Title : Smart Security Bicycle Rental system for Shopkeepers

Norafiqah Balqis binti Sulaiman, Faisal bin Hazry, Lee Xing Yang, Zakaria Mohamad Osman
Universiti Teknologi Malaysia
norafiqahbalqis@graduate.utm.my, faisal8@graduate.utm.my, xing.yang@graduate.utm.my, Mozakaria@graduate.utm.my

Abstract: This project aims to create an app for shopkeepers to manage their bicycle rental business more efficiently and securely. The app includes features such as customer location tracking and a digital logbook, which allows the shopkeeper to register customers and store their information electronically, eliminating the need for a physical logbook. In addition, the app includes a user interface on the bicycles that allows customers to interact with the system and extend their rental time without the need to communicate with the shopkeeper. The system also includes a tracker that allows the shopkeeper to track the location of the bicycles at all times. To ensure the security of the system, the app includes a notification system that alerts the shopkeeper if the bike is being misused by the customer. Overall, this project aims to streamline the bicycle rental process for both shopkeepers and customers, while also improving the customer experience and ensuring the security of the system.

Keywords: *EECS; IOT; Thignspeak; MIT APP; arduino uno; GSM*

1. Introduction (Project or Innovation)

A smart security bicycle rental system is a modern, technology-driven solution for renting bicycles. It involves using electronic locks and/or GPS tracking to secure the bikes, as well as digital payment systems for rent transactions. The goal is to provide a convenient, safe, and efficient way for people to rent bicycles for short periods of time, for both personal and commercial purposes. This type of system is often used in urban areas and tourist destinations, where there is a high demand for bike rentals.

The smart security bicycle rental system features cutting-edge technology designed to provide a secure, user-friendly, and efficient rental experience. The system includes a robust security system that ensures the safety of both the bicycles and the users, as well as a GPS tracking system that allows the bikes to be easily located and monitored in real-time. The system also includes a digital cloud logbook that provides a secure and transparent record of all rental transactions, ensuring that both the bikes and the users are properly accounted for. The system is further enhanced by a user-friendly app that provides easy access to all of the system's features and functionalities, making it simple and convenient for users to rent and return bikes. All of these features work together to provide a comprehensive and seamless smart security bicycle rental solution.

1.1.Objective

The objective of the project is to create a convenient user-friendly system that can perform certain functions. First, it allows bicycle shopkeepers to view the customers' location in real time. Besides, it should remind the customers of the remaining time and the shopkeeper will get a notification when the time is finished. This project is also aimed to make the whole renting process faster.

1.2.Questionnaires

A survey is conducted to a bicycle rental shopkeeper in Universiti Teknologi Malaysia, and one in Danga Bay to better understand the needs of the shopkeeper.

No.	Questions
1.	What is the current procedure for customers to rent a bicycle?
2.	Do you have any problem with the customer who rents your bicycle?
3.	Do you have any problem with customers who call the owner's shop saying that their bicycle breaks down on the road?
4.	Is it easy to detect if customers do not return the bike?
5.	Do you have any experience catching customers stealing bicycles red-handedly?

1.3.Analysis of Data Collection

The data from the Google Form is thoroughly reviewed, and the findings are compiled on the Jamboard. The members then see through the responses and analyse them in different aspects, such as current situation, problem faced, potential solutions, and steep analysis.

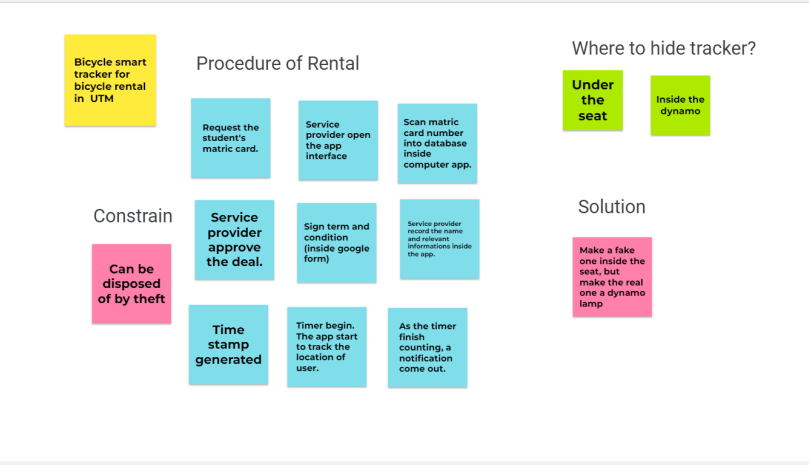


Figure 1: Data Collection Analysis On Jamboard

1.3.1. Problem Statement

First, the rented bicycles can be misused and stolen. Besides, the customers can exceed the rental time. Moreover, a physical logbook is hard to manage as it is prone to natural disaster or loss. Last but not least, the shopkeepers cannot hold on to customers' IC cards.

1.3.2. Needs

- Gps system
- To hire more staff to look after the customers
- Automatic alarm system

1.4.Design Statement

The goal is to help Bike Rental Shop Owner to prevent losing and misusing of their rented bicycles. For customers who do not return the bike, location tracking feature is needed to make the process of locating the bike much faster, hence reduces the possibility of losing potential customers due to the unattended counter because of short-staffed. The system also has a feature to prevent GPS system from being tampered. Furthermore, the system shows the customer their remaining rent time and distance traveled. Last but not least, the customer can immediately send a emergency signal to the shopkeeper if anything bad happen such as accidents or if the bike breaksdown.

