**Requirements Documentation**

1-Preface

World health organization’s 2015-report referred that the estimated road traffic death rate in Egypt is 12.8% for each 100,000 of the population.Egyptian central agency for public mobilization and statistics (CAPMAS) also referred that the number of road traffic accidents in Egypt had reached 14,500 accidents in 2015 increasing by 1% than in 2014, which caused about 6,203 death cases and about 19,325 injured.

2-Introduction:

Using smartphones to detect road obstacle and conditions is our system which we aim to help drivers to avoid obstacles on their directions and its damages by notifying them with these obstacles before the reach it with enough space to slow down their vehicle. We aim to add some feature to support user safety on the road which will be mentioned later throw this document.

Take advantage of the widespread smart phones among people is our objective to use its seniors to detect road obstacles and collect it in a database. By measuring the change of acceleration on the axis given by the 3-axis accelerometer embedded in smartphone to detect event (obstacles and conditions), and use the GPS to get the location of the detected event to be stored, in addition to use GPS to get drivers locations and notify them with the obstacles on their road, we aim to go farther with our system to use machine learning algorithms to classify the events detected and its damage into categories, System will use Google maps too to support some services required such as getting the road-services to the user and detect traffic road jam. There are many methods to implement road obstacle detection methods by adding new devices -with the sensors required such as accelerometer, GPS and more - to the moving vehicles to collect and classify information about road, but new devices means additional cost on the drivers, on the other side we aim to use smartphones and there embedded sensors with no need to new costs. This may be not as accurate as using a specific device for data processing but the huge number of smartphones and the data will be connected and analyzed with increase the accuracy of processing.

The development of our project will go throw two versions, we will deploy the following services in our first version:  
- Road services identification.

-Stay awake (to check that the user is not sleeping during driving ).

-Road obstacle detection.

-Store events detected in the database.

-Provide location services.  
-provide traffic identification services.

And the second version will deploy the following services:

-Data and events classification and categorization.  
-GUI and Voice Notification.  
-Project optimization (power and data usage optimization).

-User Accounting.

4- User requirements definition

4.1 – Customer Requirements.

* Customer need to be notified with the obstacle on their road before he reach it with “N” distance where “N” is a user specified variable.
* System has two notifications moods (Voice notification or GUI notification); Customer has the choice of the notification mood used in his mobile.
* Customer need to be notified with interactional voice each “t” time to make sure that the user is not asleep during driving which “t” is user specified variable (Stay awake services).
* We need to take care of our users so the user will specify a trusted person who the system will automatically send him help notification in case of dangerous like accidents.
* We need to optimize our algorithms to efficiently use our customer’s internet connection data and battery power.
* Customer need to deliver feedback on system services.
* Customer can cache a specific road data.
* Customer can store a specific location or services data.

**5-System Architecture:**

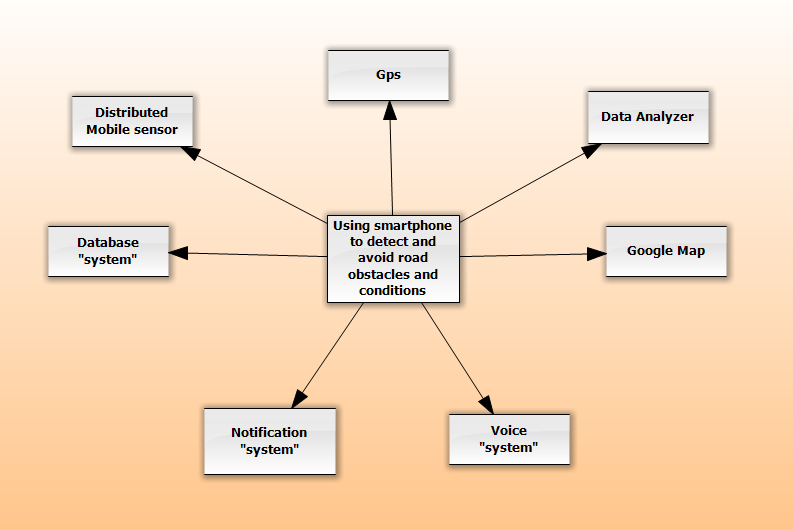
Road obstacle detection system will consist of the following components.

