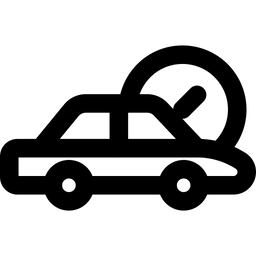
****

**Information Technology Institute**

**Java Project**

**Car Meter Project**

**Submitted to:**

**Eng. Eman Hesham**

Submitted by: Group 5

**Abdullah Hanafy**

**Ahmed Sherif**

**Asmaa Ebrahim**

**Asmaa Saied**

**Mostafa Ahmed**

Contents

[**1. Overview:** 3](#_Toc92658990)

[**2. GUI:** 4](#_Toc92658991)

[**Speedometer:** 4](#_Toc92658992)

[**Map:** 4](#_Toc92658993)

[**Initial scene:** 4](#_Toc92658994)

[**“View trips” Scene:** 4](#_Toc92658995)

[**“Save trip” Scene:** 4](#_Toc92658996)

[**Error messages dialogs:** 4](#_Toc92658997)

[**3. Serial Communication:** 5](#_Toc92658998)

[**GPS using Mobile phone:** 5](#_Toc92658999)

[**GPS Module:** 5](#_Toc92659000)

[**4. The Audio Package:** 6](#_Toc92659001)

# **1. Overview:**

The project is about a desktop application that allows the user to open a map and start tracking his location on this map.

It also shows the user the speed he is moving with by changing the value of the speed in a speedometer that appears in the GUI.

The user also has the ability to save his trip after finishing it by pressing the button “End trip” the application asks the user if he wants to save the trip he has just finished or not.

The user can find the saved trips by pressing the button “View Saved Trips” that shows the user a list of hyperlinks with the names of all trips he saved.

When the user press on one of the hyperlinks it shows the user a map that has two marks one of the start point and the other of the end one.

There also a feature that when the user exceeds a certain speed the application turns on an alarm to get the user attention which might prevent an accident or car crash.

The user also has the ability to delete saved trips by pressing the button “Delete all trips”.

If the GPS module has not been turn on yet and the user pressed “Start” button a message is sent to the user to tell him that the GPS has not started yet and he has to for seconds.

# **2. GUI:**

## **Speedometer:**

We Used “Medusa gauges and clocks” library to get the shape of the speedometer and to set its value in a thread while the program running.

## **Map:**

\*\*\*This part will be edited by Abdullah\*\*\*

## **Initial scene:**

The initial scene contains the map, the speedometer, “Start” button and “View Trips” button.

## **“View trips” Scene:**

Contains a list of hyperlinks that does not exceed 5 each hyperlink is a name of a saved trip and by pressing on one of them it shows a map with two marks one of the starting point and the other one is the end point of the trip.

## **“Save trip” Scene:**

Contains a text field that the user should enter the name of the trip in, “Save” button the user presses on it if he wants to save the trip and “Cancel” button if he does not want to save the trip.

## **Error messages dialogs:**

\*\*\*This part will be edited by Abdullah\*\*\*

# **3. Serial Communication:**

In this part, the whole focus is on how to get the GPS data so we can show and use it in our application, and that will be through establishing a serial communication between the Mobile phone/GPS module and our application.

## **GPS using Mobile phone:**

* Install Share GPS output on your mobile phone.
* Establishing a serial communication could be through USB or Bluetooth, we tried USB but we found that Bluetooth is more easy and accurate.
* Using Bluetooth, we can make the mobile and the laptop paring, and that will create serial ports for both. SerialCommunication.java will sense the mobile serial port and the serial thread will be started.
* Now, we can receive data from GPS.

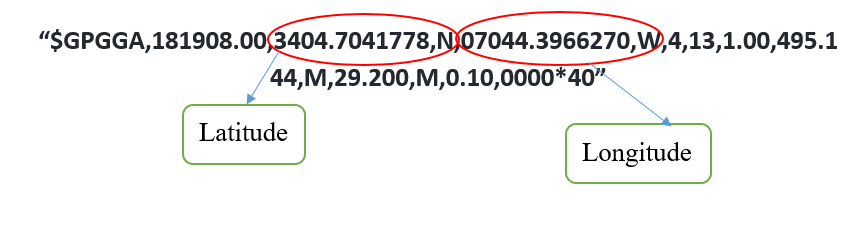
## **GPS Module:**

\*\*\*This part will be edited by Abdullah\*\*\*

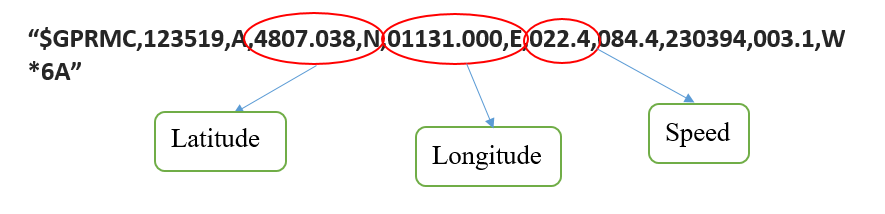
The data we are interested in are: the speed, the latitude and the longitude. The GPS data sentence are in NMEA sentence structure.

There are three sentence we got from the GPS, we are interested in two of them:

***For Example, the First one:***



***For Example, the Second one:***



# **4. The Audio Package:**

This is used to implement the audio alarm if the speed exceeded the recommended speed above the safety limit

First using the file class: The FileSystem object representing the platform's local file system.

Second using media class: Media class represents a media resource. It is instantiated from the string form of a source URI. Information about the media such as duration, metadata, tracks, and video resolution may be obtained from a Media instance.

Then using MediaPlayer class The MediaPlayer class provides the controls for playing media. It is used in combination with the {@link Media} and {@link MediaView} classes to display and control media playback.

And for testing purpose A simple GUI was used to test the playing and pausing and stopping the audio from playing, so a play pause Button was used with the name “play audio”.

Then implementing a Flag to be used to check the status of the audio.

Finally using the default constructor for initialization of each object reference

After this simple test verified its functionality a merge is implemented to CarMeter.java