Carmeter

**Abstract.** Carmeter project is about interfacing a GPS with a GUI [gauges or indicators for showing speed, longitude, latitude, etc..], so we show the data sent from GPS are to be parsed by Java application and showing them in the dedicated field on a user-friendly GUI Java application.

**Flowchart:**

Diagram

Description automatically generated

**Project Features:**

Java desktop application – Frame based – shows the CARMETER GUI with:

* Labels to display the location using longitude and latitude in addition to displaying the speed [start, stop, latitude, longitude, sound off, switch, light mode and dark mode buttons].
* The displayed information comes from parsing the GPS data.
* An audio is played as an alarm when the speed exceeds a certain limit and stops automatically when the speed is decreased and so on.
* The application can run if it’s not connected to internet by taking the choice of the user.
* You can close the port and re-open it again.
* We added some features to our application like:
  + You can switch between viewing the speedometer and the map simultaneously
  + You will find the light mode as the default mode but also you can use the dark mode.
  + You can resize the GUI application with resizing everything it includes like the map, speedometer, buttons, etc...

**Implementation:**

1. **Installation of Bluetooth GPS Output**

* Download and install “Bluetooth GPS Output” application from google play
* Pair the android phone with the laptop
* Then open the application and make sure it read the location.

1. **Installation of libraries:**

* JDK 1.8.201 (To make sure RXTX and GMapsFX work)
* Medusa-8.0.jar (This library to display the speedometer)
* GMapsFX-8.0.0.jar (To display Google map)
* RXTXcomm.jar (Used for the serial communication)
* Marineapi-0.10.0.jar (To support NMEA sentence)
* serialRXTX.dll (make sure to put this file in your jdk extension exactly in <C:\Program Files\Java\jdk1.8.0\_201\jre\bin>)

**Graphic User Interface:**

A screenshot of a car dashboard

Description automatically generated with medium confidence

* Add Medusa library to libraries (Medusa.jar).
* Make CSS Package include css.css file (CSS file include the colors codes for background and speedometer).
* Make Buttons start (to start reading speed and position using the speedometer and buttons for displaying the latitude and longitude), stop and sound off (turn off the alarm speed over 20K/H), switch, light mode and dark mode buttons.
* Set Action for each Button.
* Set design and area for each Button.
* Mak gauge to display the speed.
* Make GridPane to add the objects (like Buttons, gauge, map).
* Make 2 Text to show the values for longitude and latitude.
* Set the position for Buttons, gauge, and Text
* Insert audio file to be an alarm.
* Make HBox and add Map and the gridPane on it.
* Make StackPane to add background to the GUI.
* Put the stackPane on scene.
* Then put the scene on Stage.
* Make a GPS icon for stage.