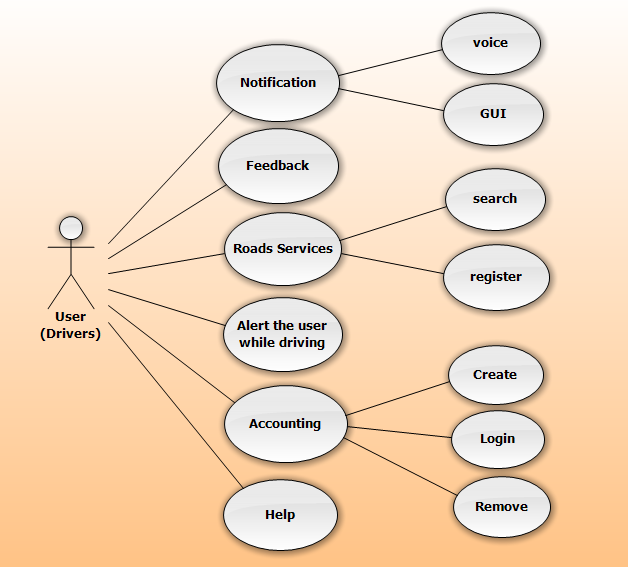
GPS:- to manage and collect locations and some user information like speed road etc.

Data Analyzer:- Data Analyzer is component that responsible for data analyzes and classification, Data Analyzer will be divided into two parts (Local and Global), local “DA” will manage data filtering and remove noisy from sensors data, event detection and data reduction, Global will be a web services depends on machine learning techniques to classify the events detected to categories of obstacles.

Google Map: A component to manga maps interactions.  
Database Manager: to store users data and obstacles information.  
Notification System to manage notification (Voice and GUI).

Use Case and Context Diagrams



****

6-System Requirement Specification

**6.1-Voice alarm**

Function:alert driver to wake him

Description:User needs to be alert with alarm to check that he awake when driving if he sleeps.

Input:none.

Output:voice alarm.

Destination:voice API.

Action: system needs to check if alarm checkbox is checked or not, if it is checked, system will send to voice API to start voice question every “T” time to user, if user reply on the voice question, system will not start alarm voice, if there is no reply, system will start alarm voice, if it is not checked, system will not ask user any voice question.

Requirements: Checked of alarm checkbox.

