CAPSTONE REPORT

Driver Distraction Mitigation System

SE490

Abdullah Janjua	140096
Omar Assem	150039
Khalil Mayyasi	150270
Abdulmalik AlRajhi	150320

Alfaisal University

Advisor
Dr.Muhammad Anan

Project Management Plan

Purpose, Scope and Objectives

Introduction:

Drivers these days get distracted while driving their cars which increases the chance of an accident. It is estimated that every year, about 421,000 people are injured in crashes that have involved a driver who got distracted in some way. Sources of distractions include: **using mobile phones, talking to people in the car, and not paying attention to the road traffic**. All distractions compromise the safety of the driver, passengers, bystanders and those in other vehicles.

Moreover, about 25% of crashes can be attributed to driver distraction to some extent according to the police reported accident data. A study based on a survey of 1367 drivers suggests that driver distraction may contribute to 14% to 33% of all serious crashes. There are many researches and studies that suggest different methodology in order to detect the driver's distractions.

Most researches and studies point out to most commonly distractions indicators such as speed, steering angle, lateral lane position and brake pressure, and eye movement based indicators, e.g., fixation duration, saccade frequency, gaze distribution and head orientation. Our system aims to eliminate distractions by alerting the driver if any of these distractions are detected.

The system consists of mobile and web applications. The mobile application will send an alert to drivers in case a distraction is detected. The driver can log-in to the website to follow up with reports and alerts history to help in improving their driving behavior behind the wheels.

Purpose:

We will build a driver distraction mitigation system, composed of mobile and web applications, to reduce the chance of drivers getting distracted while driving their cars on the road. In addition, the system will help the parents to monitor their child's behavior behind the wheels.

Scope:

The mobile application will be built to read data from different sensors and send it to the web application through built APIs. The web application will process the data and figure out if any distraction has happened, then notifies the mobile application to alert the user. The web application will allow parental control for the drivers, it will allow parents to check on their children by tracking their location through GPS using google maps APIs and the parents will be able to send some alert messages that will be delivered as voice messages to the driver.

The mobile application will automatically check if the phone is connected to the internet as well as to the GPS, otherwise alert the driver to connect to the internet and/or GPS before starting the trip. The mobile application will be built for android users only, while the web application can be used by any user registered on the system.

The system will use a database that will hold data of the trips of the users, and data of the parental accounts. The system will also show last trip’s summary for the driver in the mobile application, and previous trips summary will be displayed on the web application.

Objectives:

The end-system will help drivers get less distracted while on the road because it will be sending them notifications whenever they get distracted. This way drivers will stay focused on the road and become more safe since the chance of them getting into accidents will drop significantly.

Also, the system will allow parental users to monitor the drivers while they are on their way driving their cars anywhere and at any time. Hence, parental users will be more assured about their children or relatives that are driving the car and they can warn them about their distracted-driving behavior.

In addition, It should enable drivers to fix and improve their driving style through looking back at their previous trips’ summaries and figuring out how many times they were distracted and for what reasons. Doing so, will help them drive better over time and therefore, reduce their chances of doing accidents even further.

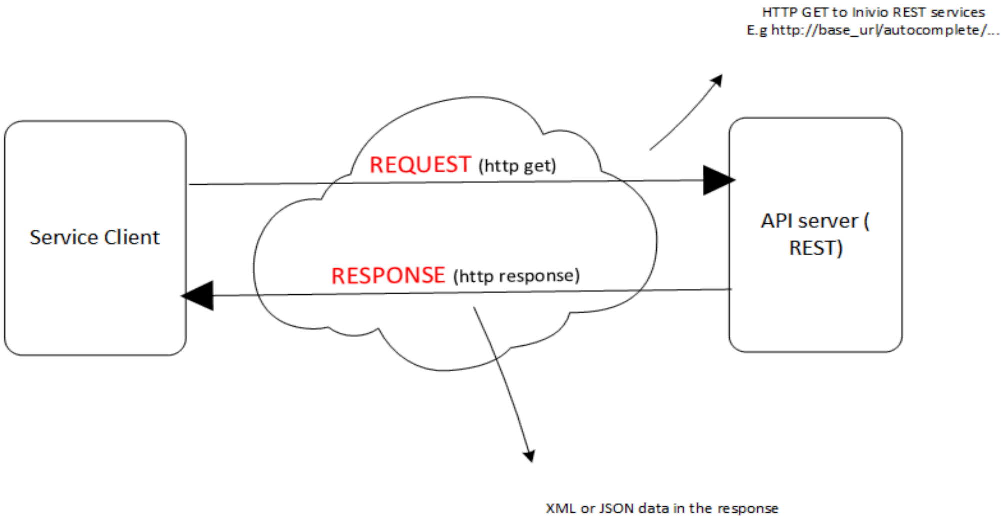
Background

Cell phone usage while driving is a major distraction. Due to the number of accidents that are related to cell phone use while driving, our project implements a mobile application system that mitigates drivers' distractions while they are driving their cars. To achieve this goal, there was a number of challenging functionalities and features that we researched first to know how they can be designed and eventually, implemented in our project:

API (Application Programming Interface):

The term API refers to Application Programming Interface. It can be used to describe the characteristics of a library and how to use it. Each API has a documentation that talks more about its functions, how you call them, and which arguments are required to be passed. However, a problem started to arise when everyone started building their own APIs without a standard way or format, and to solve this issue completely RESTful APIs were invented. The term REST stands for Representational State Transfer and it describes a standard way for creating HTTP APIs, using four actions (view, create, edit and delete) that map exactly to four HTTP verbs already implemented (GET, POST, PUT, DELETE).

An Application Programming Interface (API) acts as a door into a software program to enable other programs to interact with it without needing the developer to share its entire code, and this is of course to keep that application secure. In our project, we will be developing RESTful APIs to allow communication between the mobile and web applications, as well as, sharing of data. Examples of data shared include: car's speed, distance covered since trip's start, and driver's current location to enable parental users to monitor him.

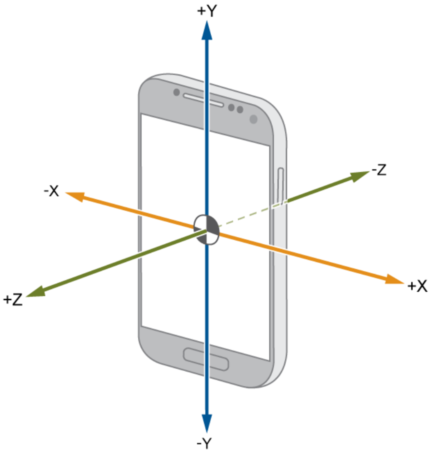

HTTP GET to Inivio REST services
E.g http://base_url/autocomplete/...
Service Client
REQUEST (http get)
RESPONSE (http response)
API server (REST)
XML or JSON data in the response

How an API works is as follows: On the server side, APIs are created and all their functionalities that need to be accessed by the client are implemented. Following that, the client will send HTTP requests to the API server requesting for some kind of response or data and the request could contain parameters that are needed by the API methods to function properly. Once the API server is done with processing the request, it sends an HTTP response to the client with the output data represented in either XML or JSON.

In our project, we will be using APIs to exchange information between the mobile and web applications, without having to fully expose the code of any of them. For instance, the mobile application will send the current speed of the driver to the web application through an API and then the web server will process it and figure out if it's over the speed limit, in which case it'll send back a response to the mobile application saying that the driver's speed has exceeded the limit. Now, it's the mobile's job to process the response and send an alert to the driver asking him to slow down.

Accelerometer:

One of the sensors that will be used to collect data is the linear accelerometer. We will be using the linear accelerometer which can use the (x, y, and z) axes and return the acceleration in those axes excluding the gravity acceleration which helps to get the speed of the car, and the acceleration will be measured in m/s^2. The orientation of the axes relative to the device is that the x and y axes are parallel to the screen of the device while the z axis is perpendicular to the screen.



We will use the linear accelerometer for multiple purposes, first to check if the acceleration has reached higher than the set threshold which we set. Secondly, we will use the accelerometer to check if there is a fluctuation in the acceleration in a short time which shows that the driver's attitude in driving is not stable like drifting and this happens when the driver speeds up and presses the brake quickly. Thirdly, we also will use it to check if there is a sudden drop in the acceleration which means that the driver has pressed the brake due to the probability of an accident about to happen.

On Screen Touch:

In order to detect if the driver is using his/her smartphone while driving, we keep track of the driver’s smartphone screen. Because the user can not navigate through the interface without touching the screen, the system will detect any on-screen touches and display a notification to alert the driver to focus on the driving. We will be using a built in event which is **onTouchEvent** to detect any on-screen touches. Moreover, we will hide the system action bar when the application is running to eliminate on screen distractions.

Text-to-Speech:

Text-To-Speech will translate a string of text to an audio file that can be played on the phone speakers. The Text-To-Speech is a per-built API on most of the android OS provided by Google. The system will get the input text from the web application and send it to the smartphone as a string of text to be translated on the mobile application so the driver can listen to the text rather than read it which will decrease the destruction by keeping the driver's eyes on the road.

