Cell Tower Placement Software Design Document

Joy Abdelmasseh, Mark Nabil ,Mark Samir, Nermeen Haggag

March 8, 2020

1 Introduction

1.1 Purpose

This software design document purpose is to fully describe the architecture of our Cell Tower Placement system. Our system is two parts. The First is a web application that upload an image for extract the coordinates, distances between other antennas after that the X,Y coordinates used in the KNN classifier to suggest the best location.Also, for the customer complain. The second mobile application that mainly monitor the change in the Cell ID and record the location of the change Cell ID. We also provide a fulfilled description about each single stage input,output and algorithms used in this stage, along with a full illustration for each stage requirements and development process.It illustrates the system components and how they interacts with each other.

1.2 Scope

This document targets the communications companies that would use the Cell Tower Placement Application to help them for getting accurate place for pointing an antenna which will save much more time rather than manual inspection. Our purpose project is an automated system to suggest the best location. It is a GPS and extraction system applied on acquired google earth images for extracting buildings and use distances from GPS then preforms pre-processing, feature extraction and classification. The system has the ability of learning from data set tests to enhance the future accuracy of the system.

1.3 Overview Of Document

The software design document includes eight main sections. The first section is an introduction to our system including our scope and purpose. The second section is the system overview illustrating our system workflow. The third section is the architecture design of the system, activity diagram, sequence diagram and class diagram. The fourth section illustrates the database design in details. The fifth section illustrates our component design including the used algorithms and techniques. The sixth section illustrates the human interface design and describe how the user will interact with our system. The seventh section is the requirement matrix that show which components satisfy each of the functional requirements. The final sections are appendices and references.

2 System Overview

Our proposed system "Cell Tower Placement" is a web application that extract an image from google maps then the system detects the buildings and specify which building is the best for antenna placement using the trained model. Also, Mobile application that can read the technology and cell id. This application has to monitor the change in cell id and record the location where network cell id is changed. All collected data has to be uploaded to the database. The user has the choice to upload live or wait till connected ti wifi. Further more the user can enter events that going to attend.