2. Project Planning

Currently method using

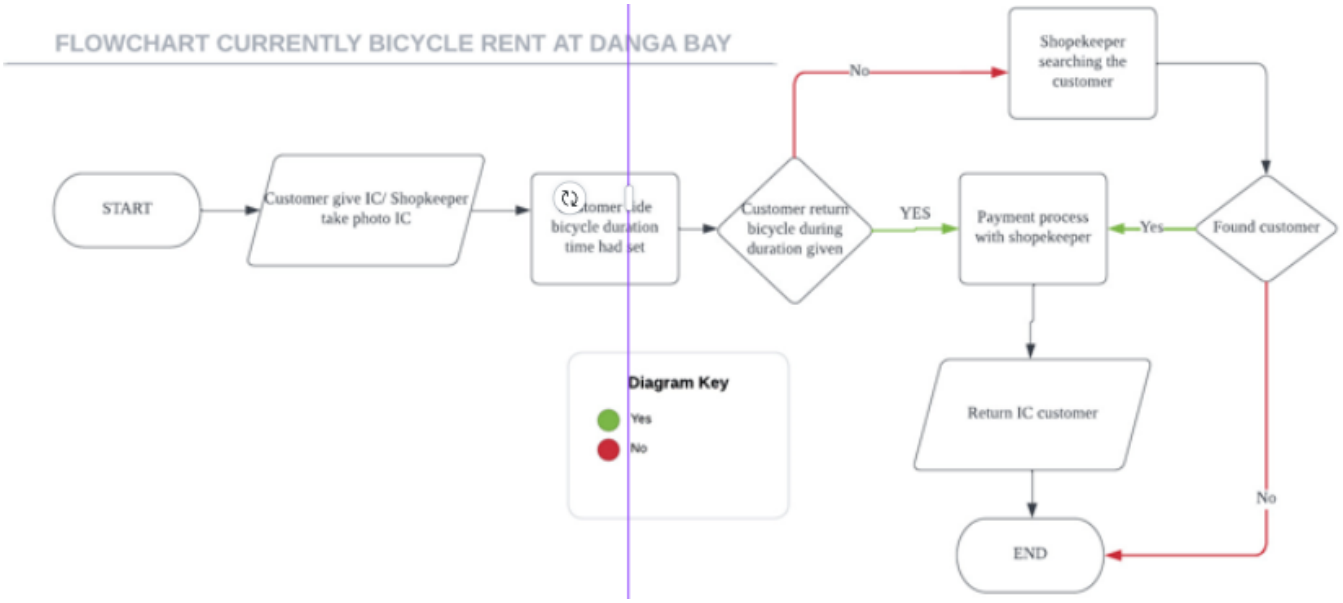


Figure 2: FlowChart Currently Bicycle Rent at Danga Bay

Suggestion new method

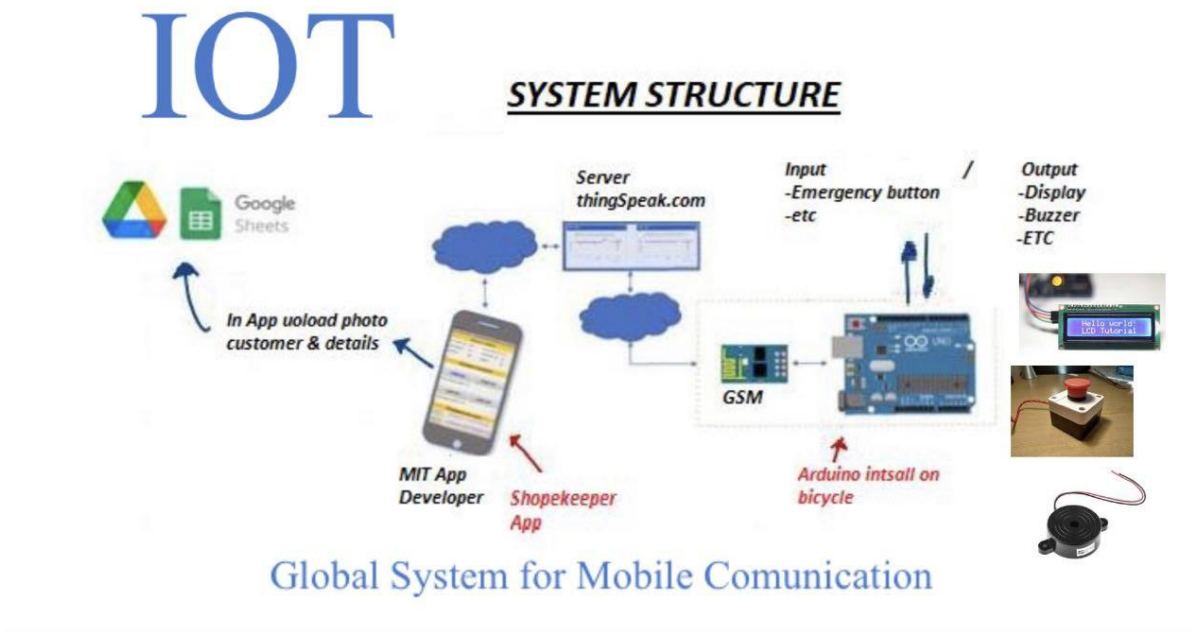


Figure 3: FlowChart Project Plan Bicycle Rent at Danga Bay Comunicato

2.1.Schematic Diagram

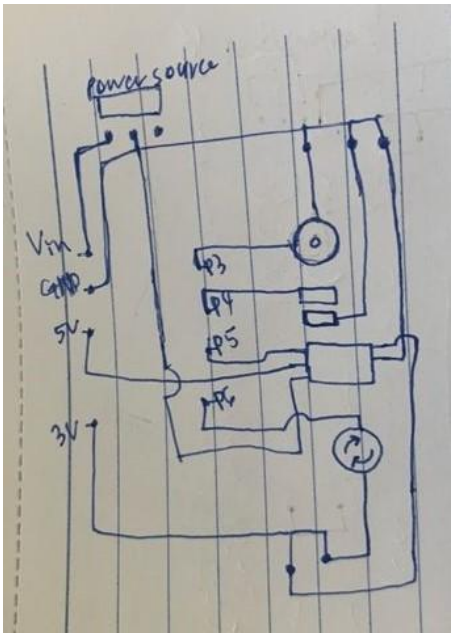


Figure 4: Schematic Diagram for Security System Circuit.