UI (User-Interface):

Nowadays, smartphones have a major role in anyone's life to the point where many drivers use them while driving which may result in major accidents. While many people are able to multitask by using their phones and driving, still some people take a huge time to focus on the device to perform a task while driving.

One way to decrease the need to get distracted for a long time on the phone is to decrease the amount of visual designs on the application because exaggeration of visuals on the screen will confuse the driver to keep looking at the screen to find what he needs. Another way is to make all the visuals on the screen as clear as possible by enlarging the size of the components (Buttons, or messages) to make it easier for the driver to interact with our application in minimal time so he can focus more on driving.

Assumptions and Constraints

- GPS service is accurate in determining the driver’s location
- Users have basic knowledge of how to use smartphones and mobile applications
- User is legally allowed to drive
- Phone operating system is android-based
- The application is not able to control in-car systems
- The application is not able to figure out if the driver is eating/drinking/talking
- The application cannot send alerts if the device is disconnected from the internet

Project Deliverables

- **Project management plan:** A project management plan is a formal, approved document that defines how the project is executed, monitored, and controlled. The primary uses of the project plan are to document planning assumptions and decisions, facilitate communication among project stakeholders, and document approved scope, cost, and schedule baselines.

- **Software requirements specification document:** describes the requirements (functional & non-functional requirements) of the software system that will be developed. It might as well provide use cases that must be provided to interact with the user.

- **Prototype:** It’s an initial model of our system, built for demonstration purposes and as part of the development process. A prototype is built, tested, and then reworked as necessary until an acceptable version is finally achieved from which the complete system can be developed.

- **Use case diagram:** Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

- **System context model:** Describes how the system will interact with external systems, beside listing these systems in one model.

- **Mobile application:** An android application that will allow users to interact with the system. In addition to sending data from OBD II and sensors to the back-end server.

- **Web application:** The web application will process data and send the results back to the mobile application. Also it will have a map to track the user's phone location.

- **Database (DB):** Our primary storage where the web-based backend will store data retrieved from the users via mobile application. The database is a collection of information that will assist us in organizing, accessing, managing, and updating data stored.

- **Demo (or demonstration):**_A fully functional software that is not ready for production, mainly used for testing and gathering end-users feedback.

- **Test cases:** Are requirements that the developers create to be used to verify and validate the working functionalities of the system.

Schedule

Date	Task Description
Semester 1 (Fall)	
Requirements	
Week 9	Requirements document
Designing	
Week 10	System context model
Week 11	Use case diagram
Week 11	Class diagram
Week 11	Activity diagram
Week 12	ER diagram
Week 13	Prototype
Semester 2 (Spring)	
Implementation	
Week 1	Creating application database

Week 6	Developing web application back-end
Week 7	Developing web application front-end
Week 9	Developing mobile application
Testing	
Week 11	Designing test cases
Week 12	Executing test cases
Week 12	Analyzing results

Budget

Item Name	Quantity	Cost
SSD	1	450 SAR
Dell Inspiron Desktop 3668	1	2999 SAR
Poster design & printing	1	200 SAR
Internet (Mobily)	1	1155 SAR

References

- http://www.icebike.org/texting-and-driving/
- http://www.databasecompare.com/what-is-data-database-(db)-dbms-and-dbs.html
- DriveSafe: an App for Alerting Inattentive Drivers and Scoring Driving Behaviors. Luis M. Bergasa, Daniel Almería, Javier Almazán, J. Javier Yebes, Roberto Arroyo http://ezproxy.alfaisal.edu:2253/stamp/stamp.jsp?tp=&arnumber=7098008
- Safety and Operating Issues for Mobile Human-Machine Interfaces Dirk Bühler, Sébastien Vignier, Paul Heisterkamp, and Wolfgang Minker DaimlerChrysler - Research and Technology P.O. Box 23 60, 89013 Ulm, Germany https://goo.gl/T9GSCs

- Driver Modeling for Detection and Assessment of Distraction http://ezproxy.alfaisal.edu:2253/stamp/stamp.jsp?tp=&arnumber=7974867
- What's an API and Why Do You Need One?, BY GREG SLETER / MAY 29, 2014

  http://www.govtech.com/applications/whats-an-api-and-why-do-you-need-one.html

- Beginners guide to creating a REST API, Andrew Havens (Ruby Developer), 13 SEPTEMBER 2012

  http://www.andrewhavens.com/posts/20/beginners-guide-to-creating-a-rest-api/

- An introduction to Text-To-Speech in Android 23 September 2009 https://android-developers.googleblog.com/2009/09/introduction-to-text-to-speech-in.html
- Autonomous Detection of Distracted Driving by Cell Phone, 2011 14th International IEEE Conference on Intelligent Transportation Systems Washington, DC, USA. October 5-7, 2011
- Detecting Driver Distraction Using Smartphones, IEEE 29th International Conference on Advanced Information Networking and Applications

Software Requirements Specification

Purpose

The purpose of the system is to create a mobile application that monitors a car's environment and driver’s behavior to make sure that he is not distracted and in case of distractions, the mobile application notifies the driver through sending alerts. The system also offers a web-portal for the driver to check his trip’s summary and for other users who need to monitor the driver (parental users) while he’s driving his car. The reason why this system was built is to reduce the number of car accidents that happen every year around the world because of distracted driving.

Scope

Some functionalities will be implemented in our system to reduce distractions and accidents from happening. These functionalities are as follows:

- Reading data from the car
- Analyzing this data and sending appropriate alerts back to the user
- Using mobile sensors to detect if the user is using the phone while driving
- Tracking the driver’s location while on the way

Other functionalities weren’t implemented in the system since some distractions are hard to control through an application. These functionalities are as follows:

- Pressing the brake pedal if the driver goes over a certain speed
- Detecting if the driver is talking to someone in the car
- Driver not focusing on the road

System Overview

After the customer installs the mobile application, he/she will have to login, or sign-in in case he/she does not already have an account. When the driver starts the trip the mobile application will collect data from mobile sensors and send it to the server (back-end) for further processing. If the server detects any distractions by the driver it will send alerts to him through the mobile application interface.

The user has an option to login to the system website to review his/her previous trips’ data. The system collects and analyzes the incoming data from the mobile application in order to provide an overview of the driver behavior behind the wheels.

Moreover, the system provides parents the ability to track the driver and send him voice messages, too.

References

- Whatis.techtarget.com - Smartphone Sensor October 2014, by Margaret Rouse http://whatis.techtarget.com/definition/smartphone-sensor
- Belitsoft.com - Software Requirements Specification Document https://belitsoft.com/php-development-services/software-requirements-specification-document-example-international-standard
- pcmag.com - Encyclopedia https://www.pcmag.com/encyclopedia/term/60249/web-interface
- Searchmicroservices.techtarget.com - user interface, by Margaret Rouse http://searchmicroservices.techtarget.com/definition/user-interface-UI

Definitions

- **Web-portal:** The interaction between the user and application running on the Web server. The user interface is the Web browser and the Web page it downloaded and rendered.

- **PaaS:** A complete development and deployment environment on the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.

- **Mobile sensors:** Any type of sensing devices available or installed on the mobile phone to gather specific types of data.

- **Server:** A computer program or a device that provides functionality for other programs or devices (clients).

- **Back-end:** A program that can’t be directly accessed by the user, and this program performs specific functions on behalf of the main program.

- **Parental control:** Features which may be included in mobile devices and softwares to allow parents to restrict the access of content to their children.

- **UI:** User interface is everything designed into an information device with which a user may interact. It includes display screens and the way users interact with the application.

- **GPS:** The Global Positioning System is a satellite-based navigation system made up of at least 24 satellites.

- **Database:** Our primary storage where the web-based backend will store data retrieved from the users via mobile application. The database is a collection of information that will assist us in organizing, accessing, managing, and updating data stored.

Functional Requirements

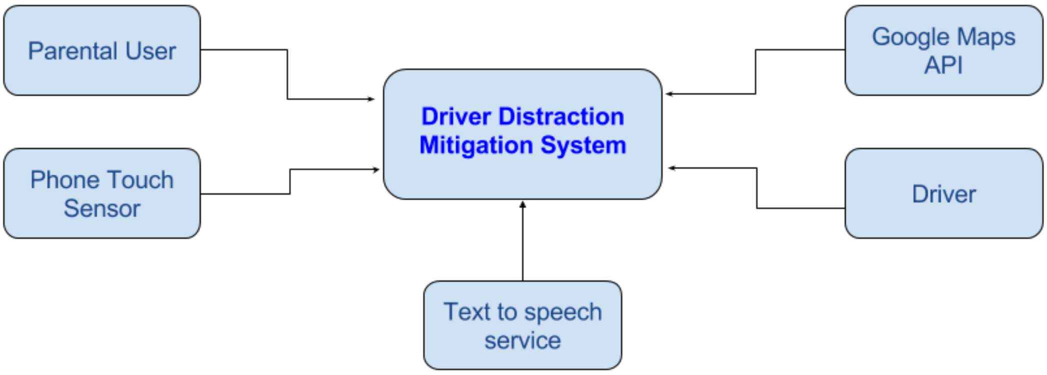
The following are the functional requirements of our system:

- The system allows the user to create & manage an account
- The system allows the user to deactivate his account
- The system allows the user to add/remove parental accounts
- The system allows the driver to start the trip using mobile application
- The system allows the user to login into the application
- The system allows the user sign-out from the application
- The system takes readings from the car via internal mobile sensors
- The system alerts the driver if the car’s speed exceeds a certain threshold
- The system forwards data from the mobile application to the web application for processing
- The system alerts the driver if it detects that the driver is using the phone
- The system stores the data to the database for future retrieval and usage
- The system displays data from mobile sensors to the user on both web & mobile interfaces
- The system will process data received through APIs, on the web-based backend
- The system will track the driver’s location and show it on the web interface for parental accounts
- The system will provide a voice message feature on the website
- The system shows a summary of the trip details to the driver after trip completion

Non-Functional Requirements

- **High performance:** The system should process and display data within less than 5 seconds to the users.
- **Robustness**: If the system loses the connection to the web server, the application shall stay up and running and not crash. Rather, it will reconnect and alert the user.
- **Usability:** The system should be easy to use and learn, both mobile and web interfaces need to be simple and user-friendly.
- **Availability:** The system is available 24/7 and can be accessed from anywhere as long as there's a stable internet connection available.
- **Space:** The system will send data to the server and save it on the database, therefore it will save memory space (ROM) on the user's device.
- **Security:** System provides username and password for each user to login into the application. The password should be at least 8 characters containing letters & numbers.
- **Compatibility:** The application will be built for only android-based devices.
- **Reliability:** The system can handle unexpected inputs from the user. In addition to ability to recover from system failures.

System Context Diagram



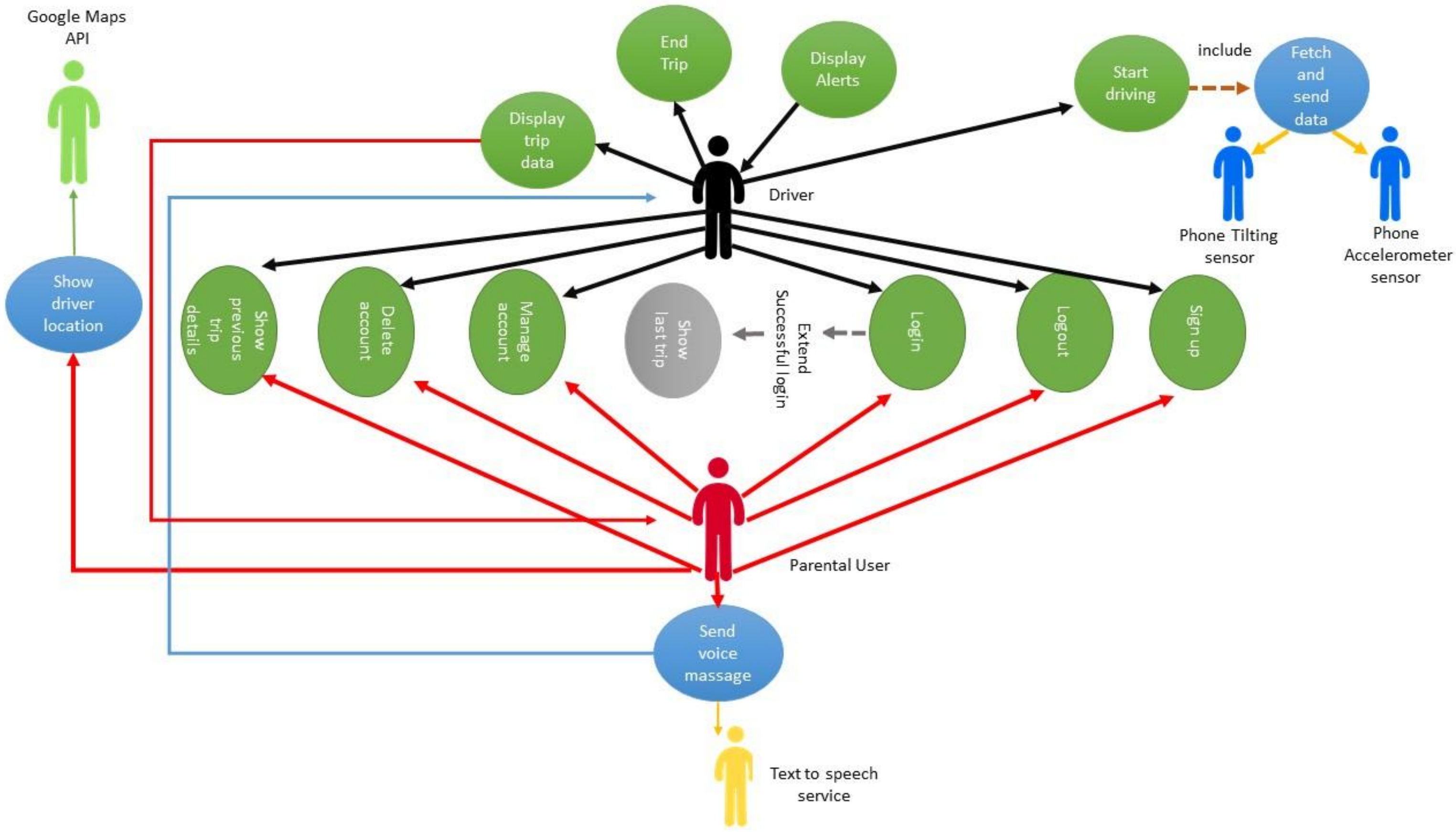
Use Case Description

- **Sign Up:** allowing both the driver and the parental user to create accounts and register in the system, and by doing so they will be allowed to access the mobile application and the web application.
- **Login:** entering username and password to access the system either through the mobile interface or the web-based interface.
- **Logout:** the user leaves his account and loses the ability to use the system functionality.
- **Manage Account:** giving the user the ability to edit some of his accounts details e.g. personal information or to add parental accounts to supervise him.
- **Delete Account:** the user will be able to delete his account from the system. He will not be able to access the system anymore until he signs up again.
- **Start Driving:** making the mobile application in a mode where the system will start checking for distractions from the driver while on the road.
- **End Trip:** exiting the monitoring mode where the mobile application will not check for distraction from the user anymore and will be waiting for the next trip.
- **Show Last Trip Data:** Getting the driver’s last trip data and displaying them when the user logs in and accesses the mobile application.
- **Send Voice Messages:** parental users will have the ability to send messages to the driver they are monitoring to alert him in case of danger or could be to inform him about something important. The parental users will send the message in the form of text but on arrival to the mobile application it will be converted to voice to make sure he’ll not get distracted while driving the car. To convert text to voice, the system will communicate with a text-to-speech API that will do that work and return the voice message ready for play.
- **Show Driver Location:** parental users will also have the privilege of tracking the driver's location while he is on the way. They will have a map showing the route taken by the driver and where in the city he is driving the car. To achieve this feature, the system will communicate with Google Maps API to show a map of the driver’s position and fetch his location from the mobile application using the location (GPS) service in the phone
- **Display Trip Data:** while the driver is driving his car and using the application, the main page on the phone will be showing details about his current trip, in case he needs to check his own driving status, e.g. average speed, distance covered, time elapsed, etc. Parental users will also view the data about the driver’s current trip and this allows them to monitor his driving behavior to make sure that he isn’t putting himself at risk.
- **Show Previous Trips Details:** using the web-application, both drivers and parental users are able to view previous trips details and revisit them at any time. These details are stored in a database and retrieved to be displayed to the users

upon their requests. Since storage is limited, not all trips will be saved, only the recent ones.