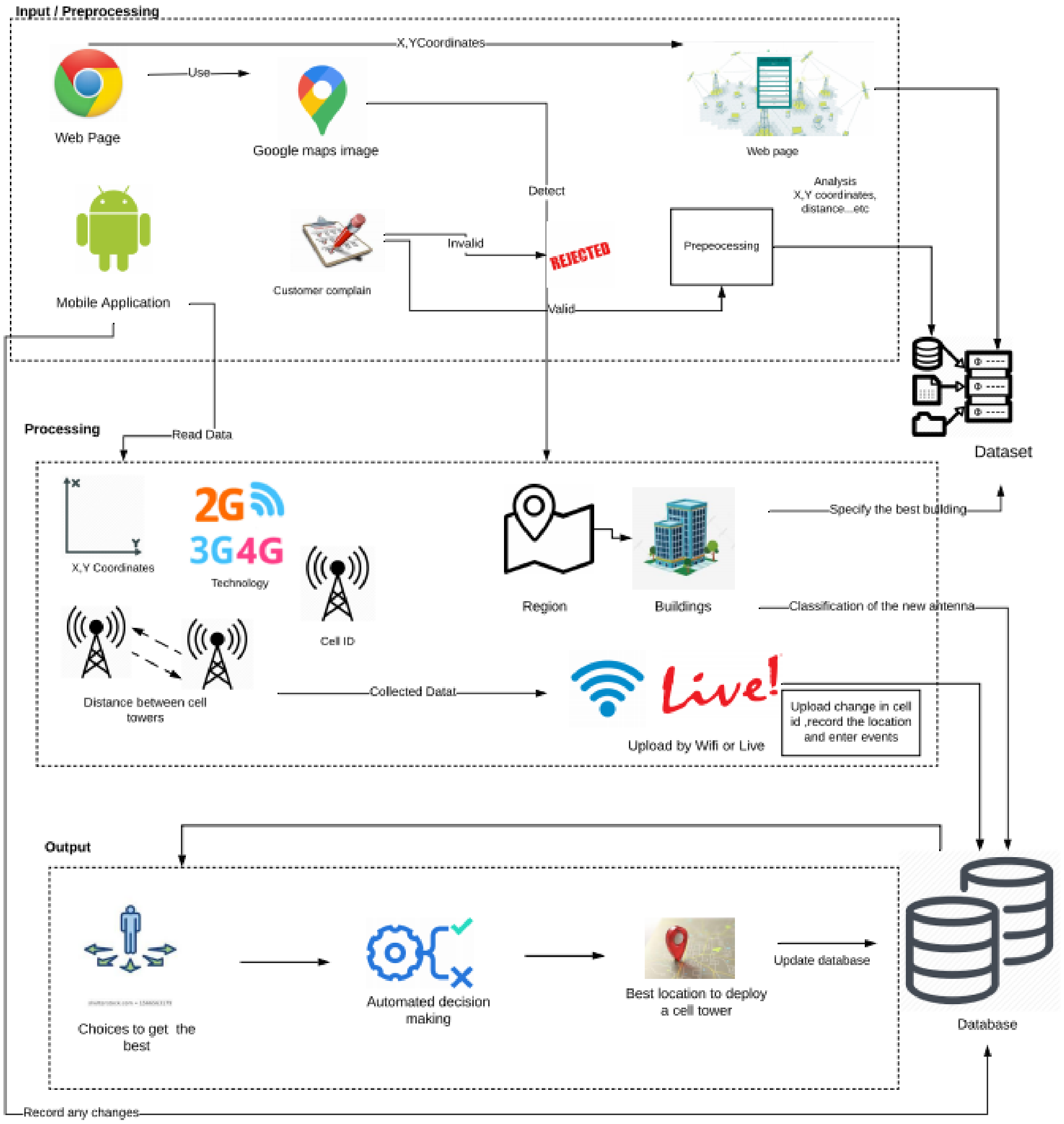

Figure 1: Overview Diagram

3 System Architecture

3.1 Data Design

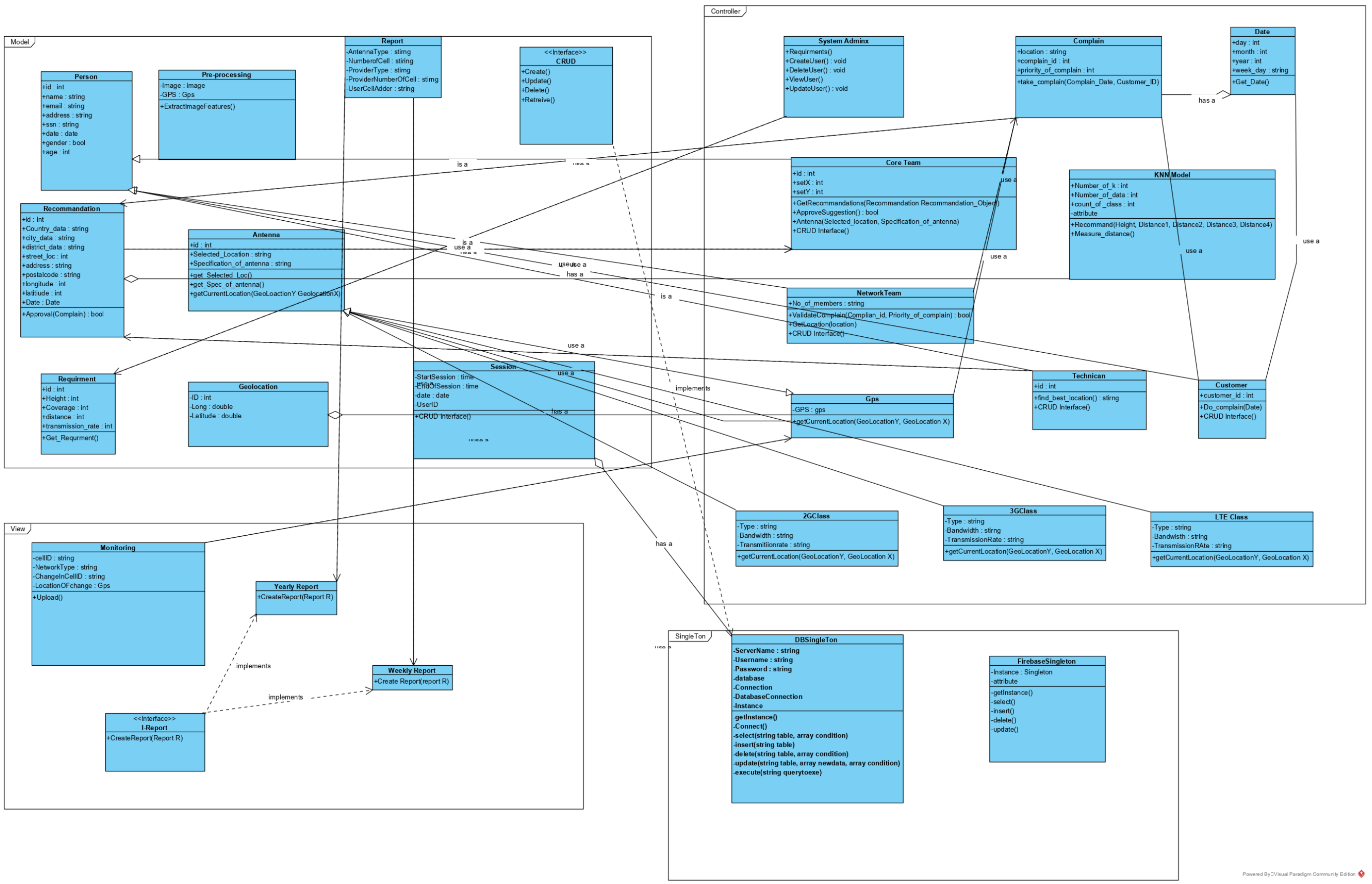

Figure 1 : Class Diagram

3.2 Architectural Design

For the Architectural Design, based on the analysis of the functional and non-functional requirements. The system was designed in a way that would insure the MVC system architectural as shown below:

3.2.1 View

It is responsible for the presentation of data and representing the User interface. We have two types of view. The first type is web page for the input forms and view maps. The second view is mobile application used to view reports and to upload any changes.