2.2.Circuit Design

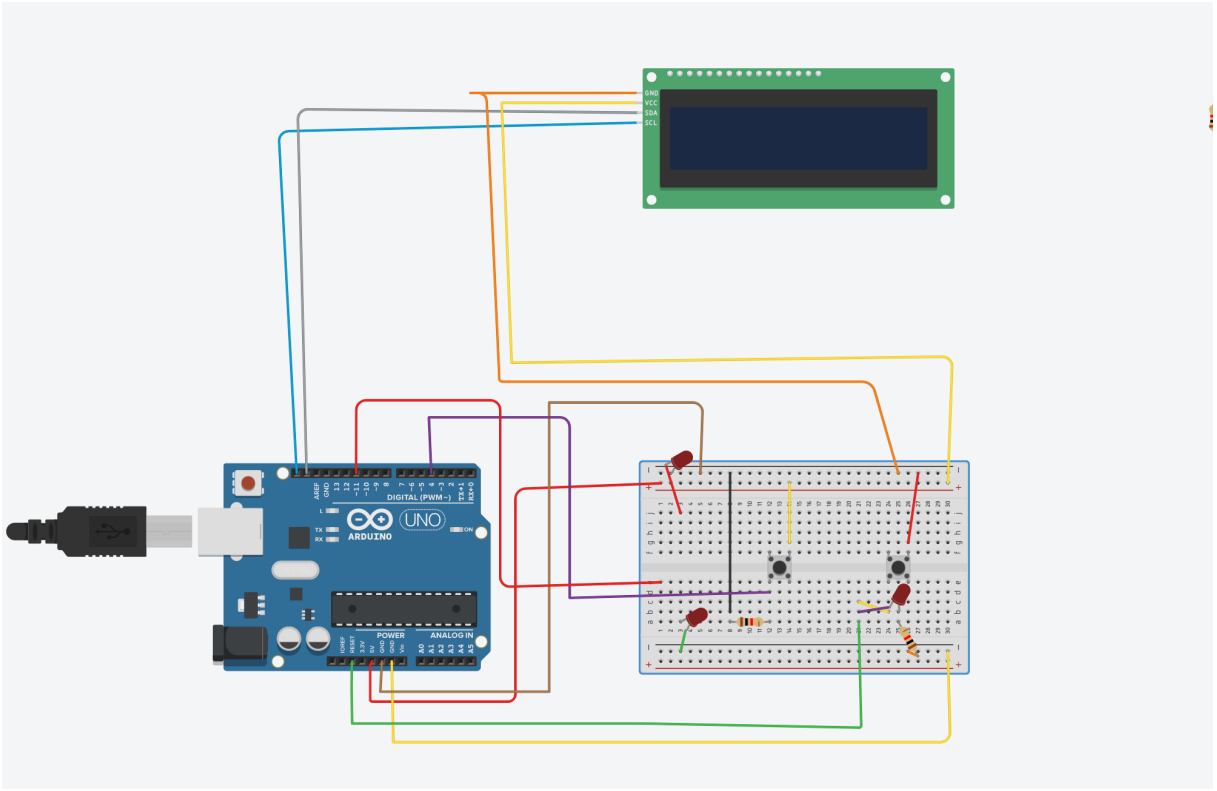


Figure5: Schematic Diagram for user interface system

2.3. Expected Cost

No	Component	Estimated Cost			Actual Cost
		Unit Price (RM)	Qty.	Subtotal (RM)	Delivery Week and Remark
1	Arduino Nano V3.0 Compatible (CH340G) (Shoppe)	26.50	2	53.00	Shoppe (GI Electronic Receipt 1)
2	Female Header 1X40 Arduino	0.70	1	0.70	Shoppe (GI Electronic Receipt 1)
3	GY-Neo6MV2 Flight Control GPS Module	21.30	1	21.30	Shoppe (GI Electronic Receipt 1)
4.	Straight Female Header 1x40ways	1.2	2	2.4	Top One
5.	Terminal Block DG128V-03 Green	1	1	1	Top one
6.	KF350-2 Terninal Block	0.40	3	1.2	Top one
7	Push Button	1	5	5	Top one
8	Donot Board	1.6	2	3.2	Top one
9	Finex Solder lead	3.8	1	3.8	Top one
10	Cable 1.0mm Black	3.0	1	3	Top one
11	Cable 1.0mm Red	3	1	3	Top one
12	Bongkah 12 way	2	1	2	Top one
13	Straight Female header 1x40 way	1.2	1	1.2	Top one
14	Single core cable `	0.5	3	1.5	Top one
15	Shrink tube 3mm	1.5	1	1.5	Top one
16	Shrink tube 2mm	1.2	1	1	Top one
17	Arduino nano	31.5	1	31.5	Top one
18	Donut Board	1.6	1	1.6	Top one
19	GSM 900a	29.90	1	29.90	Shoppe
			Total	167.8	

Figure6: Expected cost

2.4.Gantt Chart

UTM

SKEE 4723 CAPSTONE PROJECT
GANTT CHART

SKE-4723-2021-09

Present this gantt chart to your supervisor to get it endorsed on Week 6.