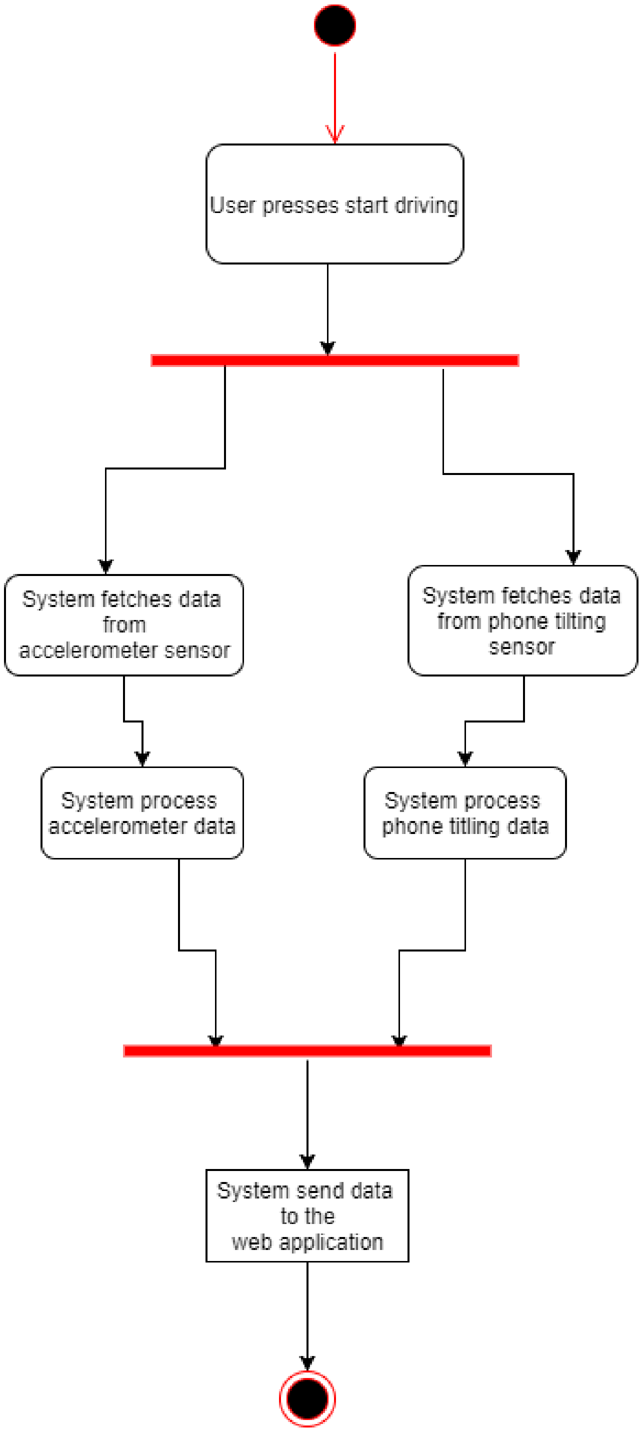
- **Fetch and Send Data:** When the user starts driving the system will start fetching data from the phone tilting sensor and the accelerometer sensor. Then, the system will do some processing with the data to detect if any distraction has occurred and then send them to the web application.
- **Display Alerts:** the mobile application while on the driving mode, will be monitoring the driver and looking for distractions. Once found, it will send the driver alerts instantly to notify him of what wrong behavior he’s doing so he can drive more safely. The alerts are visual with sound effects to ensure that he notices them easily.

Use Case Diagram

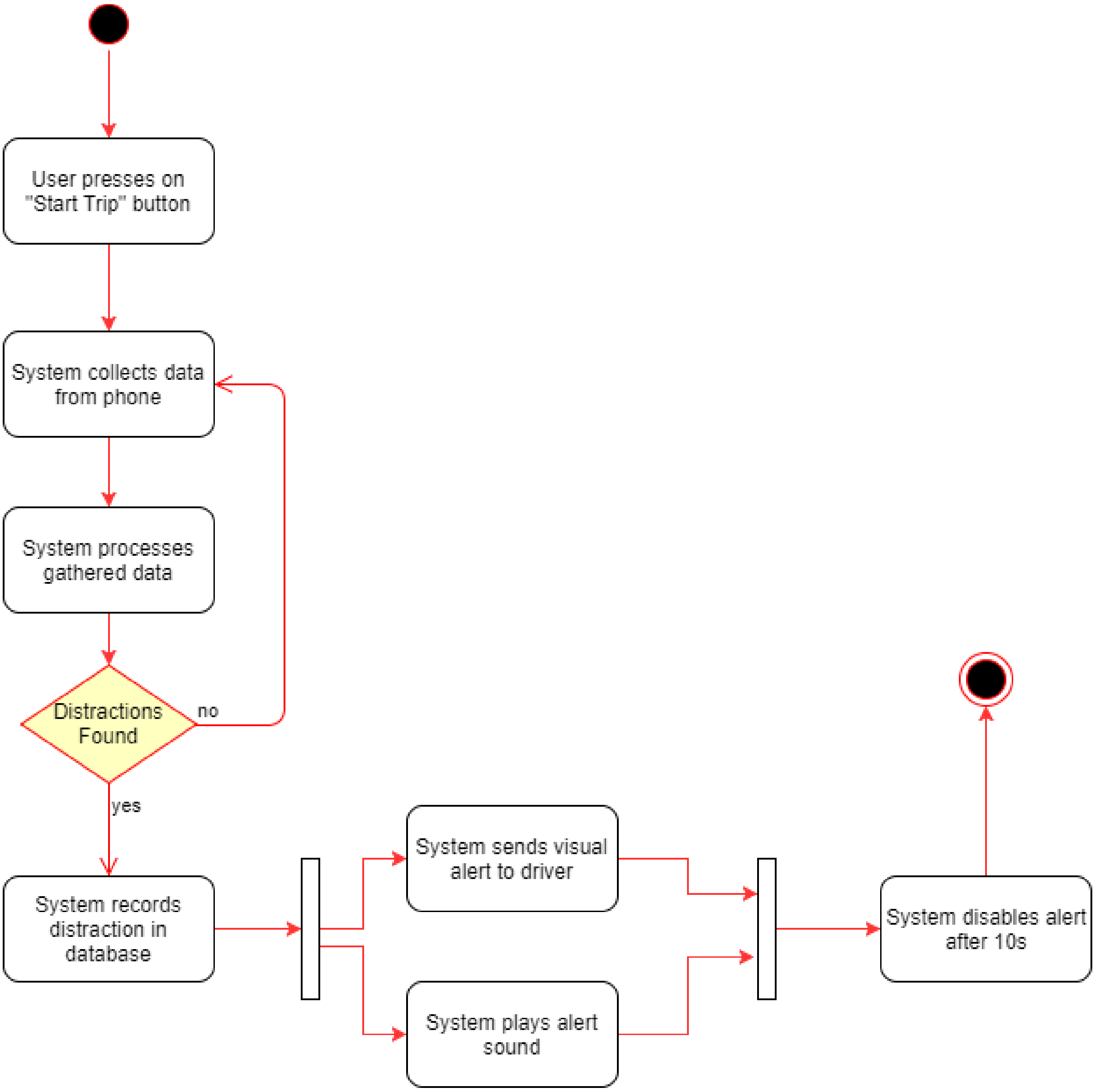


Activity Diagrams

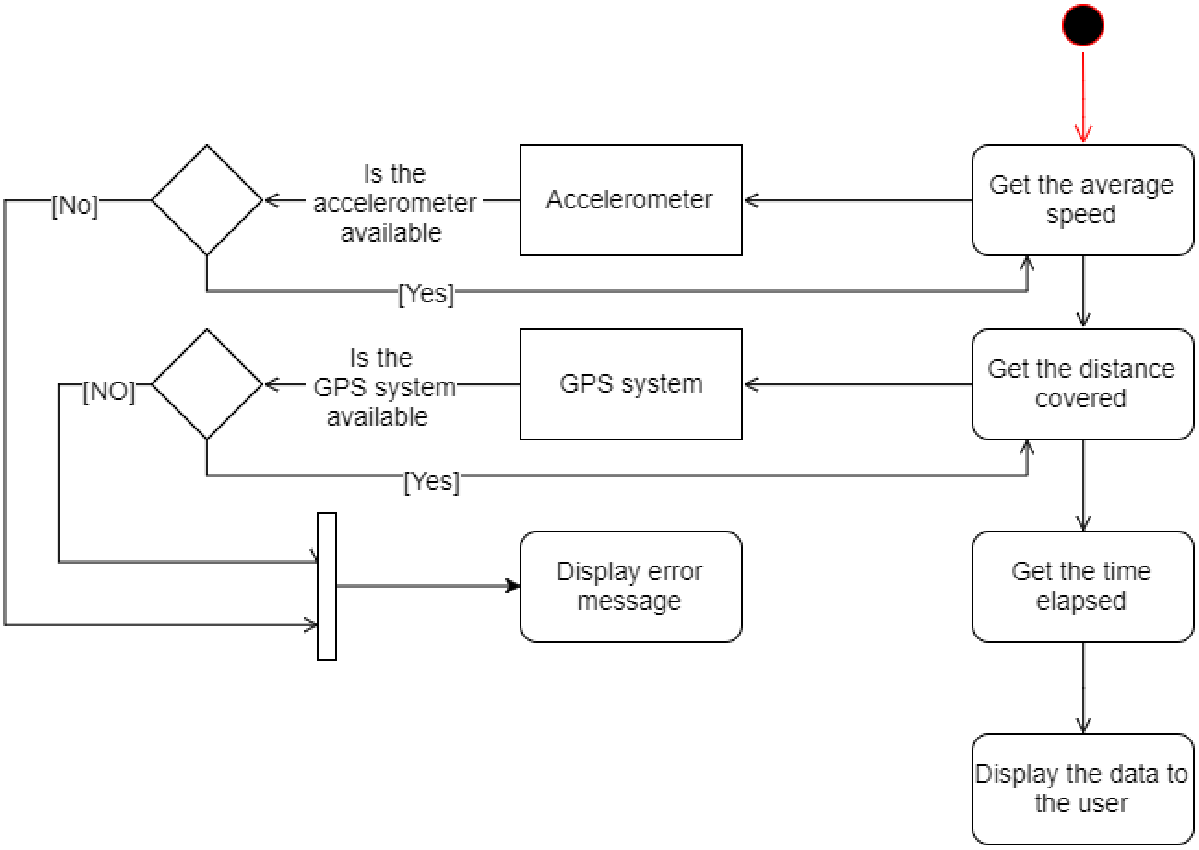
1) Fetch and Send Data


User presses start driving
System fetches data from accelerometer sensor
System fetches data from phone tilting sensor
System process accelerometer data
System process phone titling data
System send data to the web application