3.2.2 Controller

It is responsible for binding the view and model. The interactions and requests made within the view are taken and sent to the database to fetch data with the use of models then it forward data to view again to be shown. Some of the controllers we are having; Admin controller that responsible for handling the requirements that other users will work on. Report controller that responsible for viewing reports to the users when the results are shown.

3.2.3 Model

- The Model is used to connect the web application with the mobile application.

- Algorithm: K-Nearest Neighbours (KNN) is machine learning algorithm based on supervised learning technique. The nearest between the new data with the available data and put the new data into the category that is most nearest to the available categories.

- Libraries: Google Maps is a web mapping service developed by Google. It offers satellite images, streets maps and real-time traffic conditions. We use Google maps API for extracting the features for getting the X,Y Co-ordinates and the distances between antennas. We use numpy library for handling of the array in python. The Sci-Kit learn library which contains the classifier we are using in our system.

- Core: Antenna Location to get the X,Y coordinates from any region to classify the antenna and get the best location.Also, the distances between the antennas to work with the KNN Classifier.

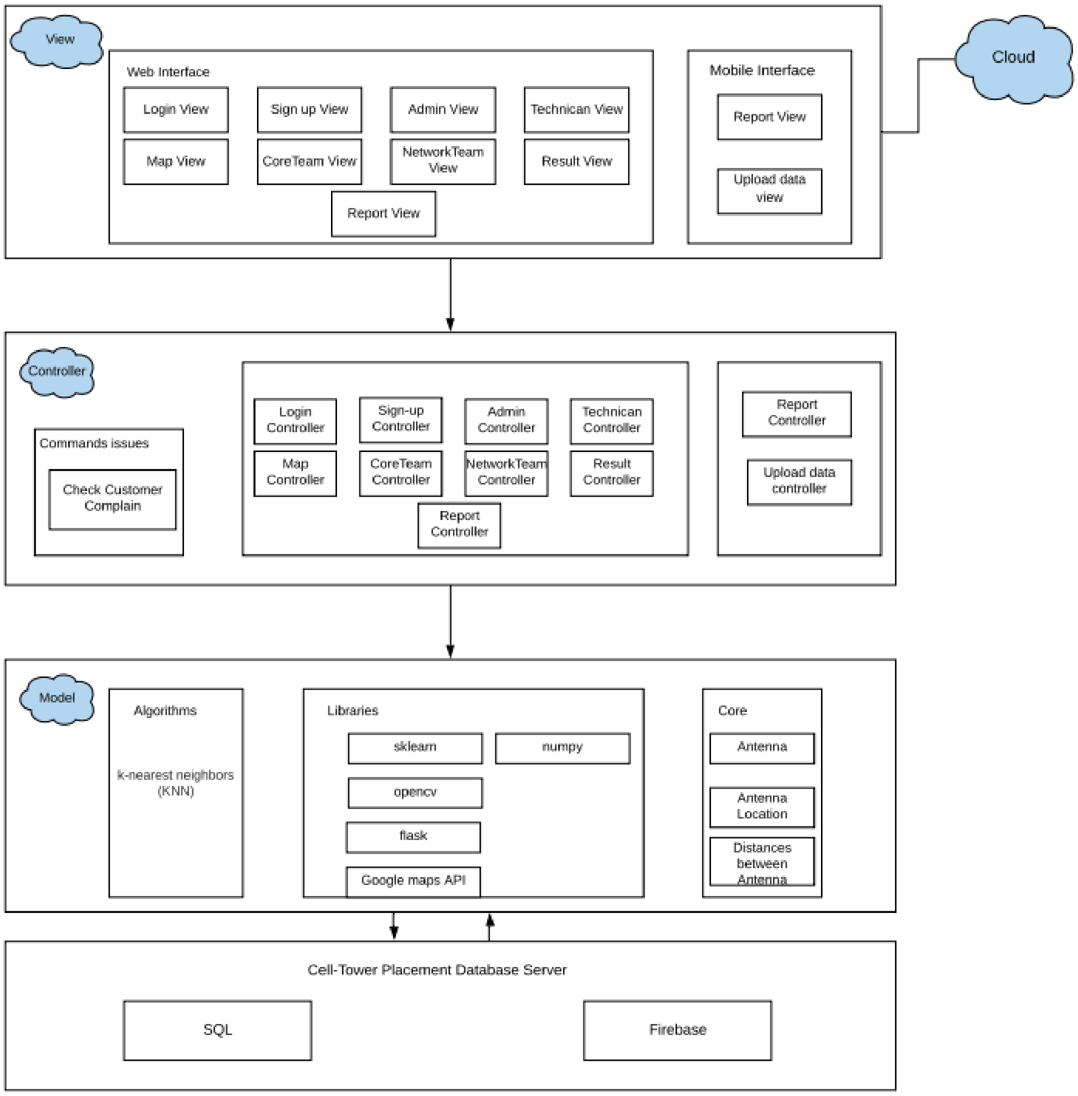

Figure 2: Software Architectural Design

3.3 Decomposition Description

3.3.1 User Authentication Sequence

- Rationale: The user should register for an account before being able to use the application.
- Input: Username, Password.

- Output: Validation response.

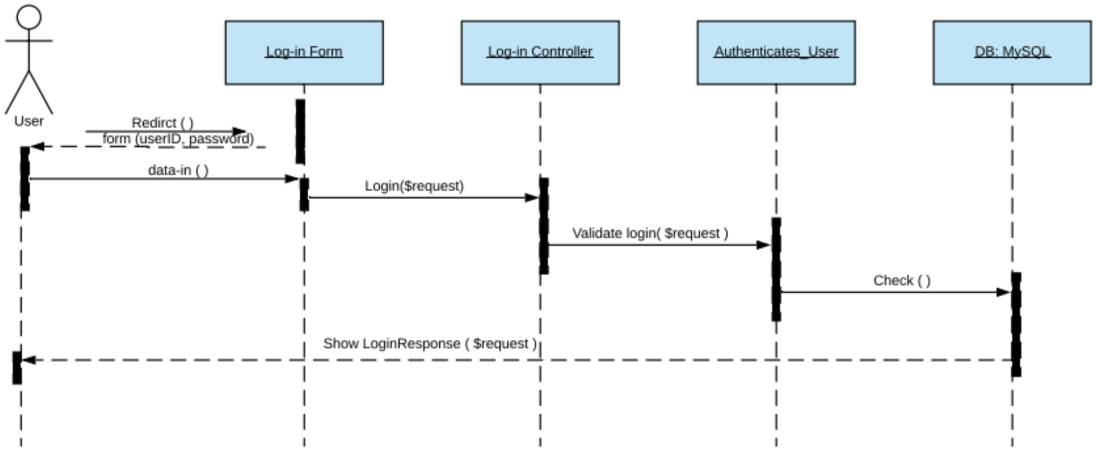

Figure 3: Login Sequence Diagram

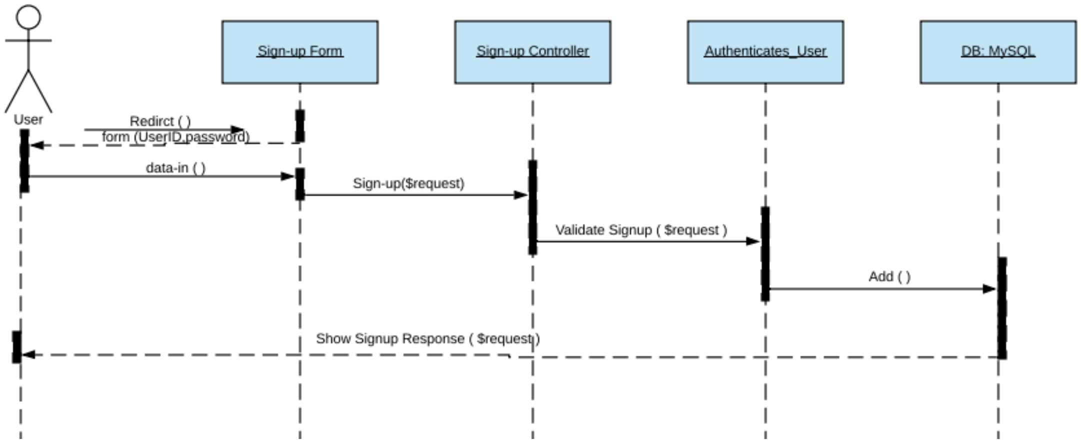

Figure 4: Sign-up Sequence Diagram

3.3.2 Best Location Sequence

- Rationale: The Technician team only show the best location. The core team need to get the requirements from the database to set the X,Y coordinates to the KNN classifier.
- Input: X, Y Coordinates.
- Output: The best location.