Group: C2G02

Please tick the checkbox upon task completion

No	Tasks	Start Week	End Week	Capstone Week 5	6	7	8	9	10	11	12	13	14	Remark upon Task Completion
1	Presentation day	5	5	☐										
2	Discussing divided task department	5	6	☐	☐									
3	designing Communication system	6	7		☐	☐								
4	Receive latitude and longitude in app	7	7			☐								
5	Implementing map feature in app	7	7			☐								
6	Designing Security System Circuit	7	7			☐								
7	Creating code LCD time countdown	7	7			☐								
8	Constructing the security system circuit	8	9				☐	☐						
9	Designing Communication system and server for sensor data to server	8	9				☐	☐						
10	Implement app interface - make a map	8	8				☐							
11	Display the count down and mode it 900s	8	8				☐							
12	Implement app interface - collect multiple data from Thinks	8	8				☐							
13	Coding reset count down when finish serial monitor	8	8				☐							
14	Constructing the security system circuit	9	9					☐						
15	Finalizing the communication system into PCB board	9	9					☐						
16	Implement app interface -implement a timer counter inside app	10	10						☐					
17	Customize the return button	10	10						☐					
18	Real-Time Monitoring of Emergency and Bridge Signals us	10	10						☐					
19	Overcoming Challenges in System Integration and Testing	11	11							☐				
20	Editing video project demo version 2 & Seminar presentation day	11	11							☐				

Figure7: Ghant chart

3. Engineering Design

3.1.Component Design

Security System

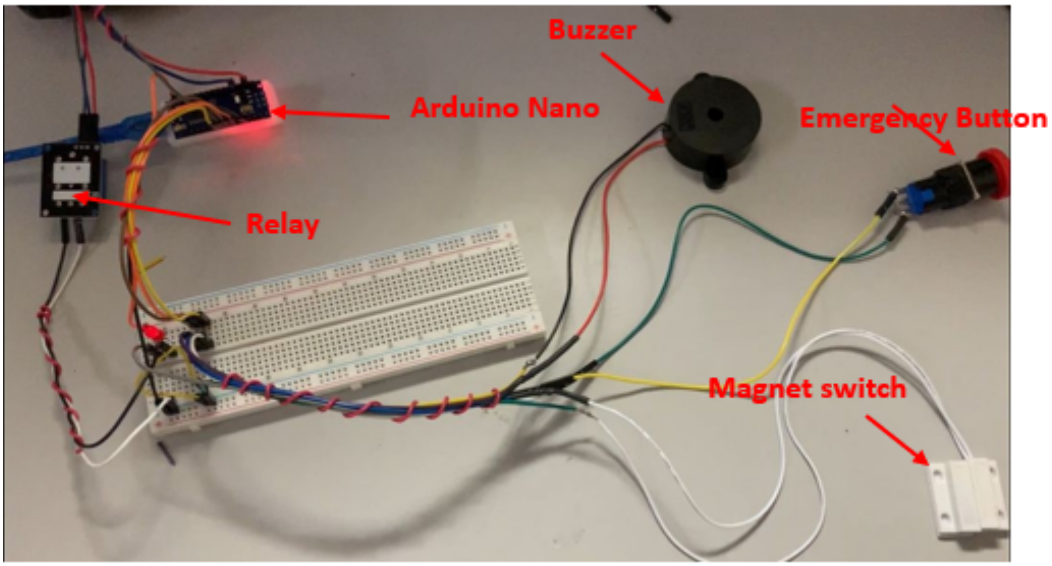


Figure 8: Security System Hardware

1. Arduino Nano: A small, complete, and breadboard-friendly board based on the ATmega328P (Arduino Nano 3.x) or ATmega168 (Arduino Nano 2.x). It has similar functionality to the Arduino Duemilanove, but with a different form factor.
2. Relay: Electrically operated switch that is used to control the flow of current in a circuit.
3. Buzzer: An electrically operated switch that is used to control the flow of current in a circuit.
4. Emergency Button: A device that is used to quickly alert people in case of an emergency or danger.
5. Magnetic Sensor: A device that detects the presence and strength of a magnetic field. Detecting the current flowing in an electrical conductor by sensing the magnetic field generated by the current.

In normal operation (box closed) the magnet switch will be closed and the relay will be on the normally closed mode to keep the electrical bicycle supplied by a power source. When the security is breached (box opened), causing the magnet switch to be open, causing a high signal to be sent to Arduino Nano causing the buzzer to go off and the relay will be cutting off the power source while sending an alarm to the shopkeeper. In a case of emergency, the user can push the emergency button down. This action will send a high signal to the Arduino Nano and causing the buzzer go off then the shopkeeper will be notified through the MIT application on their phone.

User interface (UI) System

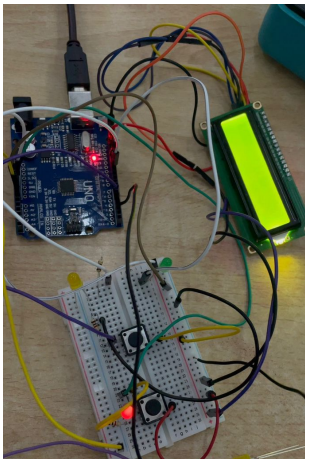


Figure 9: UI System Hardware

1. Arduino Board: The main microcontroller used to run the program
2. LCD Screen: Display for showing information about the countdown timer
3. Buttons: Used for user input and control of the countdown timer
4. LiquidCrystal_I2C Library: Used for interfacing with the LCD screen
5. buttonPin1 & buttonPin2: Digital pins used to read the state of the buttons
6. buzzer: An electronic device used to generate a sound to indicate the end of the countdown
7. signalStart & systemready: Output signals used to indicate the state of the system (start or ready)
8. countdownValue: A variable used to store the value of the countdown timer
9. countdown: A flag used to indicate if the countdown is running
10. setup function: Initialises the LCD screen, sets up the buttons and signals and defines them as inputs or outputs
11. loop function: Reads the state of the buttons and performs actions based on their state (start the countdown, show "System Ready" on LCD, or countdown)