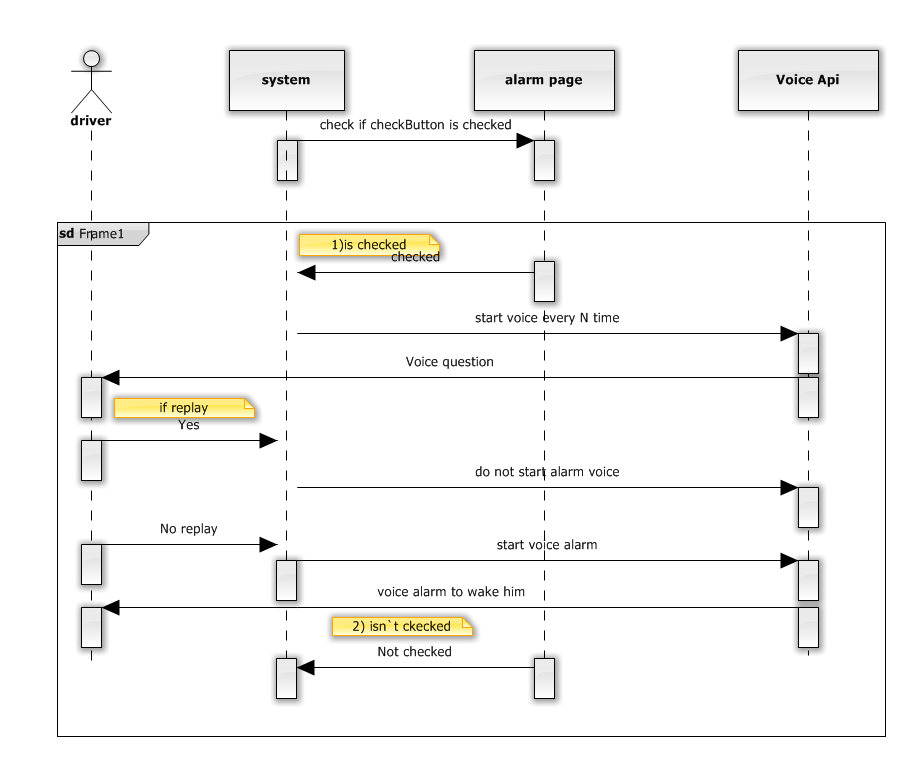
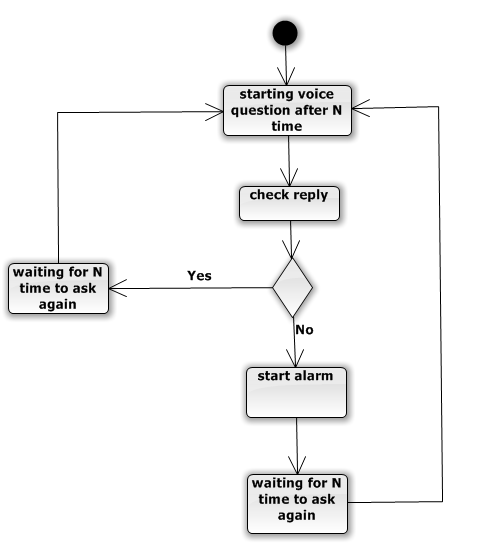
Pre-condition:

1-application must be active.

2-driver needs to be in driving mode.

3-driver needs to check alarm checkbox and determine time for asking him.

Post-condition:waiting for “T” time to ask again.

-Sequence Diagram   
\*note “what will happen after voice a

**6.2-Voice**

Function:notifydriver with obstacles using voice.

Descriptions:user need to be notified with obstacles on road before reaching it by N distance and N time.

Input:location, distance, time(N).

Source**:** GPS, Data Base (global), Data Analyzer (global)

Output: Notify user with voice by obstacles before N time.

Destination**:** API voice.

Action**:** system needs to get current user location from GPS.

System needs to search database to find nearest obstacles to user

System will calculate distance between current location and nearest obstacles (N).

If n<=120 system will start a voice before N distance from obstacles and N time

User will reply to voice then system will stop the voice

If user does not reply, system will wait 2 seconds and start voice again

Requirement:nearest bump

Pre-condition:

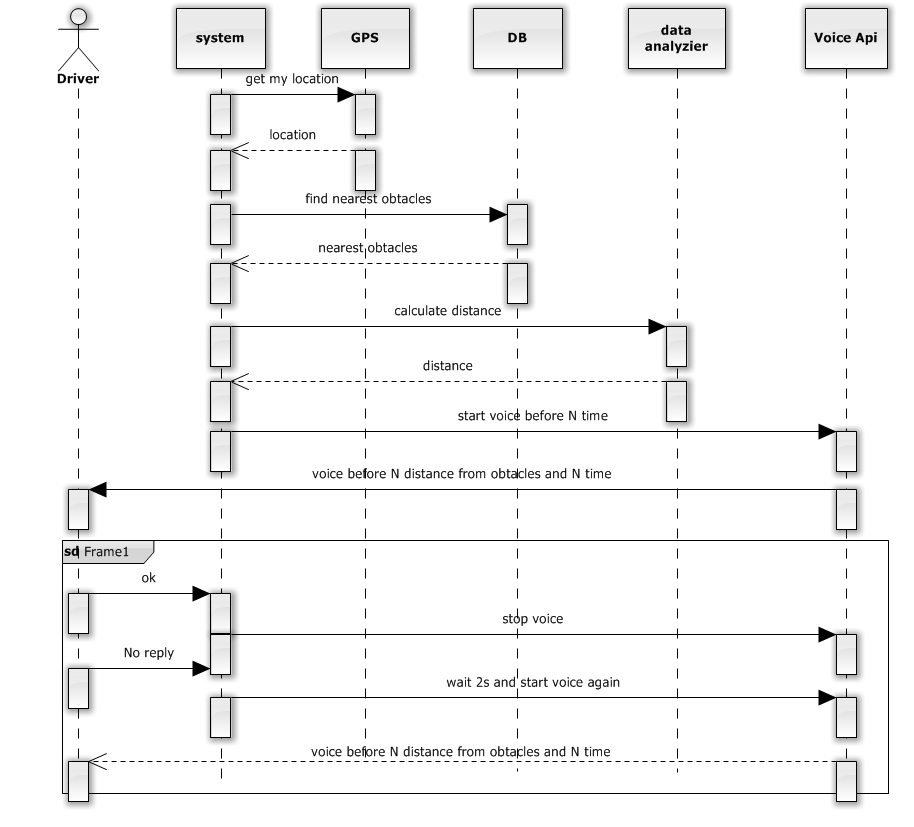
1-application must be active

2-driver need to be in driving mode

3-there is an obstacles

Post-condition:waiting for N time to notify anther nearest obstacles

**\*\*\*function overlapping with find\_nearest\_obstacle().**



**6.3-Accounting**

6.3.1-Login:

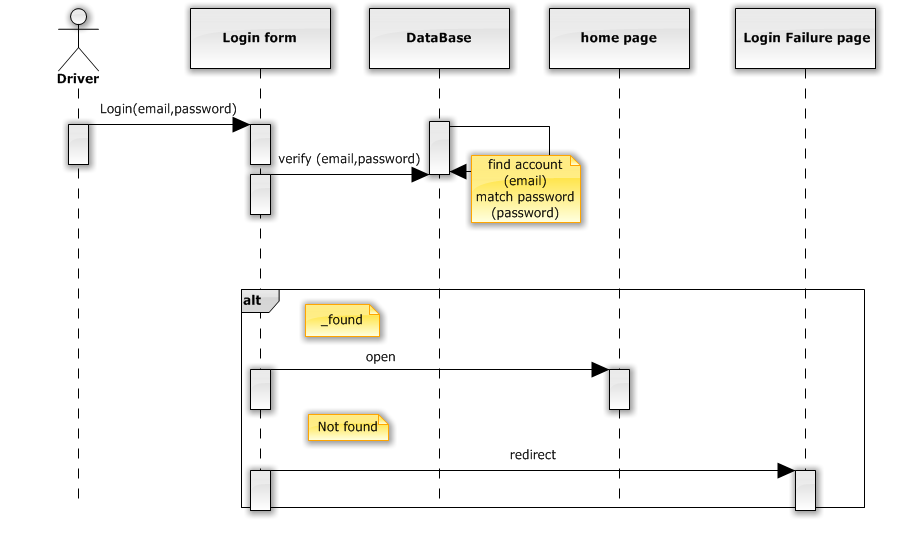
Users require: user need to login in to the system in order to use and control on the features of the application.

System requires:

1-Check email and password for the user in database

2-If found, user can used app system.

3-If not found ,user cannot use the app system and system will go to redirect to login failure page which can create new account for user, or confirm about email or password (try again).



6.3.1-Remove Account:

User require: user want to remove his account from system.