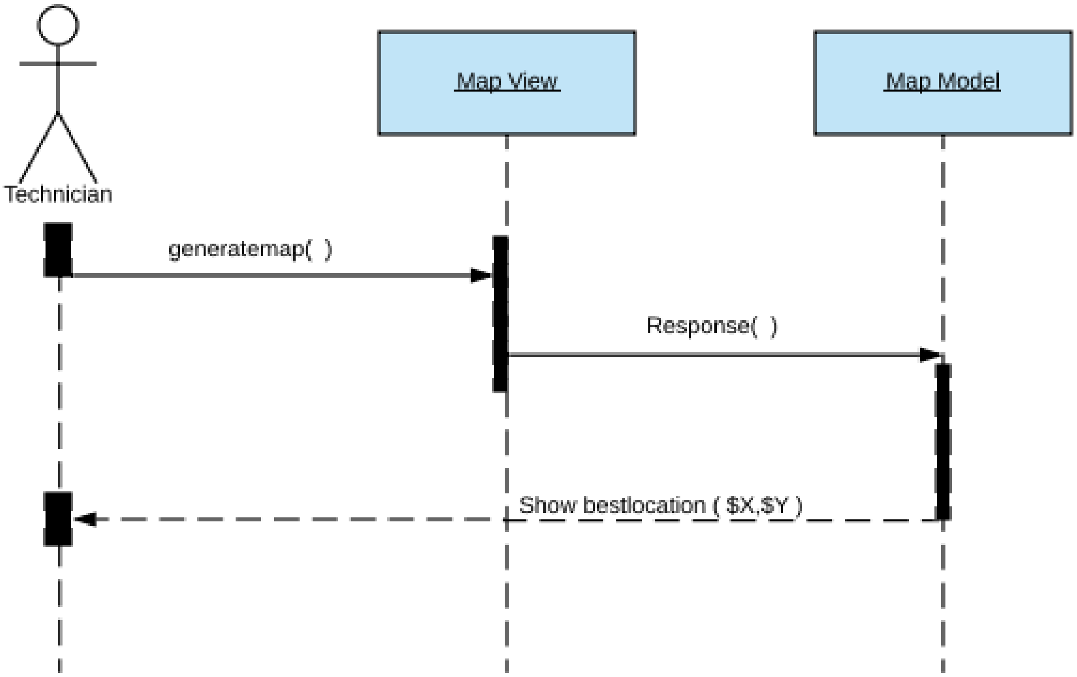

Figure 5: Technician Sequence Diagram

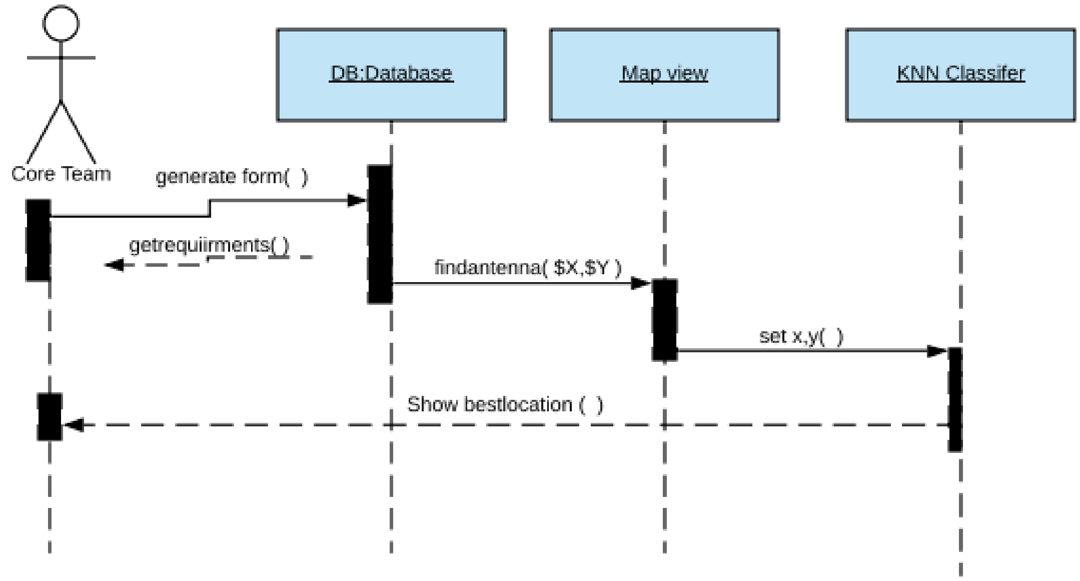

Figure 6: Core Team Sequence Diagram

3.3.3 Extracting Image Sequence

- Rationale: The user upload an image to extract the features to get the nearest five distances and X,Y coordinates .
- Input: Image of google maps .
- Output: Read Technology, Cell ID.