The program uses an Arduino board, an LCD screen and buttons to control a countdown timer. In the setup phase, it initialises the LCD screen, sets up the buttons and signals and defines them as inputs or outputs. In the loop phase, the program reads the state of two buttons and determines the actions to take based on the button states. When buttonPin1 is high and the countdown isn't running, the LCD is cleared, countdown starts and the value of countdownValue is set to 10. The signalStart is set to high and systemready to low. If buttonPin1 is low, the program displays "System Ready" on the LCD, sets the signalStart to low, and systemready to high. During the countdown, a for loop counts down from the value of countdownValue and displays each second on the LCD. When the countdown finishes, the buzzer is turned on for multiple intervals and the LCD displays "Time Finish" and "Do you want to continue renting." The code doesn't handle buttonPin2 and its related functions

Communication system

Figure 10:Prototype System Communication

1. **Arduino Nano Microcontroller:** This is a small, low-cost microcontroller that provides the necessary inputs and outputs for the system. It acts as the central processing unit of the system, communicating with other components such as the GPS module, security sensors, and interface systems.
2. **GSM 900a:** This is a cellular communication module that allows for data transfer between the system and the ThingSpeak server. It receives data from the security and interface systems and sends the data to the server for storage and processing.
3. **GPS Module:** This module receives data on the location of the bicycle and sends it to the microcontroller for processing and storage on the ThingSpeak server.
4. **Security Sensors:** These are devices that detect changes in the environment, such as movement or changes in temperature. They provide the necessary data for the security system to operate properly.
5. **Interface Systems:** These systems control the user interface, such as buttons and display screens, to allow the user to interact with the system. They provide data on the rental status of the bicycle and send it to the ThingSpeak server for processing.

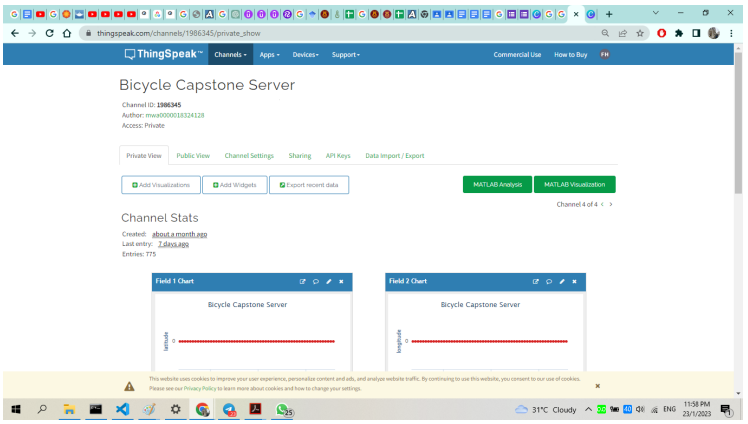


Figure 11: Thingspeak Server Data

- Field 1 and Field 2 on the server will be used to store the longitude and latitude coordinates of the bicycle
- Data in Field 3, 4, 5, and 6 will be sent to the server as high and low data
- Field 3 and Field 4 will also be used to store data from emergency buttons and security sensors.
- Field 5 and Field 6 will be used for the interface system, specifically for

