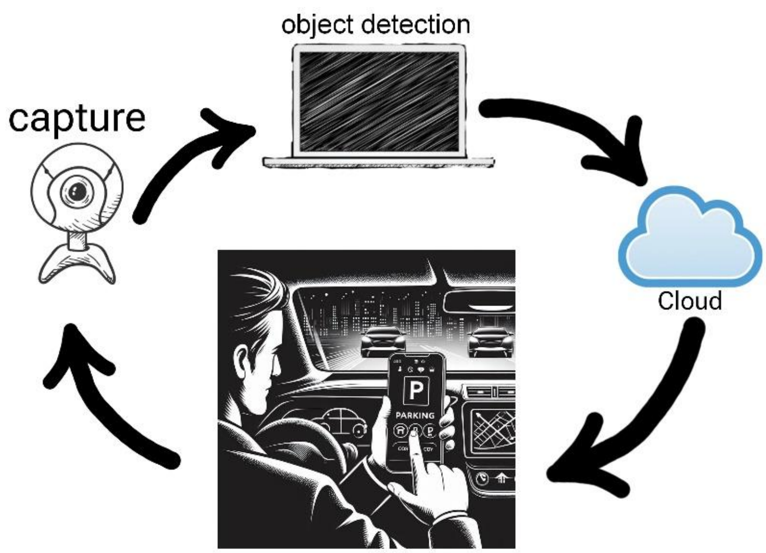

Figure 8: Diagram of Data Flow Using AI, Cloud and IOT

Hardware

2.3.1.1 Arduino Uno: The Arduino Uno is an open-source microcontroller board based on the Atmega328P microcontroller. It features digital and analog input/output pins, a USB interface for programming and power, and is widely used for prototyping and creating interactive electronic projects (Seager, 2001).

Figure 9: Figure of Arduino Uno

2.3.1.2 Ultrasonic Sensor: A transducer device that emits high-frequency ultrasonic waves and detects their reflections to measure distances or detect objects. It operates based on the principle of sending and receiving ultrasonic signals, with applications ranging from proximity sensing in robotics to measuring distances in industrial and automotive systems (Nakamura, 2012).

Figure 10: Figure of Ultrasonic Sensor

2.3.1.3 Servo Motor: A servo motor is a specialized motor that uses feedback for precise control of its rotational or linear position. It consists of a motor, a feedback sensor, and a control circuit. The feedback system continuously monitors the actual position of the motor and adjusts it to match the desired position (Kusuda, 1999).

Figure 11: Figure of Servo Motor

2.3.1.4 LED Light: LED stands for "light-emitting diode." LED lights are a type of lighting technology that uses semiconductor devices to emit light when an electric current passes through them. Unlike traditional incandescent bulbs, which use a filament to produce light, LEDs generate light through the movement of electrons in a semiconductor material (Lee and Kwon, 2015).

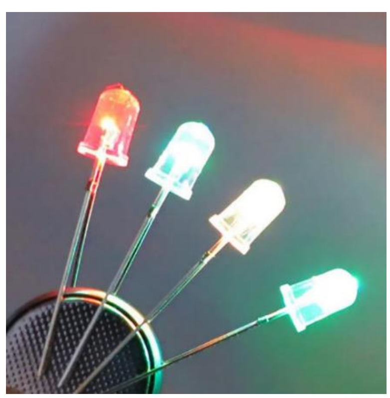

Figure 12: Figure of LEDs

2.3.1.5 Jumper Wires: Jumper wires are electrical wires used to create connections between different components on a breadboard or within an electronic circuit. They are typically insulated wires with connectors on either end that can be easily inserted into the holes of a breadboard or connected to the pins of electronic components (Nakamura, 2012).

Figure 13: Figure of Jumper Wires

2.3.1.6 Resistors: Resistors are electronic components designed to introduce a specific amount of electrical resistance into a circuit. Resistance is a property that opposes the flow of electric current. Resistors are commonly used in electronic circuits to control the amount of current, divide voltages, and protect components(Electronics Notes, n.d.).

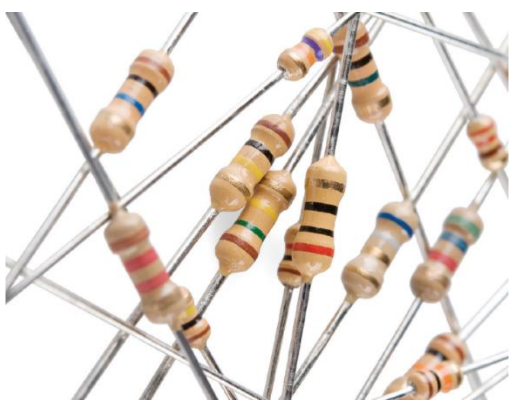

Figure 14: Figure of Resistors

Software

2.3.2.1 Arduino IDE: Arduino IDE is an open-source software for writing, compiling, and uploading code to Arduino microcontroller boards. It supports various Arduino models, providing a user-friendly platform for IoT development.