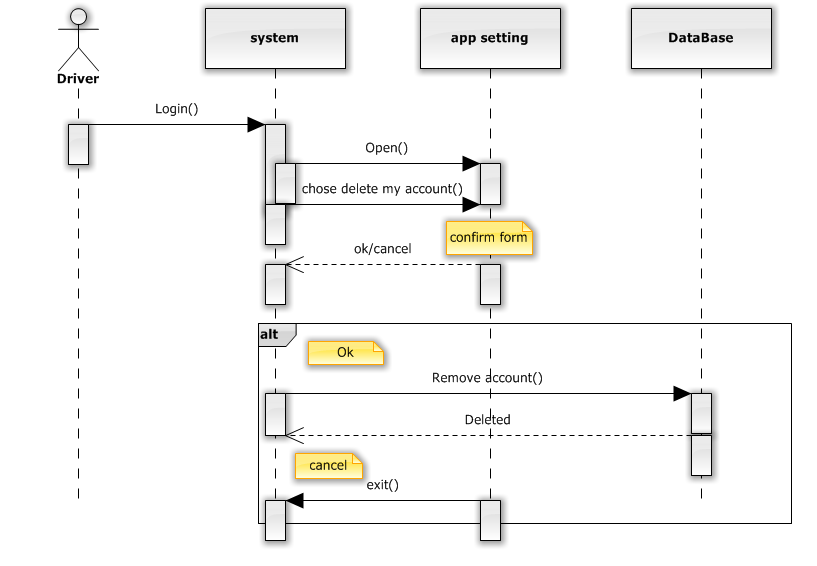
System requires:

1-system open app setting, user choose delete my account

2-system asks for confirmation message form to user.

3-if user check ok, this account will be removed from database, system reply with massage account removed to user then redirect to login page.

4-if user check cancel, this operation will be canceled.



6.3.3-Edit Profile data:

Users require: user want to edit his data in the system.

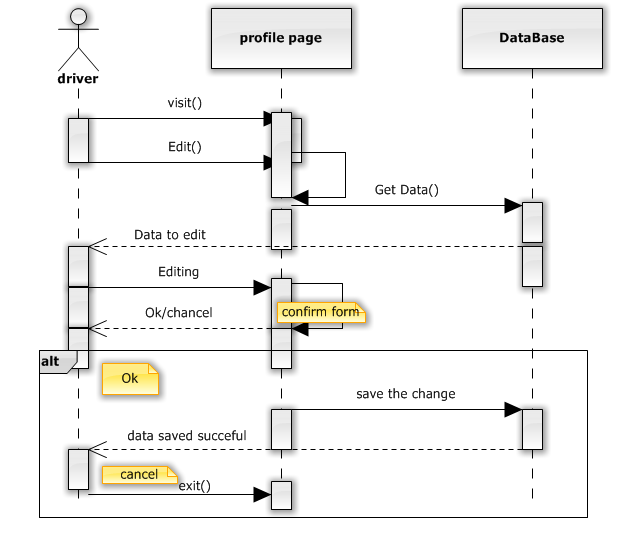
System requires:

1-system will get the data from database.

2-when user finish editing and click ok, system will ask for confirmation form user (ok, cancel).

3-if user check ok data changed will save in database and return message data updated successfully to the user.

4-if user check cancel, operation will be canceled.



6.3.4-Create Account:

User requires:user need to create new account to use app.