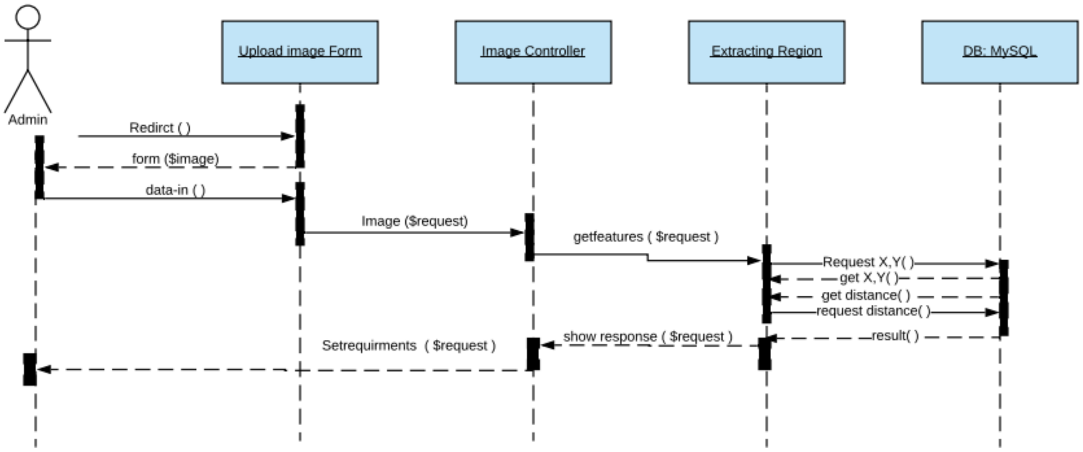

Figure 7: Admin Sequence Diagram

3.3.4 Mobile Sequence

- Rationale: The user upload an image to extract the features to get the nearest five distances and X,Y coordinates.
- Input: Events and upload change .
- Output: Monitoring the change in cell ID and record the location where network cell ID change .

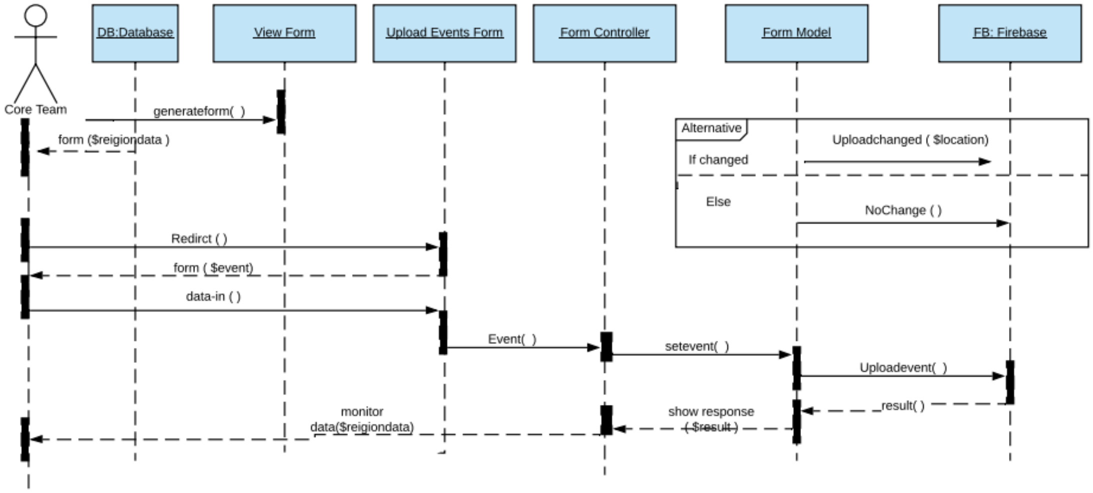

Figure 8: Core Team Mobile application Sequence Diagram

3.3.5 KNN Classifier Sequence

- Rationale: The user upload an image to extract the features to get the X,Y coordinates and nearest five distances and to categories the point using KNN classifier.
- Input: Image and information about the region extracted.
- Output: The best location .