2.3.2.2 Tinkercad: Tinkercad is a user-friendly online platform designed for 3D design and electronics projects. It simplifies the creation of digital prototypes through its intuitive drag-and-drop interface. Widely embraced by beginners and hobbyists in the IoT community, Tinkercad facilitates the easy design of 3D models and simulation of circuits, making it an accessible tool for bringing creative ideas to life virtually (all3dp, 2022).

2.3.2.3 Jupyter Notebook: Jupyter Notebook is an open-source, web-based application enabling interactive coding, data analysis, and visualization. Widely used with its support for multiple languages, especially Python, it's a go-to tool for IoT projects and machine learning tasks (Silaparasetty, 2020).

2.3.2.4 Microsoft Word: Microsoft Word, or MS Word, is a widely-used word processing software by Microsoft. Known for its user-friendly interface and versatile features, it simplifies simplifies document creation and editing, making it essential in professional and academic setting.

Individual Contribution plan:

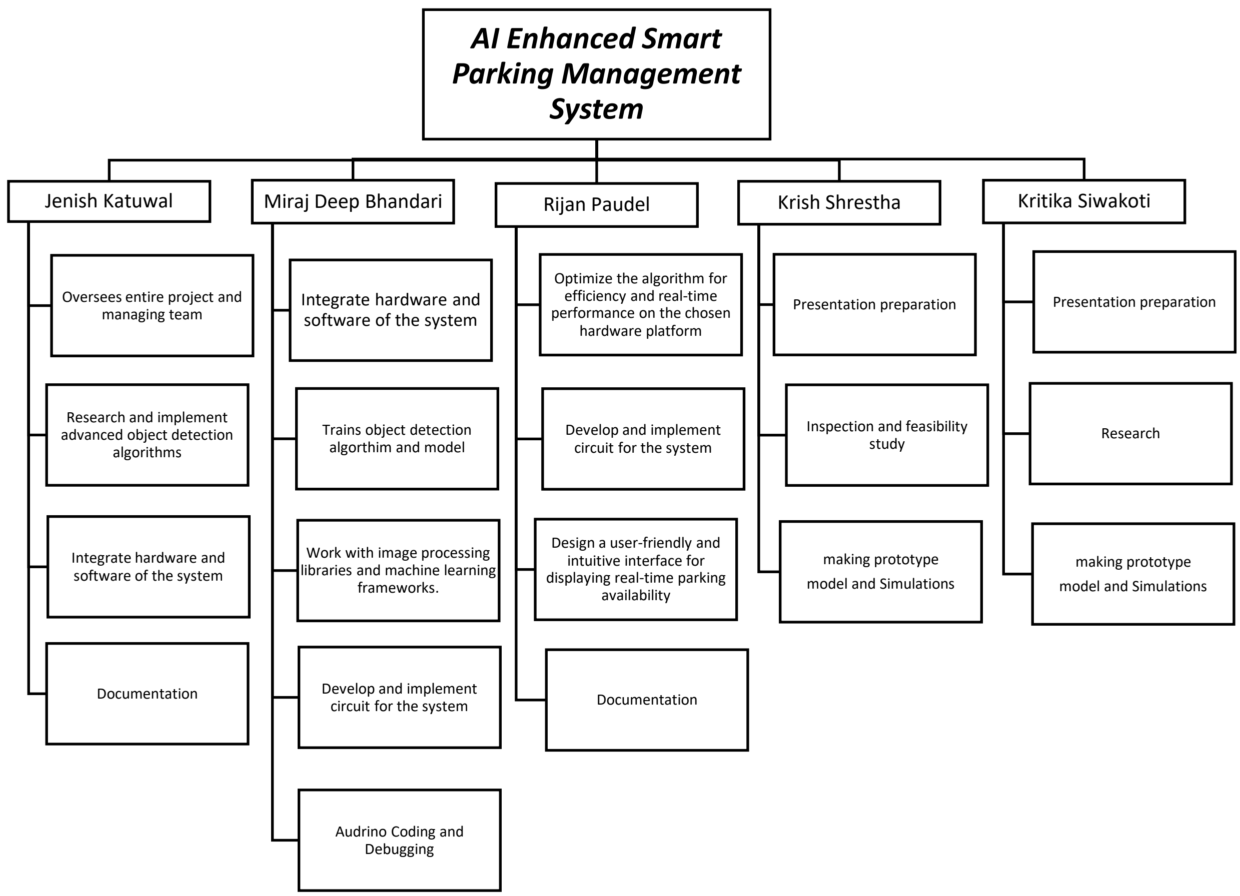

Figure 15: Individual Contribution plan: