Assignment 2

Palak Mahajan
Department of Electrical and Computer
Concordia University
Montreal,Canada
palak.mahajan@mail.concordia.ca

Adit Shah
Department of Electrical and Computer
Concordia University
Montreal,Canada
adit.shah@mail.concordia.ca

***Abstract*—The goal of the project is to prevent the theft of bicycle and bike. It is primarily intended for cyclists who ride regularly. Users would be able to track their bikes even if they were not present. GPS sensors, accelerometer sensors, gyroscopic sensors, microcontrollers, and a communication interface are among the hardware interfaces included. Data from the device's sensor is sent to the user via the user interface for communication. A push notification is a type of notification that sends a message to a user's phone. Aside from the scenario outlined above, the location's history at various time intervals would be required.**

***Index Terms*—event driven modeling, theft, gyroscope, accelerometer, gps**

I. INTRODUCTION

In today's world, vehicle safety is crucial. Vehicles parked near the market or shopping complex may not be secure, necessitating the implementation of appropriate security measures. The goal of this application is to create a system that determines a vehicle's exact location and retrieves the vehicle's location in terms of longitude and latitude in the event of theft. Also, the location's history at various time intervals would be required. The development is done for a scenario using the observer subject pattern.

II. ACTIVITY DIAGRAM

An activity diagram in UML is the representation of the work flow and the activities that are involved in a process. It represents flow of control in the system. This UML diagram is useful for modelling the dynamic aspects of the system. The rounded rectangles reflect the actions in a system. After the sensors have been activated, the tracking activity of the three sensors is represented by rectangles. Concurrent actions in the process flow are denoted by the terms fork and join. The diamonds represent the branching and merging that occurs in a system. The blackened circle indicates the start of the action, while the surrounding black circle indicates the end of the activity.

When a change in the position of the bike is detected, all of the sensors are activated and detect the change in position, angular velocity, and direction. In addition, a virtual map is available with which one can locate the bike on the phone's map and it also provides the shortest distance between the person's current location and the bike. The application retrieves all of the data from the sensor and stores it in the database. The bike's location may be tracked in real time. The bike tracking history is saved in a database for a set amount of time, and the user can view it at any moment in the future.

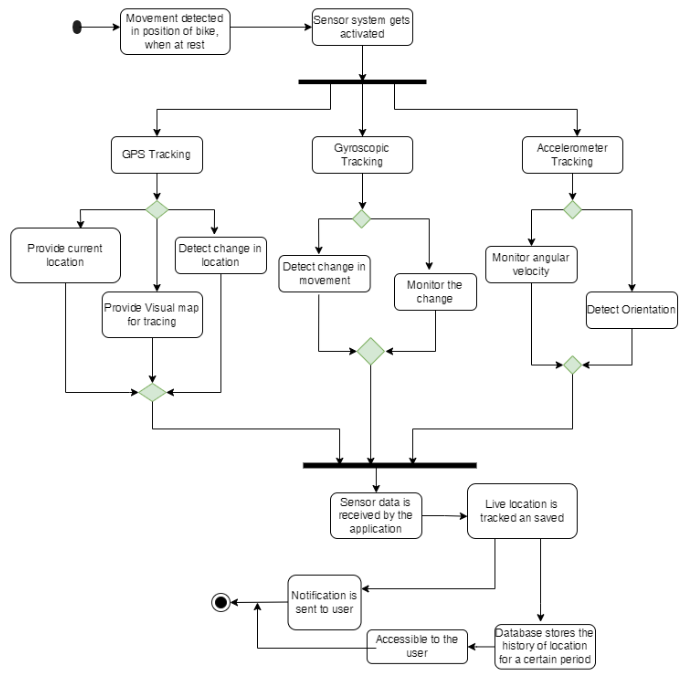

Fig. 1. Activity Diagram

III. CONTEXT VIEW MODEL

Context Model explains how the system being modelled is arranged in an environment with other systems. In other words, it depicts the system's structure. Boxes, Connectors, and Flows are important components of the Context Model. The Context Model has the advantage of being the ideal place to start when planning and analysing requirements.

As shown in the figure, Anti-theft System is interrelated to four other systems: the GPS system, Accelerometer sensor, Gyroscope sensor and Push notification system. All of the boxes are dependent on each other.

All of the boxes in the model are identical to those listed above in fig 2, with the exception of the database, which is included for the purpose of storing location data for a set period of time in fig 3.

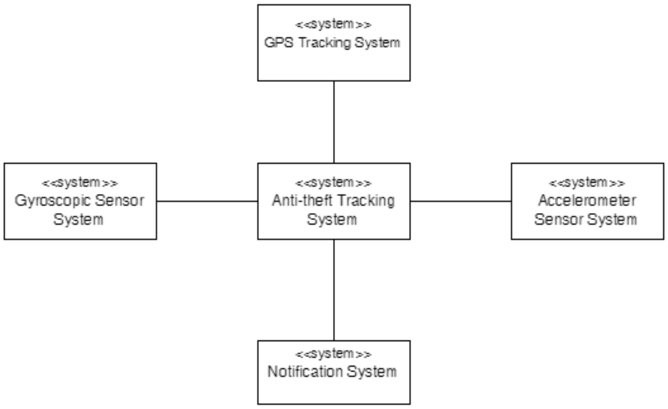

Fig. 2. Context View Model I

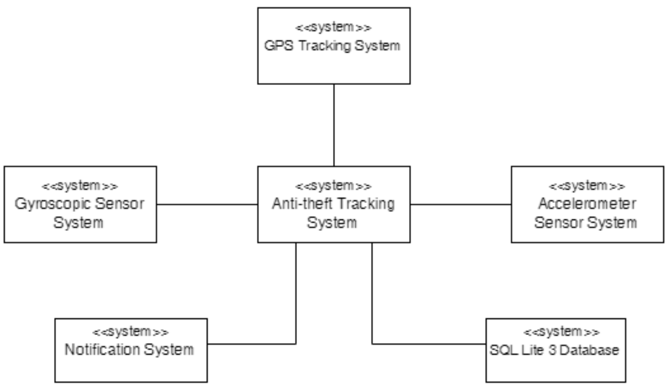

Fig. 3. Context View Model II

A database is used in this paradigm to keep the memory element. The tracking history is saved, which can be accessed by the user in the future. The previous model did not keep track of the cycle's history, therefore it was fully dependent on the current location.

IV. Event Driven Modelling

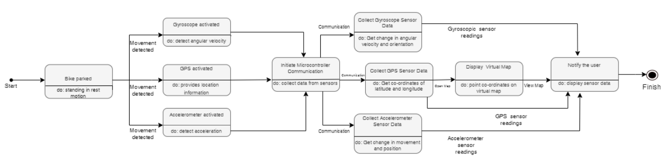

Fig. 4. State Diagram

Event driven modeling assumes that a system has a finite number of states and that events (stimuli) can trigger a transition from one state to the next. Event-based modelling with state diagrams is supported by the UML. The primary purpose of state diagrams is to represent states and transitions. A solid circle represents the initial state, while a dashed circle represents the ending state, in most diagrams. A state diagram depicts a behaviour model including states, transitions, and actions. State diagrams are used to model a class's complex behaviour in response to time and external inputs.

The state of the system, as well as a description of how the state change occurred, are described below:

A) Initiate microcontroller communication: The microcontroller's communication interface behaves as a relay between the system and the user, providing real-time sensor data to the user.

B) Collect GPS sensor data: The GPS sensor acts as a navigator, providing the bike's location using the Global Positioning System. It tells you the bike's latitude and longitude.

C) Collect data from the accelerometer sensor: The accelerometer is used to track the position and movement of the bike. It displays the object's acceleration along the x, y, and z axes.

D) Collect data from the gyroscopic sensor: The object's angular velocity and orientation are determined using the gyroscope. The projection of angular velocities along the x, y, and z axes is provided by the gyroscope's output.

E) Display Virtual Map: The GPS locations are marked on the virtual map, and the user can view and track the bike with this mapping system.

F) Notify the user: The sensor output's real-time data can be communicated with the user, resulting in an immediate notification.

V. Object Oriented Pattern

It is a pattern which is commonly used for designing the software. It is based upon the principle that an object commonly called subject maintains the list of observers and whenever there is a change in a state of subject, a notification is sent to the observer by calling the update method. For receiving the notification, user needs to undergo the registration process. In the class diagram shown below, when there is a change in the location of bike, the accelerometer, gyroscope, and GPS Sensor will get activated and the changed value will be sent to the subject and through the function notify observer, user will receive the notification of the changed coordinates and user can view the coordinates on Google map.

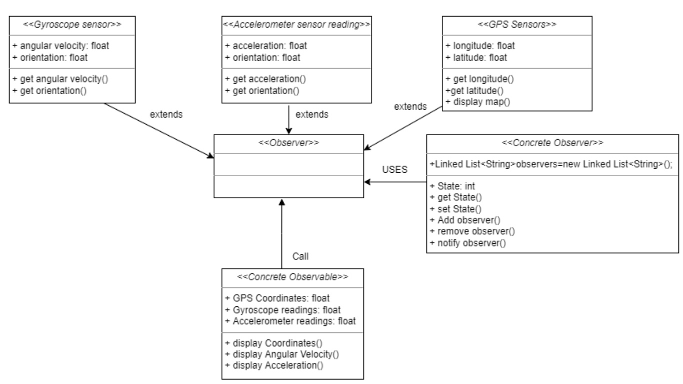

Fig. 5. Class Diagram

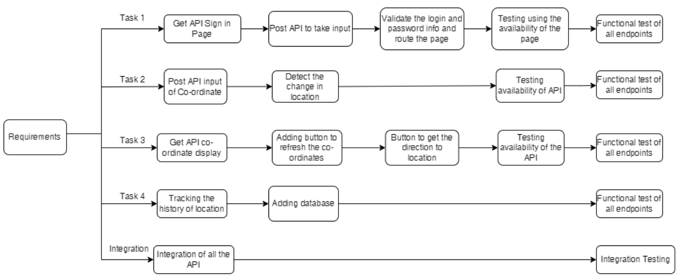

Fig. 6. Incremental Programming

VI. Incremental Programming

Incremental programming is to complete the parts of program with its implementation and then integrate with each other to make the whole system work.

Here in this project we started with defining the milestones of 3 basic tasks and integration of all the task to form a working system.

The milestone we set is based on the basic requirement is fulfilled and the API is working as the intention for more than 90

The whole process has to go through two phase

Implementation

Testing

Implementation

First the task milestone are selected based upon the complexity of the task.

When we reach the milestone we start with testing of the system.

Testing

It is done in two part

Availability of the API

Functional test

When we complete all the three parts we integrate all of them to make a working system, which is followed by integration testing.

VII. Development

A. Technology and Framework

We have used different technology and framework for making the whole system.

1) Flask API(Framework)
2) HTML(Web Dev)
3) CSS(Web Dev)
4) Bootstrap(Web Dev)
5) Python(Backend)
6) Postman (Testing)

B. Decisions

We have used the flask instead of fast is because of the documentation and the community support. As fast api is evolving we don't have needed support for the same.

HTML , Bootstrap and CSS for designing the web pages

We have used python as a backed scripting language.

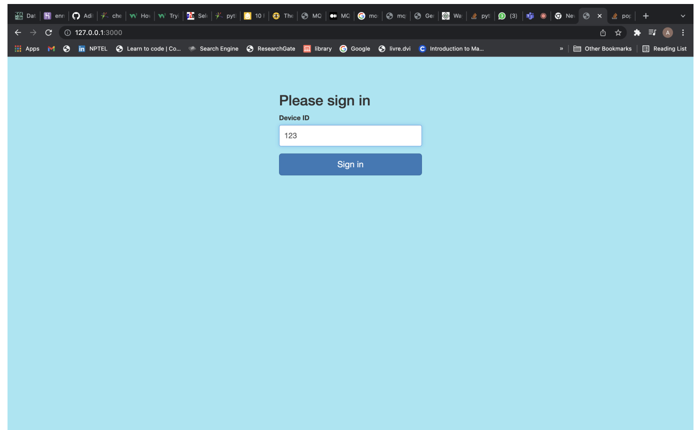

Fig. 7. Sign In Page

Postman is used to test the API working.

This is the login page so that you can only access the data of the given user.(Fig 7)

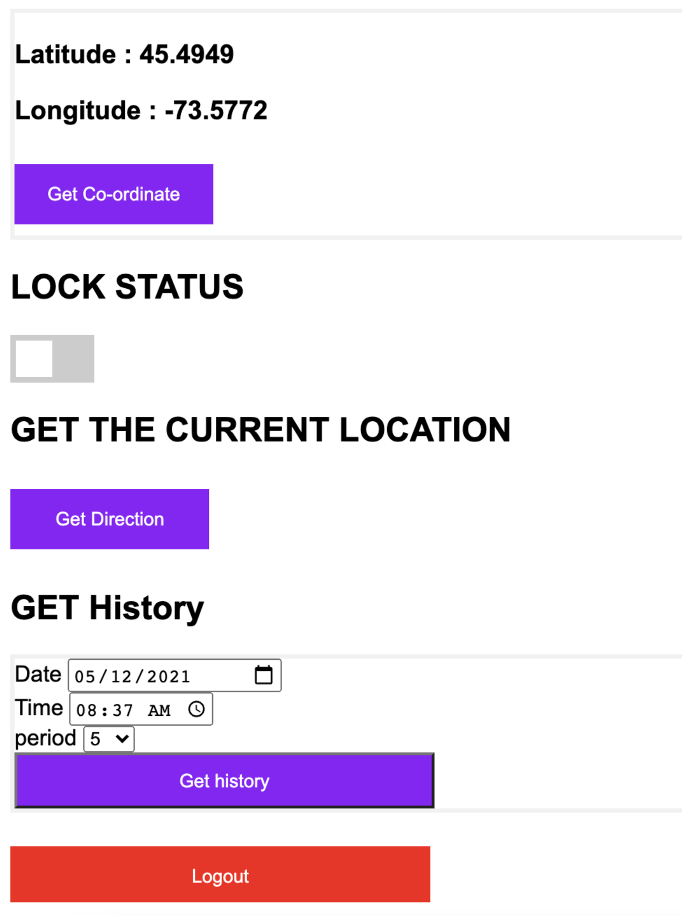

Fig. 8. Dashboard

Fig 8 It shows the dashboard where you can see set the lock condition and access the latest data of latitude and longitude of the given devise and have the option to view history by

History

Device ID	DATE and TIME	GPS LATITUDE	GPS LONGITUDE	ACCELEROMETER X	ACCELEROMETER Y	ACCELEROMETER Z	GYROSCOPE X	GYROSCOPE Y	GYROSCOPE Z
1234	202112051441	105	90	2332	234	32	432	6634	971
1234	202112051445	105	90	2332	234	32	432	6634	971
1234	202112051446	105	90	2332	234	32	432	6634	971
1234	202112051452	100	90	2332	234	32	432	6634	971

Fig. 9. History of device 1234

setting the timer and day and date Fig 9 and Fig 10 shows the

History

Device ID	DATE and TIME	GPS LATITUDE	GPS LONGITUDE	ACCELEROMETER X	ACCELEROMETER Y	ACCELEROMETER Z	GYROSCOPE X	GYROSCOPE Y	GYROSCOPE Z
123	202112051440	105555	90111	2332	234	32	432	6634	971
123	202112051440	105	90	2332	234	32	432	6634	971
123	202112051456	100	90	2332	234	328	432	6634	971
123	202112051456	108	90	2332	234	328	432	6634	971
123	202112051457	108	90	2332	234	328	432	6634	970
123	202112051457	108	90	2332	23	328	432	6634	970

Fig. 10. History of device 123

history of the device which is stored and shows the pings of the device movements.

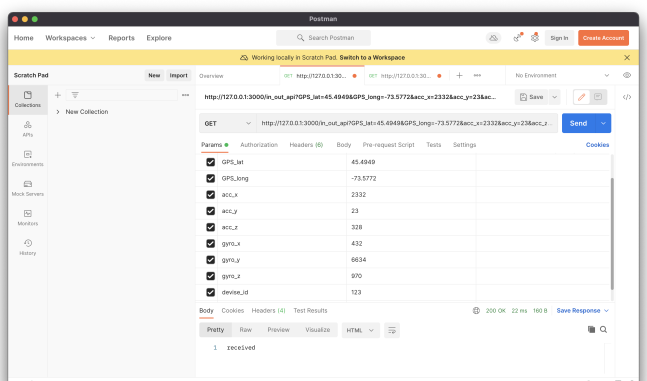

Fig. 11. Postman for testing the API

Fig 11 Here we are using the postman to inject the API with the information and see the input of the devices logged into the database. For the purpose we g=have used SQLite database as it is very lite weight and also can be stored on the filesystem.

Fig 12 and Fig 13 displays the alert when the position of the by-cycle changes with respect to the last.

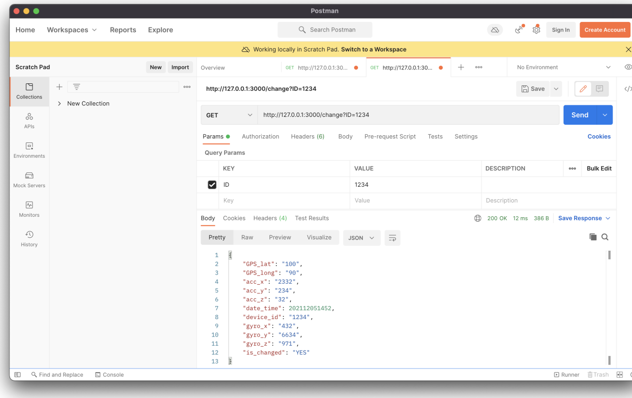

Fig. 12. Postman for receiving the alert

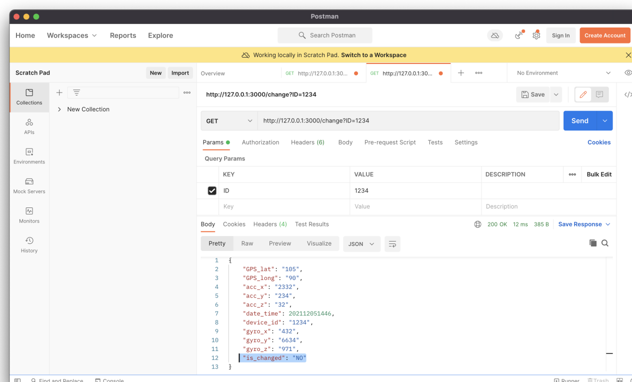

Fig. 13. Postman for receiving the alert

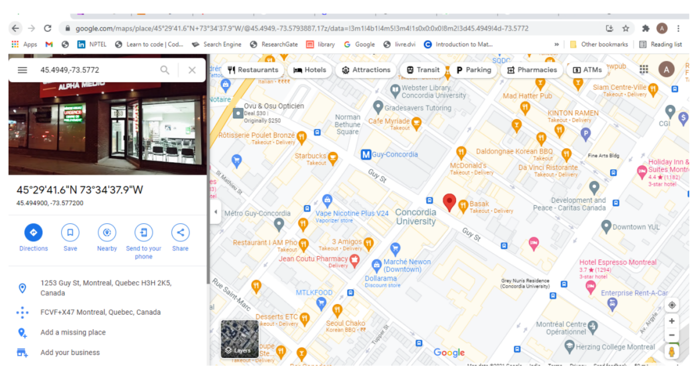

Fig. 14. Map

VIII. Acknowledgement

We would like to thank Prof. Yan Liu from the Department of Electrical and Computer Engineering at Concordia University for her valuable suggestions, to complete this assignment.

References

[1] https://www.sqlite.org/datatype3.html
[2] https://flask.palletsprojects.com/en/2.0.x/
[3] https://getbootstrap.com/
[4] https://www.postman.com/
[5] Assignment 1 Report