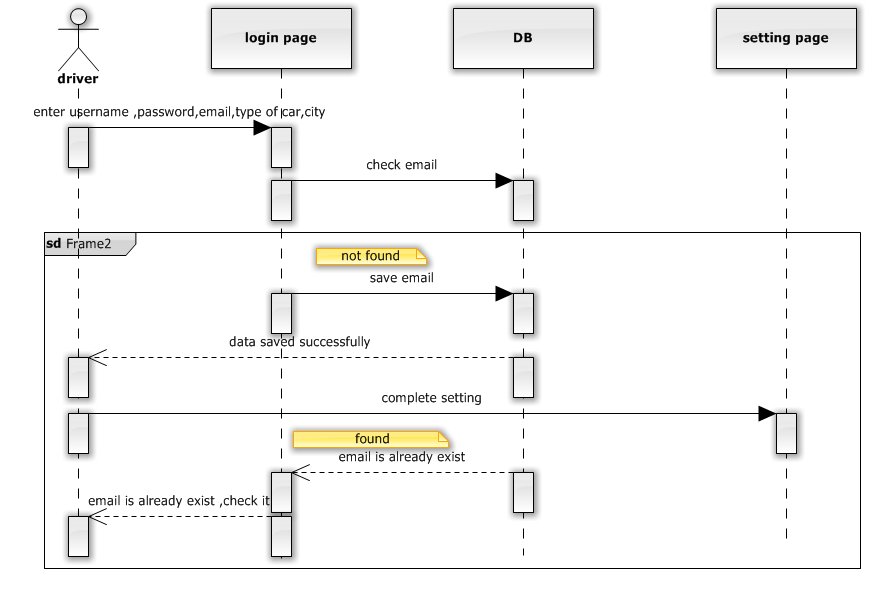
System requirement:

1-driver will to enter required data.

2-system will check email in database if it is found or not

3-if email is not found, system will create new user and save data and send to user that data is saved successfully and ask him to complete setting of his account.

4-if email is found, system will show that email is already found and ask user to make sure of his email.



\*\*diagram update.

**6.4-Find road Services**

We will add an Android service which displays nearby places in Google Maps Android API V2 corresponding to the user current location in the Google Maps and place type selected in the Spinner view. On clicking a marker, the place details like place name, vicinity and photos -if available- will be shown in dialog fragment window. \*\*\*specified of customized

Classes for doing this service:-

PlacesTask: This is an inner class of the MainActivity class and is extending the class AsycTask. PlacesTask is used to fetch JSON data from Google Places Web Service in non-ui thread.

ImageDownloadTask:This is an inner class of PlaceDialogFragment and is extending the class AsyncTask. This is used to download image from Google Places Web Service.

Place: This class represents a nearby location with the information like latitude, longitude, place name, vicinity and photos. We are making this as a Parcelable class in order to retain the instances of this class during screen rotation.

Photo:This class represents a photo returned from Google Places Web Service with the information like width, height, photo reference and attributions. This is also a parcelable class.

Attribution: This class is used to represent photo attributions. This is also a parcelable class.

PlaceJSONParser:A utility class to parse JSON data.

Function: find nearest “X” services.

Description: user needs to find “X” services in his position.

Inputs: User location - Service “X”.

Outputs: map with the “X” services marker location and the road to reach this service.

Action: the application will get the user location then take the category of the service that the user entered.

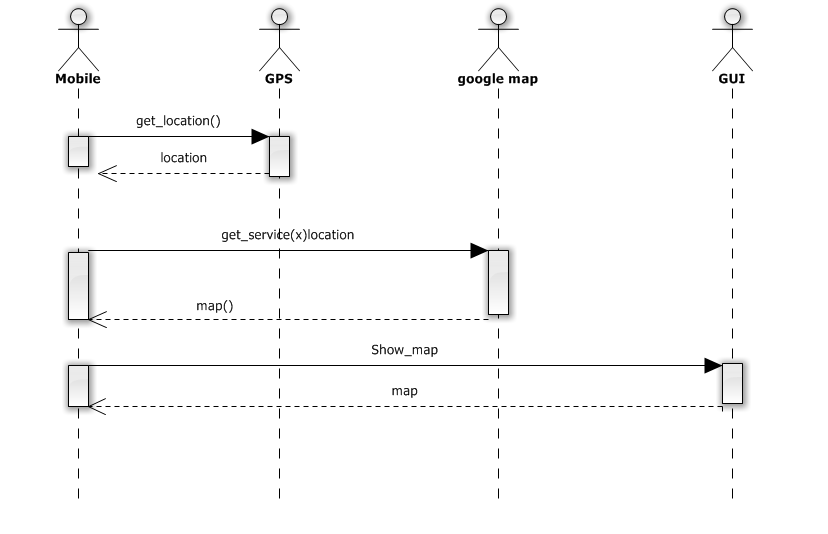
the application get the locations which have the same type and near to the current location of the user, then show the map to the user and put markers to the locations that the application get them and allow to show image to user when select a specific marker. If the user choice one location the application will show the road he has to take to reach the location.

Pre-condition:-

-The user GPS should be turned on.

-The user should turn on the Internet.

-The user should enter the type of the service.



6.5-Finding traffic jam using Google maps

Function: discovering the traffic crowded regions on Google maps.

Description: the function would view the Google maps and allow user to choose from multiple options.

Input: the current location of the user ‘Longitude and Latitude’ of the user.

Output: the activity of Google maps.

Destination: GUI. \*\*\* Web browser

Action: user would ask about the traffic state in the current moment.

-the application should turn the user to the Google maps activity and allow user to choose from multiple options by clicking on the top right button and then choose ‘Traffic’. \*\*\*Auto-select traffic

-The map would be changed into colored one with four main colors.

-Green color: representing a normal speed of traffic.

-Yellow color: representing slower traffic conditions.

-Red color: indicating congestion.

-Dark red color: (previously red and black) indicates nearly stopped or stop and go traffic.

Requirements: the current location of the user.

Pre-conditions:

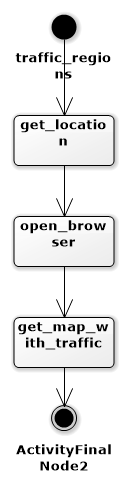
-mobile phone would be turned on.

-the application is opened.

-GPS Is enabled.

Post-conditions: none.

Side effects: none.



6.6-Find nearest Obstacle and sent notification

System requirements:

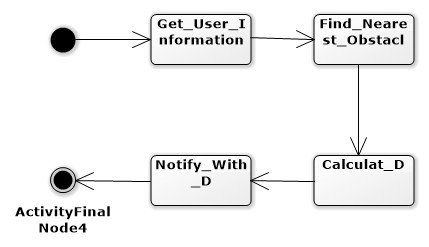
2-System will detect if user is moving or not throw GPS.

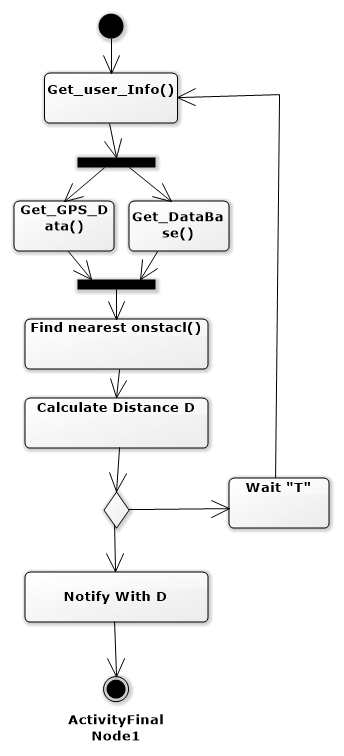
2-If user moving system will get user information including (location – road – direction- speed – N).

3-After collecting user data system will calculate the distance D to determine when to send obstacle notification according to the next table.

|  |  |
| --- | --- |
| Event | Action |
| If D > N | Calculate delay time “T” the reprocessing |
| If D < = N | Notify\_With\_D(D) |

-Activity Diagram.