recording the start of a rental and when the system is ready

3.2. Flowchart

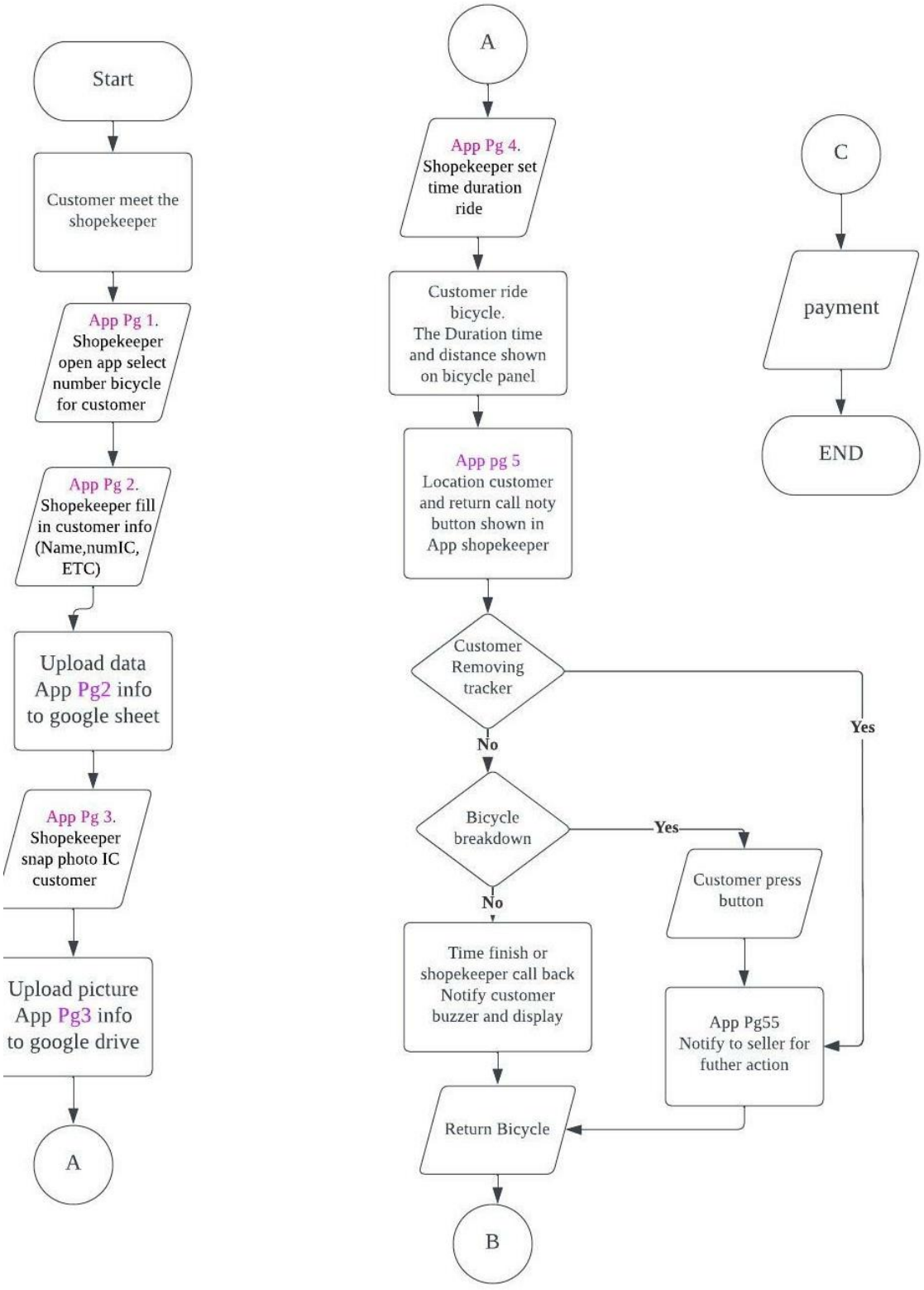
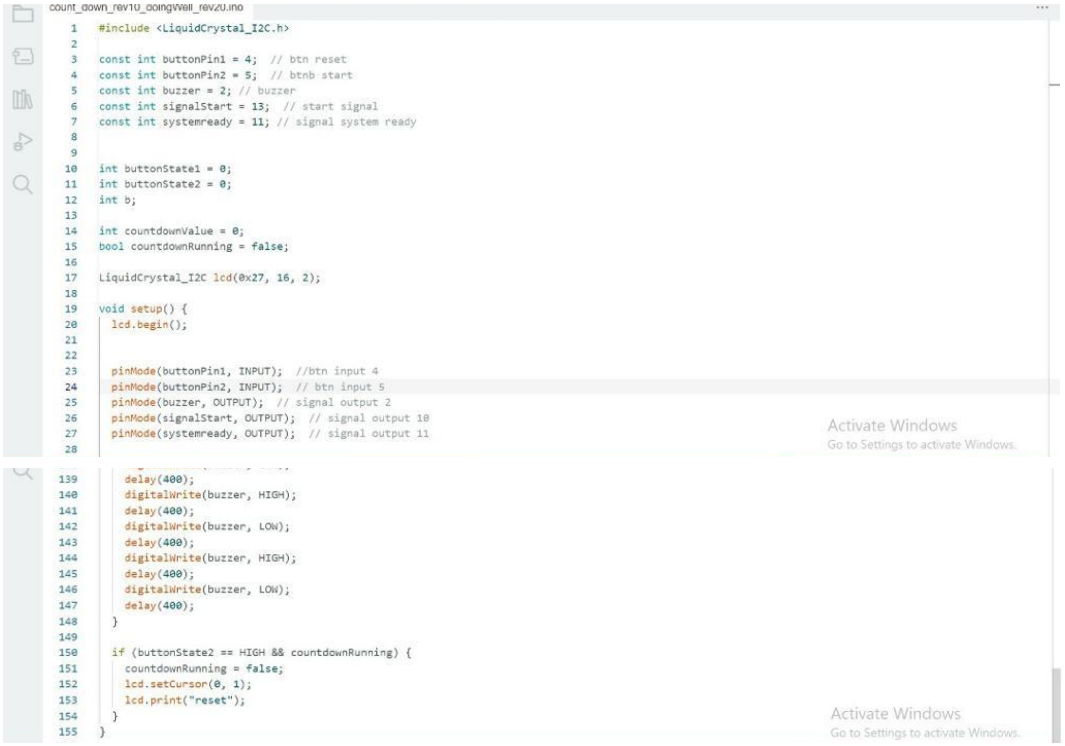


Figure 12: Flow chart system

3.3. Coding



```
#include <LiquidCrystal_I2C.h>

const int buttonPin1 = 4;  // btn reset
const int buttonPin2 = 5;  // btnb start
const int buzzer = 2; // buzzer
const int signalStart = 13;  // start signal
const int systemready = 11; // signal system ready


int buttonState1 = 0;
int buttonState2 = 0;
int b;

int countdownValue = 0;
bool countdownRunning = false;

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.begin();


  pinMode(buttonPin1, INPUT);  //btn input 4
  pinMode(buttonPin2, INPUT);  // btn input 5
  pinMode(buzzer, OUTPUT);  // signal output 2
  pinMode(signalStart, OUTPUT);  // signal output 10
  pinMode(systemready, OUTPUT);  // signal output 11

    delay(400);
    digitalWrite(buzzer, HIGH);
    delay(400);
    digitalWrite(buzzer, LOW);
    delay(400);
    digitalWrite(buzzer, HIGH);
    delay(400);
    digitalWrite(buzzer, LOW);
    delay(400);
  }

  if (buttonState2 == HIGH && countdownRunning) {
    countdownRunning = false;
    lcd.setCursor(0, 1);
    lcd.print("reset");
  }
}
```

Figure 13: Codding

4. Product Realisation

4.1. Hardware

Security system	1. Arduino Nano 2. Relay 3. Buzzer 4. Emergency Button 5. Magnetic Sensor
User Interface system	1. LCD Screen 2. Arduino uno 3. Button 4. Buzzer
Communication system	1. GSM 900a 2. Arduino Nano 3. GPS Neo-6
Photo Taking	1. Handphone camera

4.2.Software

Application for shopkeeper	Mit App Inventor
Server for communication	ThingSpeak
Data storage	Google Drive, google sheet