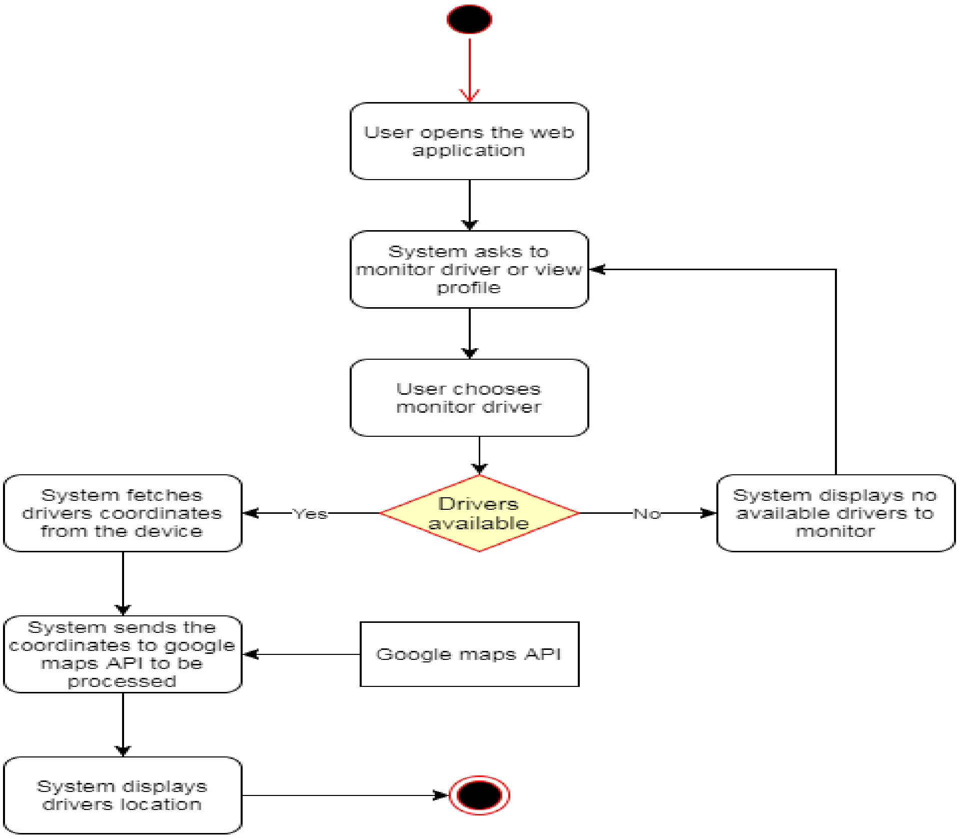
2) Display Alerts


User presses on "Start Trip" button
System collects data from phone
System processes gathered data
Distractions Found
no
yes
System records distraction in database
System sends visual alert to driver
System plays alert sound
System disables alert after 10s

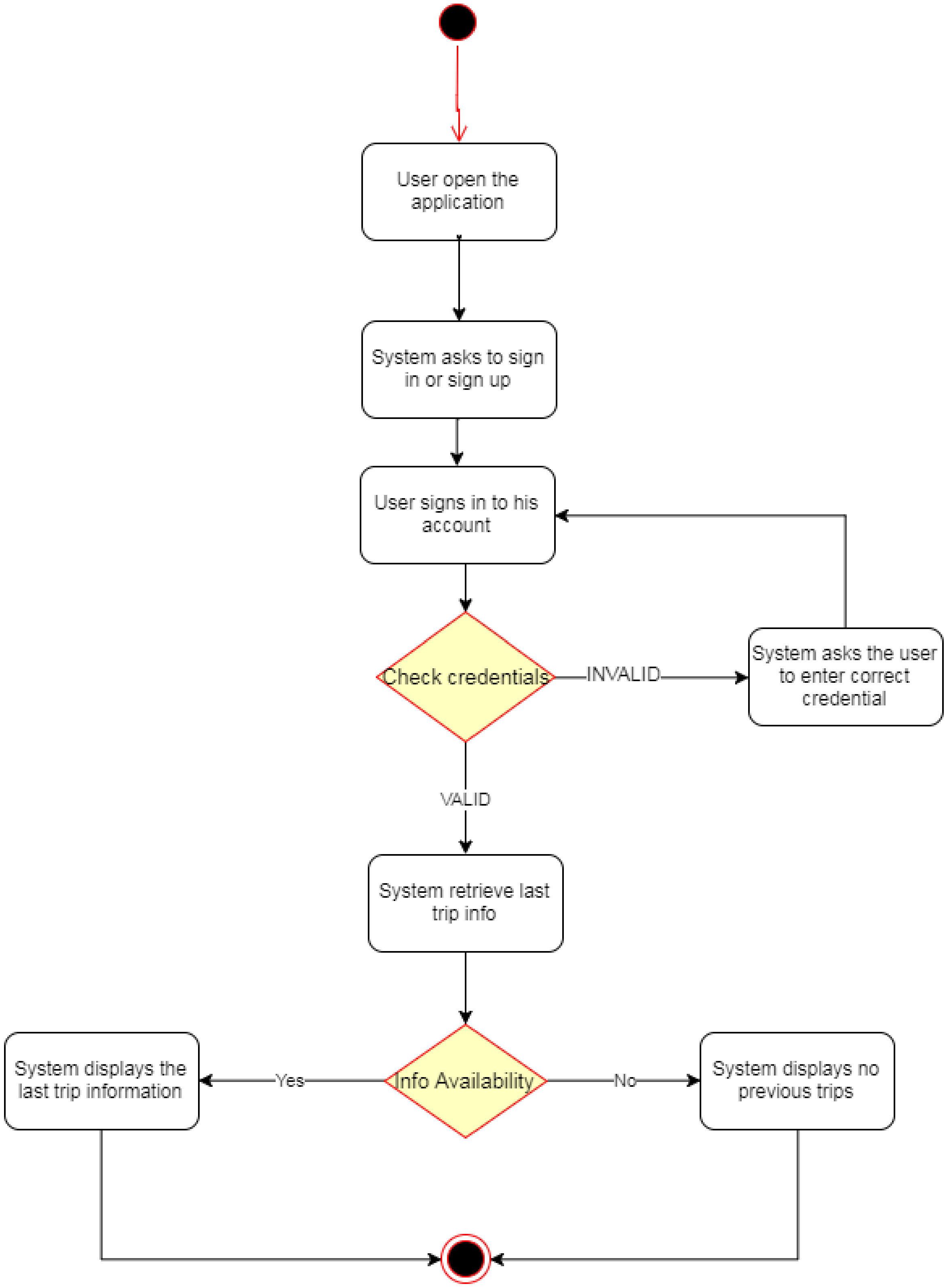
3) Display Trip Data



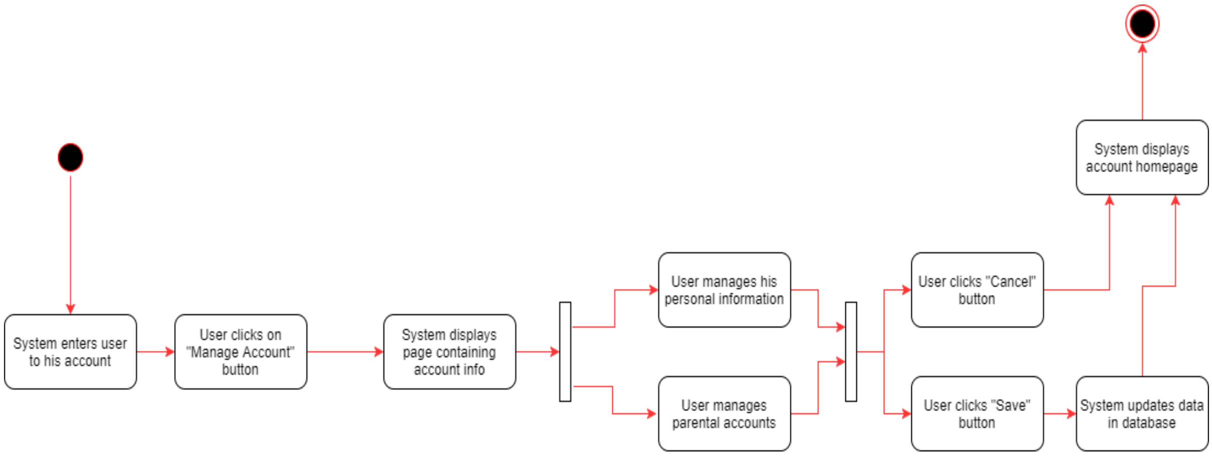
4) Driver Location


User opens the web application
System asks to monitor driver or view profile
User chooses monitor driver
Drivers available
Yes
No
System fetches drivers coordinates from the device
System displays no available drivers to monitor
System sends the coordinates to google maps API to be processed
Google maps API
System displays drivers location

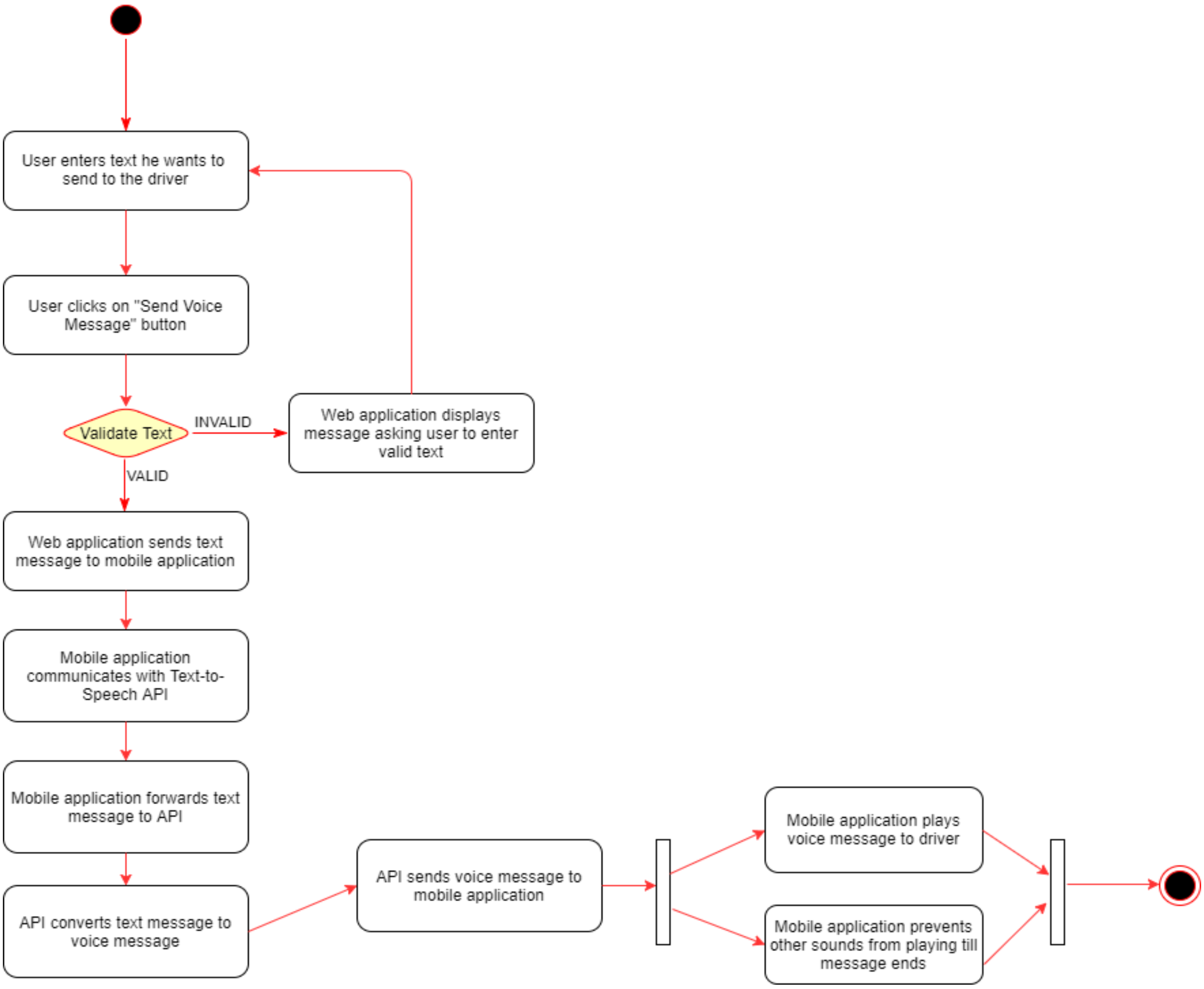
5) Login Activity


User open the application
System asks to sign in or sign up
User signs in to his account
Check credentials
INVALID
System asks the user to enter correct credential
VALID
System retrieve last trip info
Info Availability
Yes
No
System displays the last trip information
System displays no previous trips

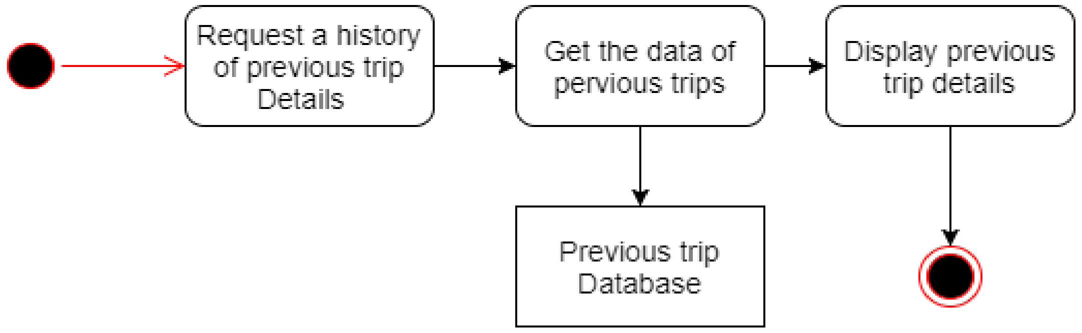
6) Manage Account


System enters user to his account
User clicks on "Manage Account" button
System displays page containing account info
User manages his personal information
User manages parental accounts
User clicks "Cancel" button
User clicks "Save" button
System updates data in database
System displays account homepage

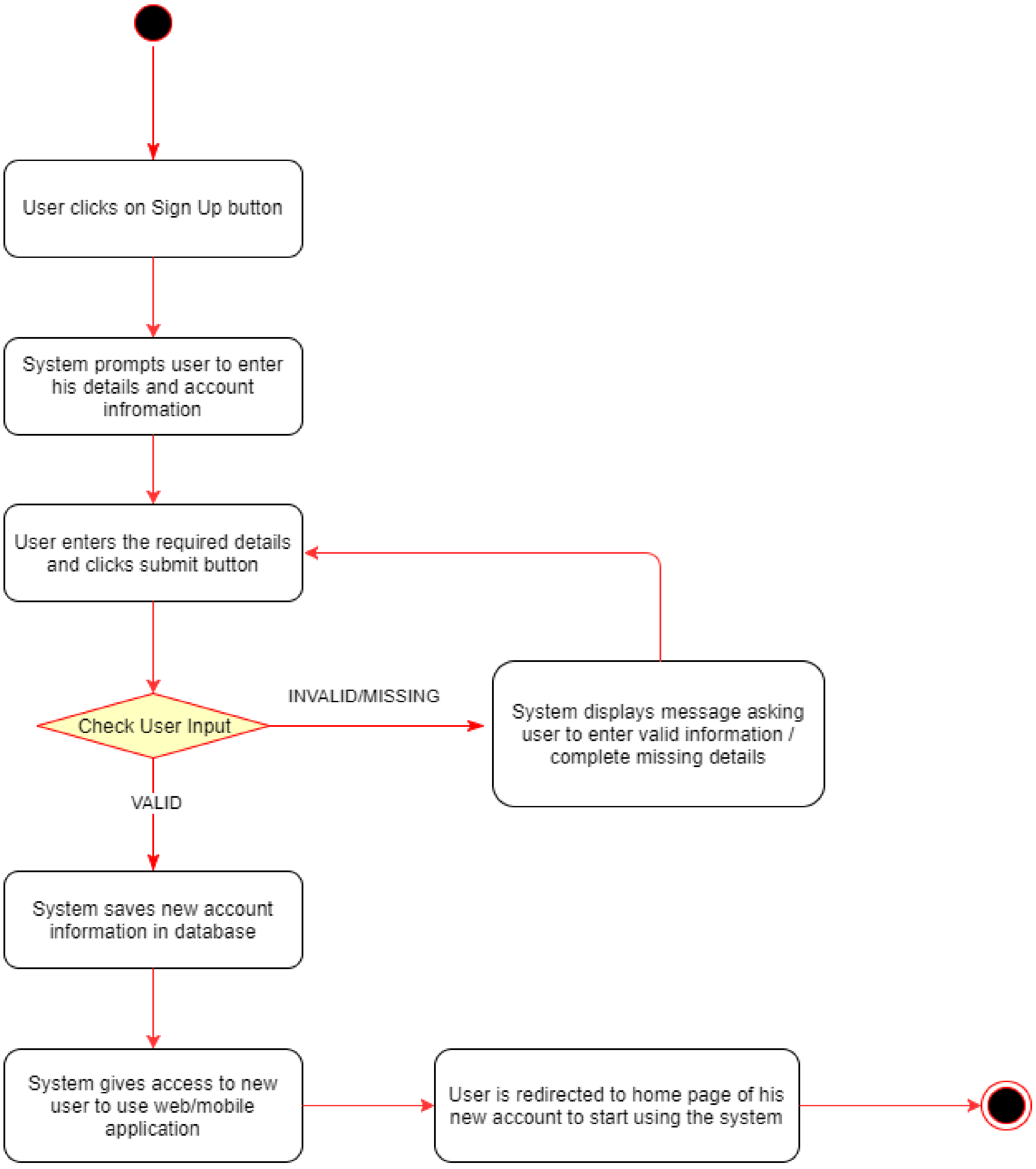
7) Send Voice Messages


User enters text he wants to send to the driver
User clicks on "Send Voice Message" button
Validate Text
INVALID
Web application displays message asking user to enter valid text
VALID
Web application sends text message to mobile application
Mobile application communicates with Text-to-Speech API
Mobile application forwards text message to API
API converts text message to voice message
API sends voice message to mobile application
Mobile application plays voice message to driver
Mobile application prevents other sounds from playing till message ends

8) Show Previous Trip Details

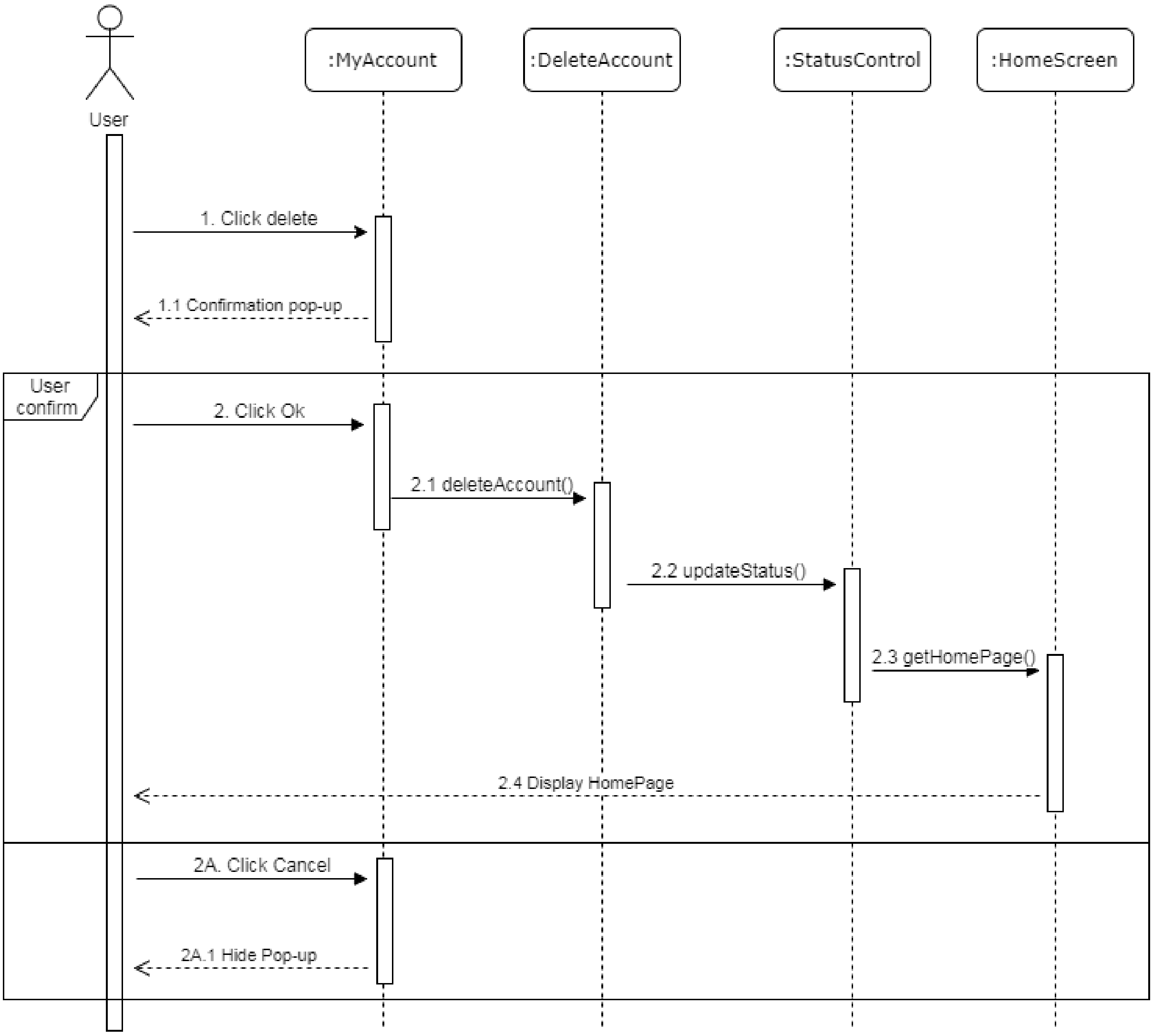

Request a history of previous trip Details
Get the data of pervious trips
Display previous trip details
Previous trip Database

9) Sign Up

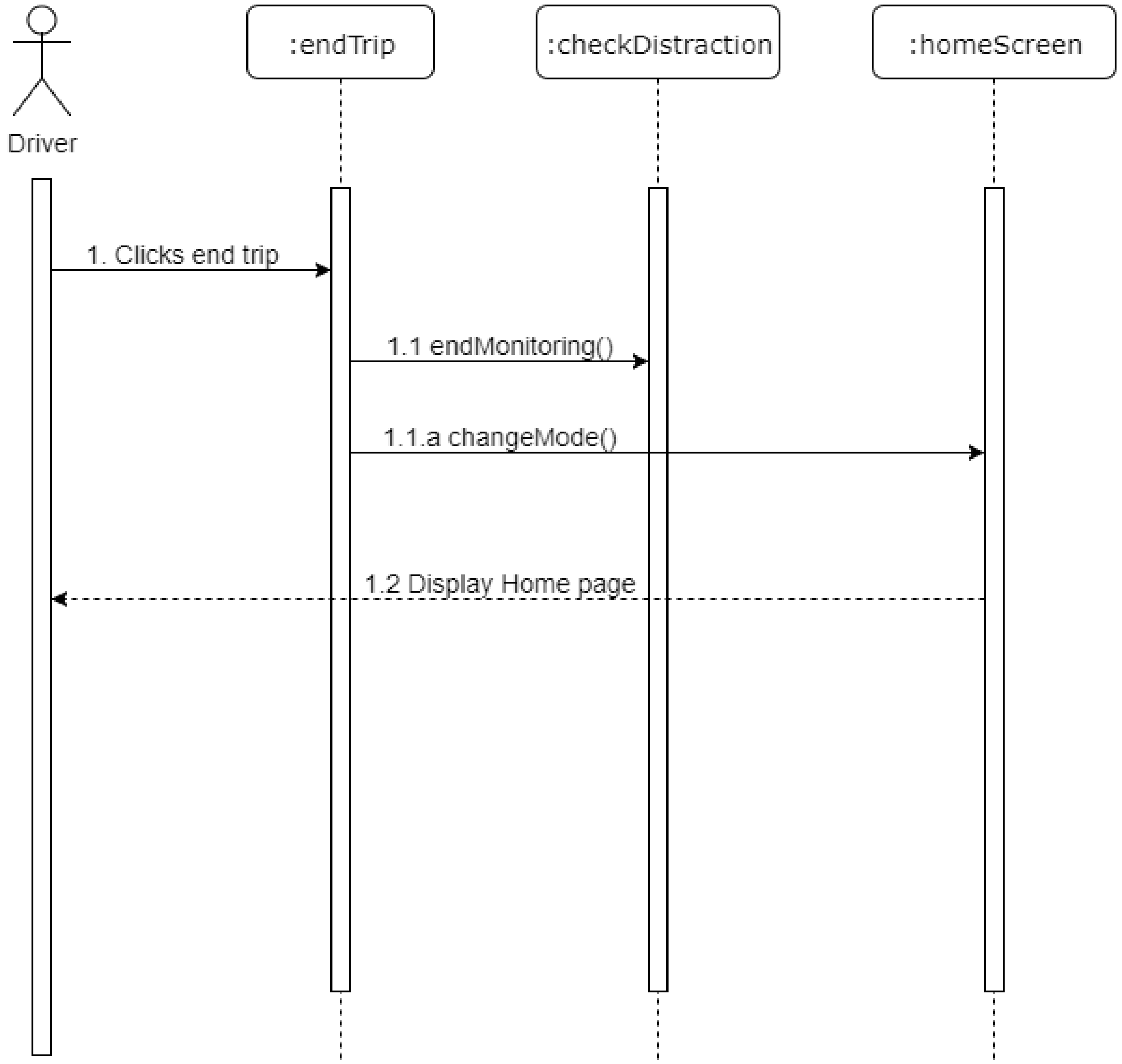

User clicks on Sign Up button
System prompts user to enter his details and account infromation
User enters the required details and clicks submit button
Check User Input
INVALID/MISSING
System displays message asking user to enter valid information / complete missing details
VALID
System saves new account information in database
System gives access to new user to use web/mobile application
User is redirected to home page of his new account to start using the system

Sequence Diagrams

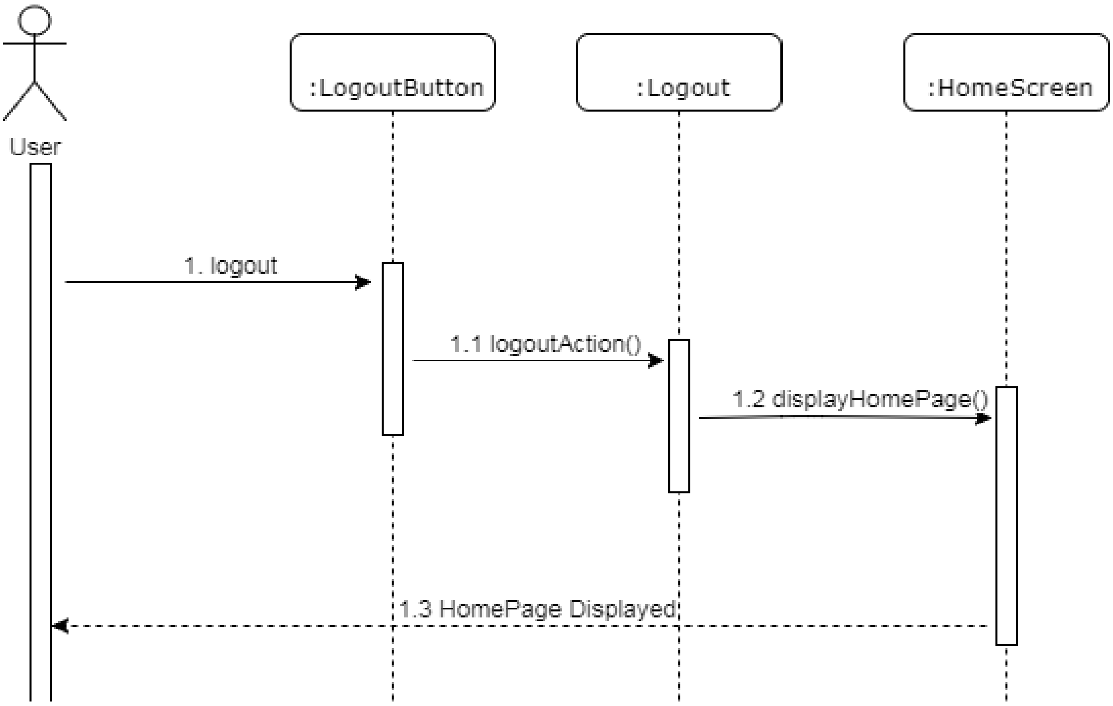
1) Delete Account



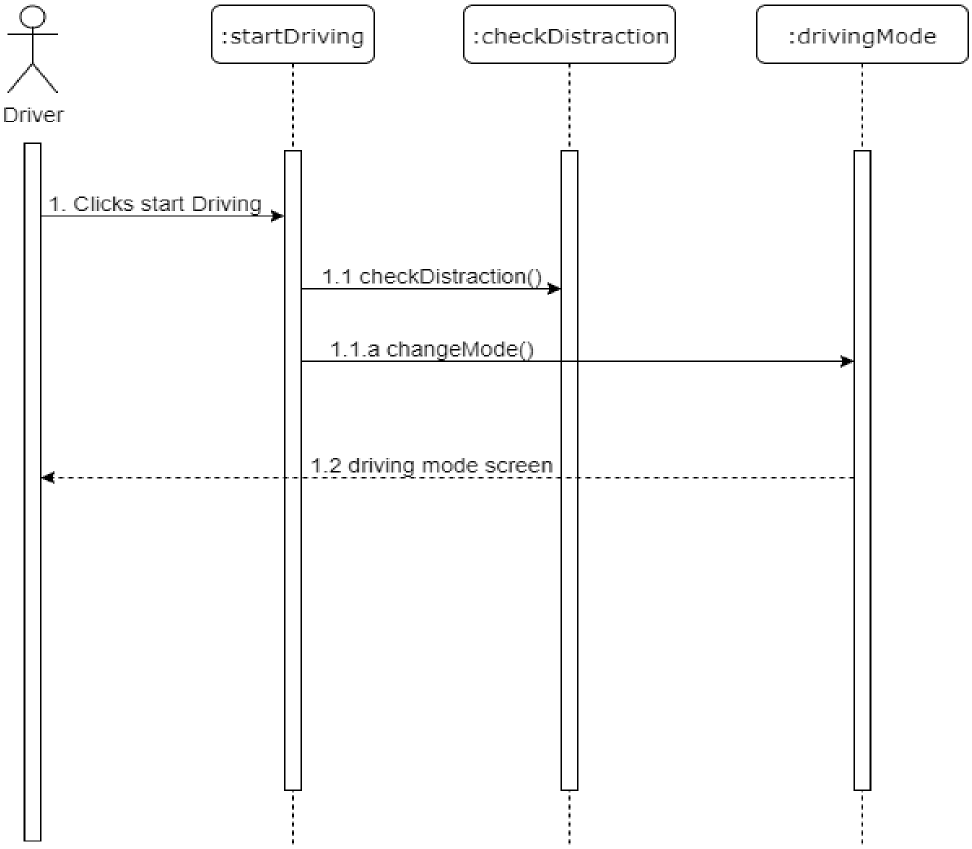
2) End Trip


Driver
:endTrip
:checkDistraction
:homeScreen
1. Clicks end trip
1.1 endMonitoring()
1.1.a changeMode()
1.2 Display Home page

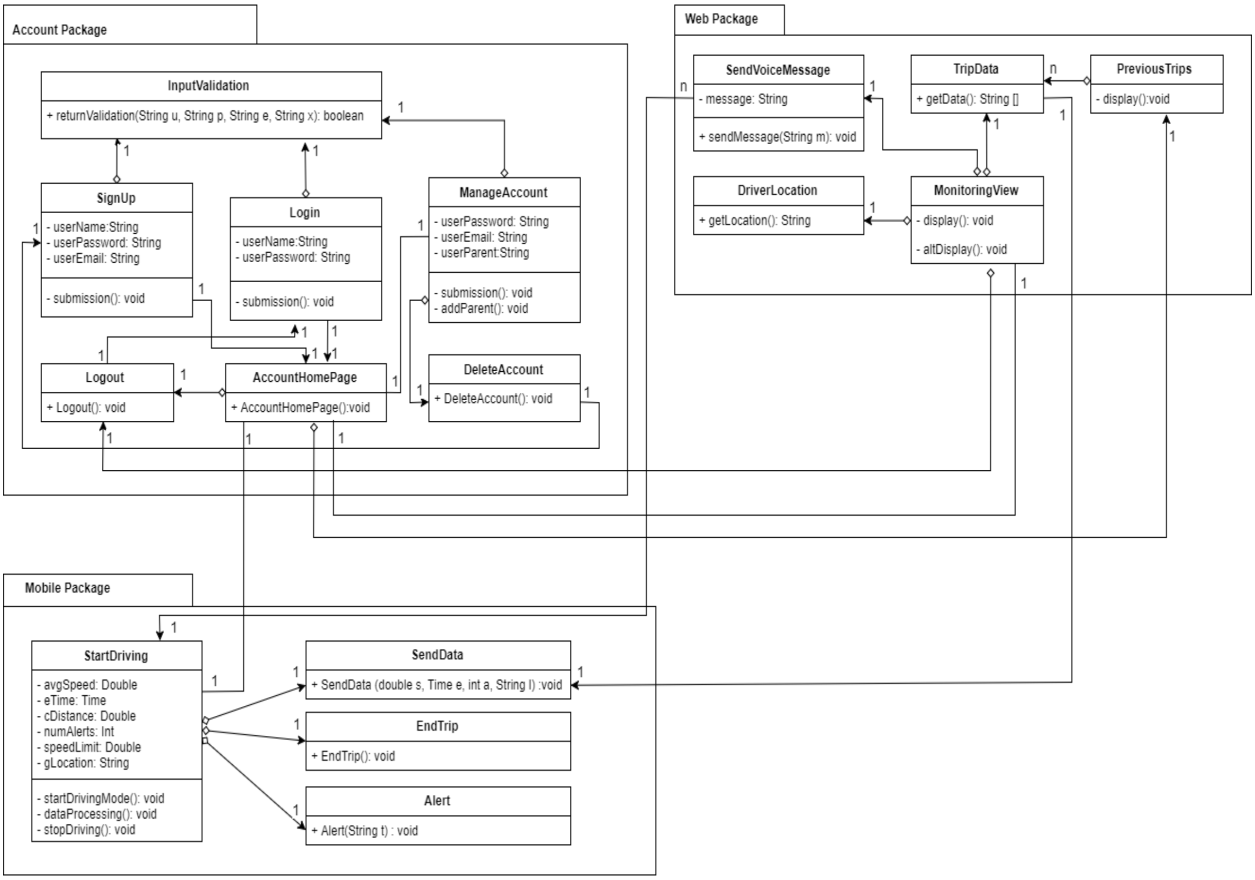
3) Logout



4) Start Driving



Class Diagram



We classified the classes of our system into three main different packages:

- **Account package:** This package is concerned with the classes that relate to the user accounts, how to create, manage or delete them, and access to the system in general.
- **Mobile package:** This package consists of the main classes that are in the mobile application and each class contains functionalities that are available to the mobile user.
- **Web package:** This package will contain all the functionality of the web application, such as displaying trip information in real time while the driver is driving. Beside displaying the history of previous trips.