Function: Get\_User\_Info()

Description: get user required information

Source: GPS – Database

Output: User Info object

Source: Main App activity  
Action: System will update user object data, using GPS to get user location, speed, road and get N from database

Pre-Conditions:

-user info object is predefined

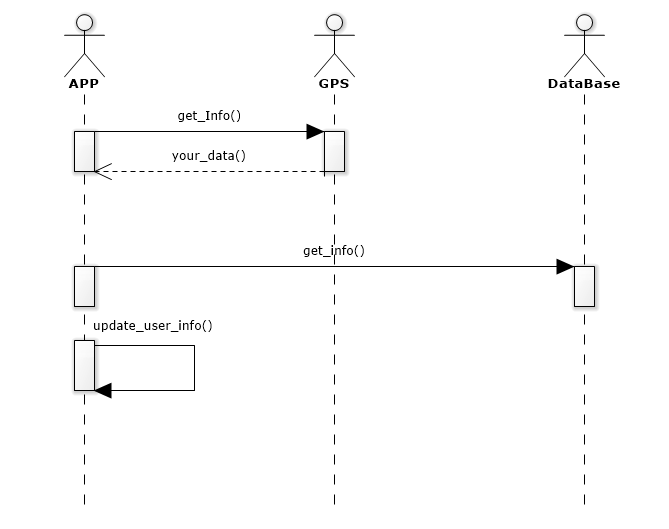
-GPS and location services is activated

-Mobile Internet.

Post-Condition:

-user info object data are updated and =! Null

-Sequence Diagram.

Function: Find\_nearest\_obstacl()

Description: find the nearest obstacle to be notified with to avoid.

Input: user\_Location - user direction – road

Source: Main App Activity

Output: nearest obstacle object

Destination: Main App – data Analyzer to calculate “D”

Action: system will use user’s current location to find the nearest obstacle existed in the database and will return its information to update nearest obstacle object data and will call the data analyzer to calculate “D”.

Pre-conditions:

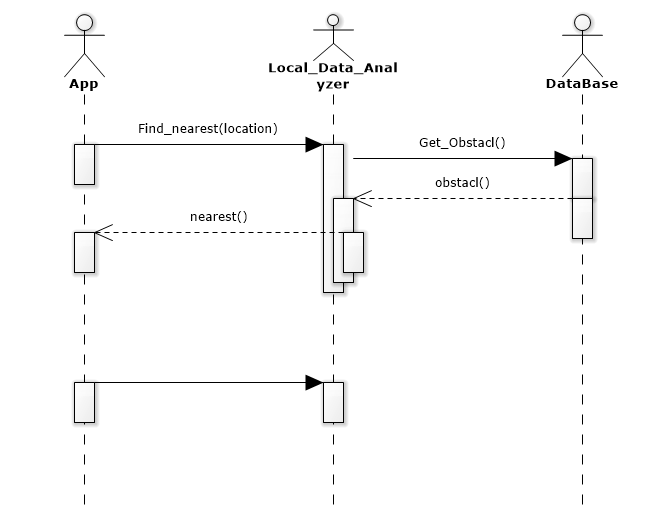
-nearest obstacle object is predefined.

Post-conditions:

-nearest obstacle data updated and != null

-data analyzer called.

-Sequence Diagram.



Function: calculate\_D()

Description: calculate the distance between the current user location and the nearest obstacle to be notified with.

Inputs: use\_location - nearest\_obstacle - “N” - speed

Source: find\_nearest() - user\_info

Outputs:

- “D” distance between the user and the obstacle.

-time “T” to wait until sent the notification id there are delay.

Destination: notify\_with\_obstacle()

Action: Data analyzer will use the location of the user and the nearest obstacle to calculate the distance “D” and will take the decision if it will sent the notification with “D” or will delay the notification according the table

|  |  |
| --- | --- |
| Event | Action |
| If D > N | Calculate delay time “T” the reprocessing |
| If D < = N | Notify\_With\_D(D) |

-To calculate the delay time “T” system will calculate distance “M” where (M = D – N)

-and will use the speed of the user to calculate the estimated time to reach M after “T” system will be reprocessing.

Pre-conditions:  
nearest object =! Null

Post-conditions:  
notify\_with\_D() function called.  
  
-Sequence diagram   