4.3.Final Product

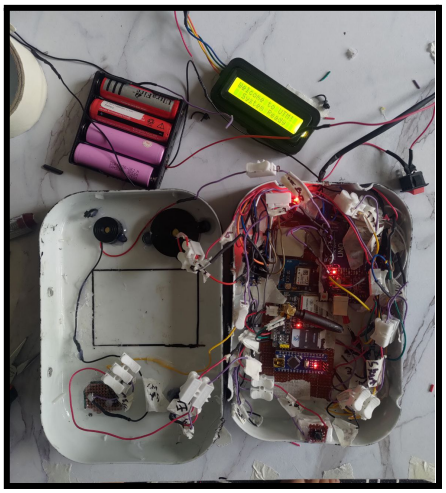


Figure 14 and 15: Smart security system

5. Results and Discussions

5.1.System Test Functionality

This testing was conducted to determine the system successfully integrated with all systems. Figure below show result we for conducted from the project

Emergency Signal	Bridge Signal	MIT App Display
High	High	System Interrupted
High	Low	Emergency Activated
Low	High	System Interrupted
Low	Low	Normal Operation

Figure 16: Truth table Emergency signal and Bridge Signal

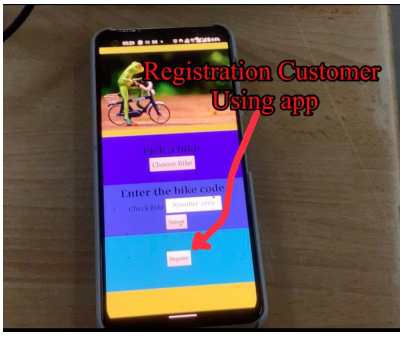
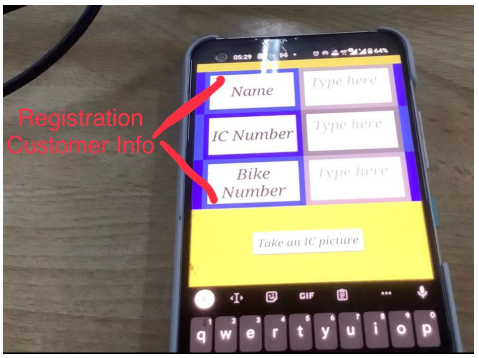
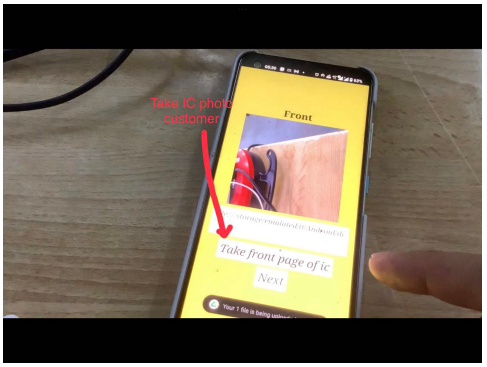
Signal System Ready	Signal Start Rental	MIT App & UI display
High	Low	Not Renting (System ready)
Low	High (rental time 10min)	Start Renting
Low	Low	Continue Renting?

Figure 17: Truth table system ready signal and start rental Signal

5.2.Procedure

We successfully integrated the system with the app, resulting in a fully functioning setup. The figure below showcases all the features and functions in the app that are now operational.

Registration Customer

MIT App & UI display	Outcome Final Result
Home page APP • Press registration for registration customer	
Register customer information • Fill their name, IC and bike number using for rent	
Register customer information • Snap photo IC customer	

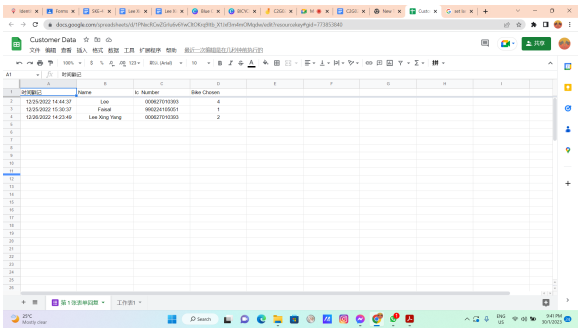
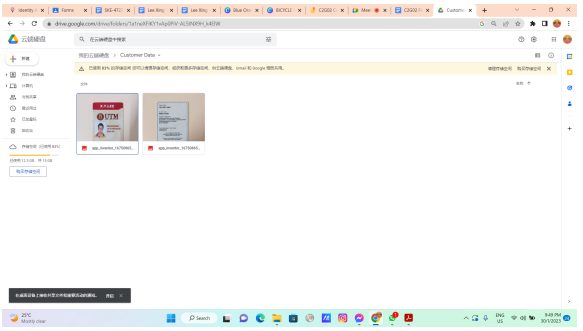
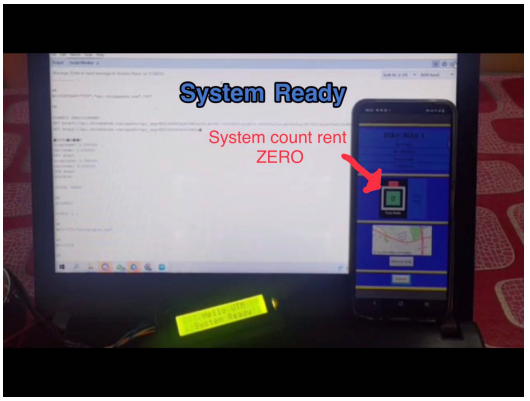
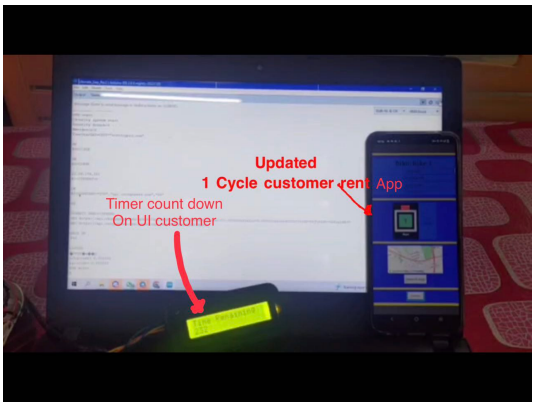
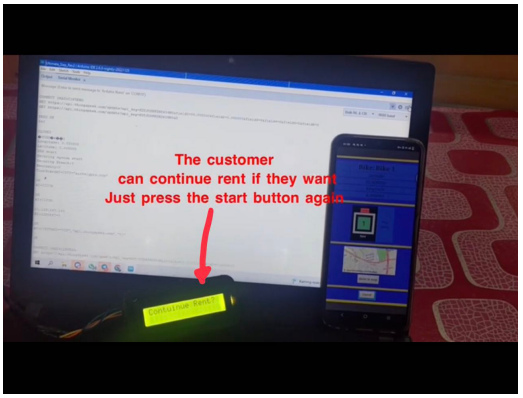
Digital log book • Customer information will be store in google sheet	
Google Drive • Customer's IC will be stored on google drive	

Figure 18: Procedure shopkeeper renting registration customer app system

Customer tracking rental system

MIT App & UI display	Outcome Final Result
System ready • Not Renting	
Start Renting • Rental time 10min	
Notification time finishes. Do you want Continue Renting? • The customer can continue rent if they want just press the start button again	

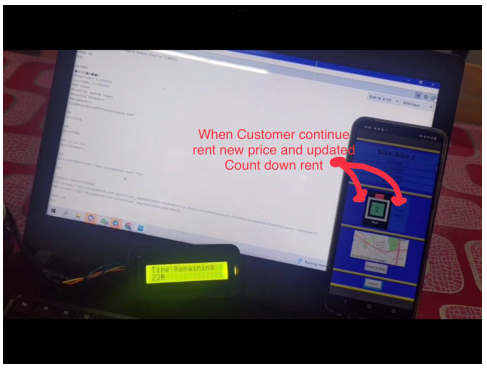
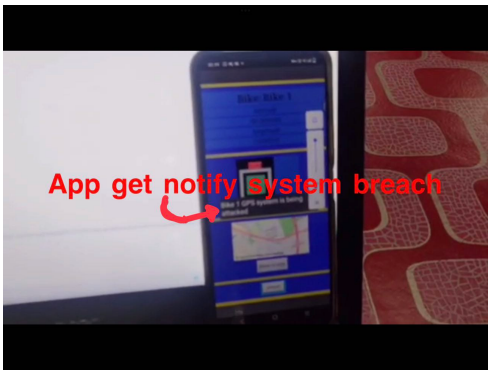
Customer continue renting • When customer continue renting the new price and updated count is update at shopkeeper app	
System breach • **App get notify when system bicycle is being breach**	

Figure 19: Procedure shopkeeper renting track customer rent app system

5.3.Observation

The GPS system is able to send the location to a satellite. However, the GPS signal is not available in an indoor environment. The gps needs some time to send the signal when it is changed from indoor to outdoor. This situation may be due to the antenna used mainly for drones and it is not suitable for bicycle usage. For further improvement, we can change the antenna. The emergency system and security system are able to activate the buzzer. The application is able to receive signals of location. The application is also able to work correctly. However, the delay of signal transmissions is long, which is 40 seconds on average.

6. Conclusion

In conclusion, the system is able to function with some drawbacks. The GPS signal needs some delay to be sent and received. This can be improved by changing the antenna. The emergency and security systems and the application are functioning correctly. However, The signal transmissions are slow. The prototype should be further improved and tested so that it can meet the commercial requirement.

7. Acknowledgement

The lead researchers extend their profound gratitude to the School of Electrical Engineering at Universiti Teknologi Malaysia for providing invaluable support during the execution of this Capstone project. Additionally, they extend their sincere recognition to their co-students, professors, and assistant engineers who actively participated in the surveys and made significant contributions to the project's success.

Video CapStone :
https://drive.google.com/file/d/18DoTET063WoyDQN3DP2vysRRyeOnmYoN/view?usp=sharing

8. References

1. Garage (2022) How to use a 1602 i2c Serial LCD Display with Arduino https://youtu.be/BJ93XCcD858
2. Alim mulydi (2022) Membuat countdown time serderhana project arduinohttps://youtu.be/g4l4SX7zMek
3. Picmicrolab (2018) Lcd countdown timer https://youtu.be/uwGG00PlY6k
4. *mit app inventor send data to thingspeak*. (2020, May 30). YouTube. Retrieved January 30, 2023, from https://www.youtube.com/watch?v=Nq4JpEpUMjk
5.

6. Lau, S. H., Tan, S. J., Fong, S. L., & A Jalil, E. E. (2017). Let's bike! The factors that influence urban cycling.
7. Termida, N. A., Zaperi, N. H., Daniel, B. D., Endrayana, D. B., & Hardini, P. (2022). Subjective Factors Influencing Students to Use Bicycle on Campus: A Case Study in Universiti Tun Hussein Onn Malaysia. International Journal of Sustainable Construction Engineering and Technology, 13(4), 121-133.
8. Read data from thingspeak.com with MIT APP INVENTOR bersama mentor Rifqi Akbar. (2021, May 23). YouTube. Retrieved January 30, 2023, from https://www.youtube.com/watch?v=Kkzpy6HfJzs
9. MIT App Inventor Built-in Blocks. (n.d.). MIT App Inventor Built-in Blocks. Retrieved January 30, 2023, from http://ai2.appinventor.mit.edu/reference/blocks/
10. Faisalhazry, F. hazry. (2022, December 13). *Sending GPS coordinate to thingspeak using GPS module neo 6 and GSM 900A*. Arduino Forum. Retrieved January 31, 2023, from https://forum.arduino.cc/t/sending-gps-coordinate-to-thingspeak-using-gps-module-neo-6-and-gsm-900a/1065052
11. Admin. “Send GSM SIM800/900 GPRS Data to Thingspeak With Arduino.” *How to Electronics*, 29 Feb. 2020, how2electronics.com/send-gsm-sim800-900-gprs-data-thingspeak-arduino.