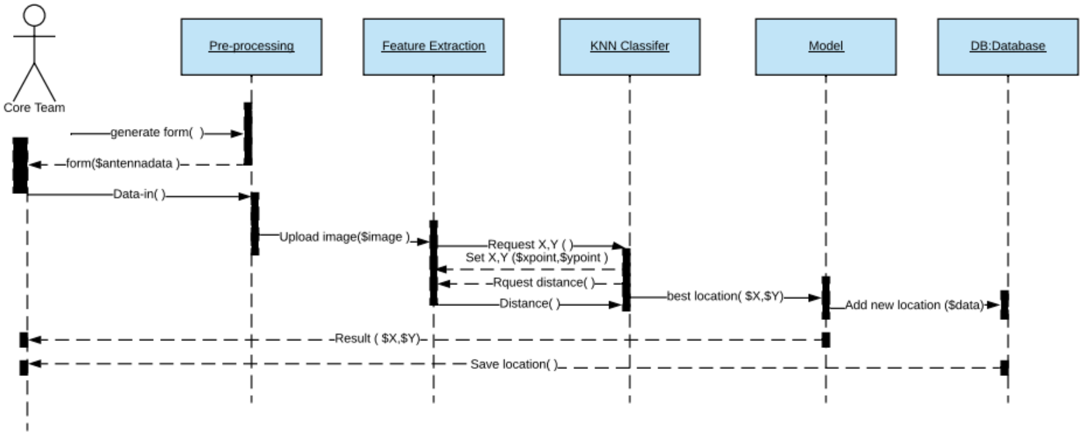

Figure 9: KNN Classifier Sequence Diagram

3.3.6 Activity Diagram

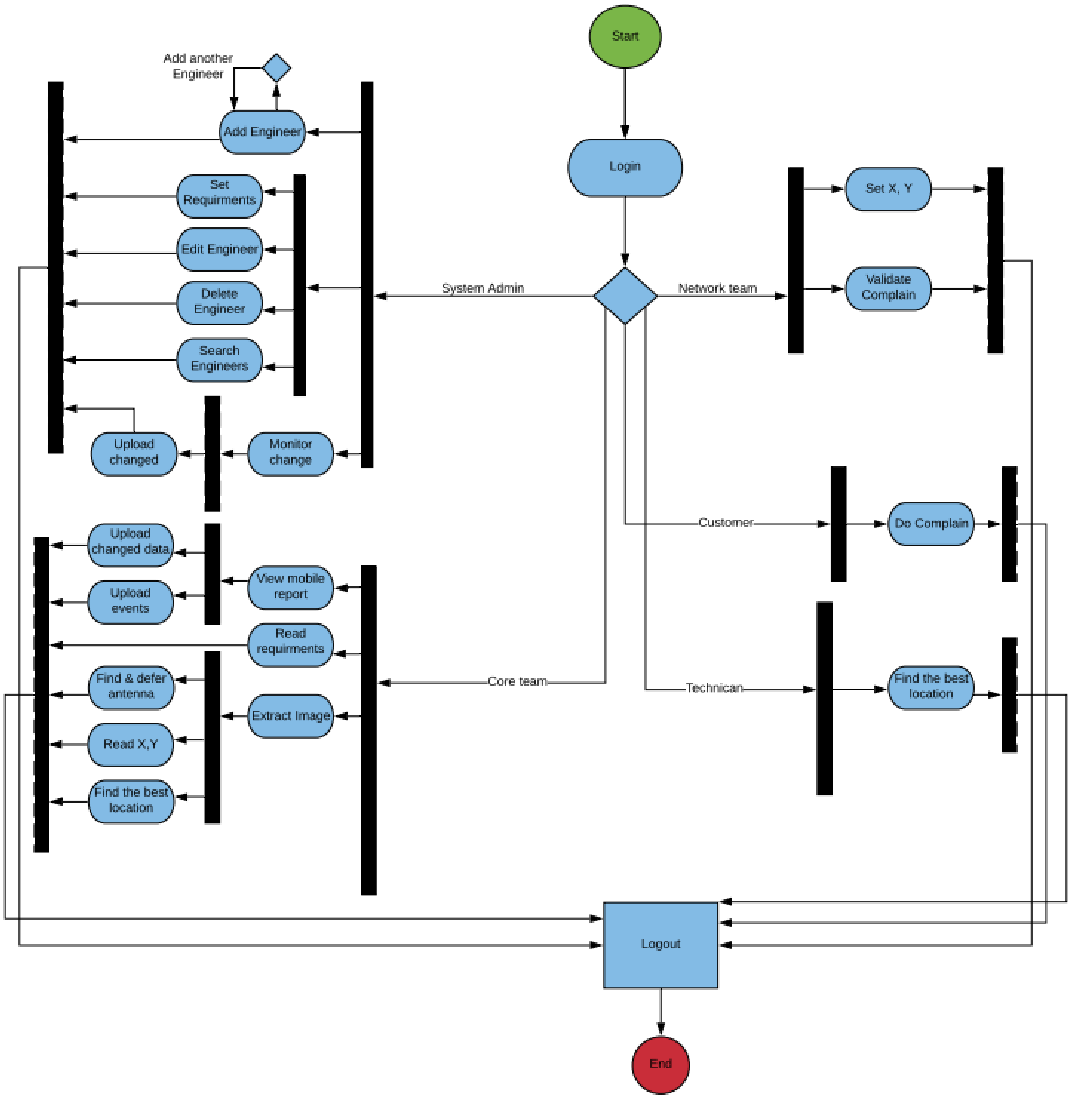

Figure 10: Activity Diagram

3.3.7 Process Diagram

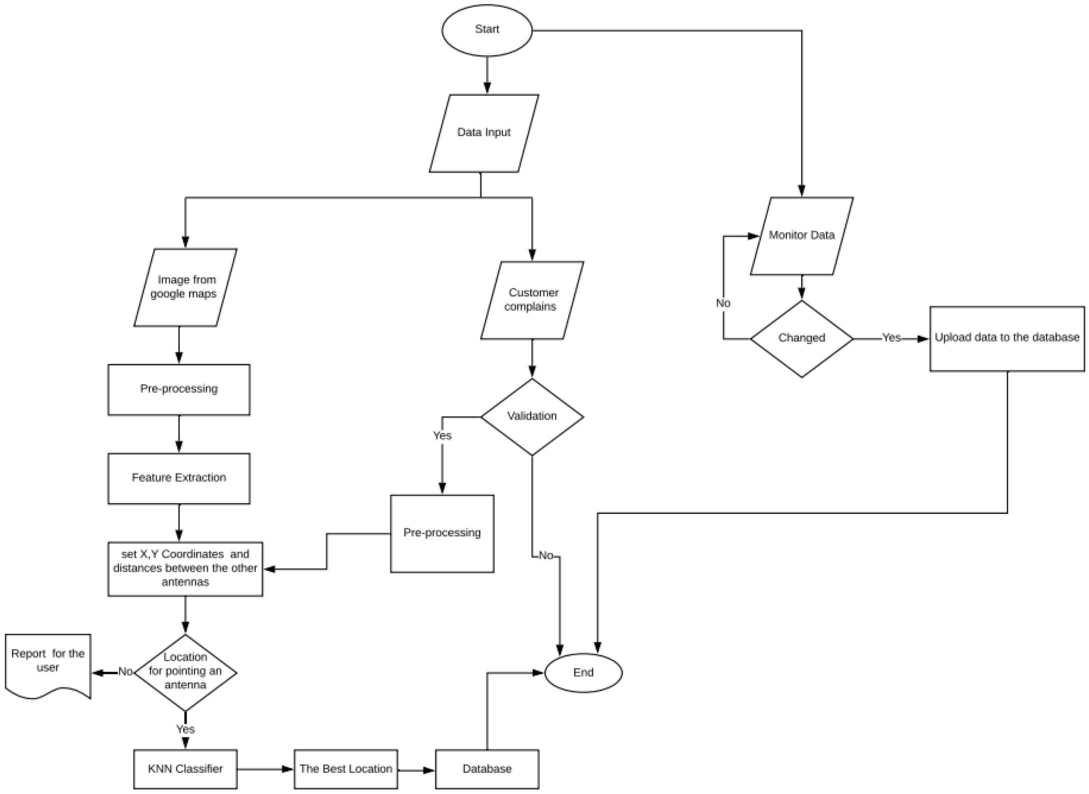

Figure 11: Process Diagram

3.4 Design Rationale

As mentioned previously, we have used Model-View-Controller (MVC) as our architecture as it helped us separate the functionally and data of our system from presentation. So, we can easily make modifications, re-use and optimize functionality part as it is our core. also, the software we are developing efficiency and accuracy is very important aspect of it so it will be very sensitive with data so this should be developed in very accurate and efficient way.

4 Component Design

4.1 Database Diagram

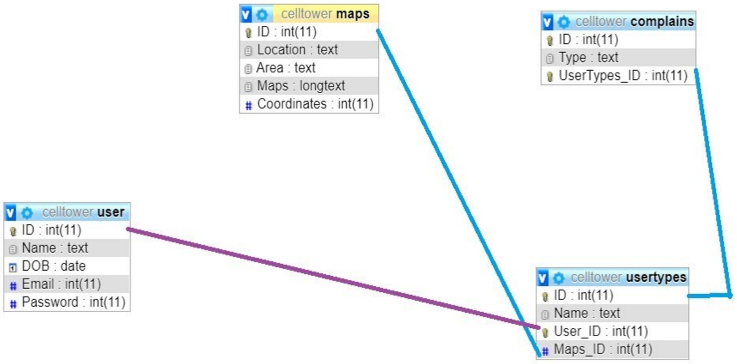

Figure 1:Database diagram

4.2 Input/Preprocessing

In this phase we prepare our data/input. The first input is throw a web application by using google maps to crop or select specific region that face network problem and crop it and start working by image processing/Google API to extract features and specification of the region , like distances between cell towers and XY Coordinates of each cell tower.

4.3 Mobile Application

In this phase we build a mobile application that support our system by monitoring network type and any change in the network cell ID and record the location changes happen in. Collected data is uploaded to data model . The user has the choice to upload live or wait till he/she connected to WiFi. Also, the user can enter events that he/she is going to attend.

4.4 Customer Complain

In this phase if the customer make a complain about network lag or any network problem as call fail. it has 2 phases Accepted phase and Rejected phase, in rejected phase no changes applied. but in of acceptance it enter preprocessing step.

4.5 Database

In this phase we build a big data model that record and track data changes continuously throw the mobile application to store the tracked data as : traffic, annually events , upcoming events. The user has the choice to upload immediately or later using WiFi.

4.6 Classification

In this phase we are using KNN Classifier to get the accurate cell tower location. KNN is Machine Learning algorithms based on Supervised Learning technique. Algorithm work by assuming the similarity between the new data with the available data and put the new data into the category that is most similar to the available categories.. This means when new data appears then it can be easily classified into a well suite category by using KNN.

4.6.1 Visualize data

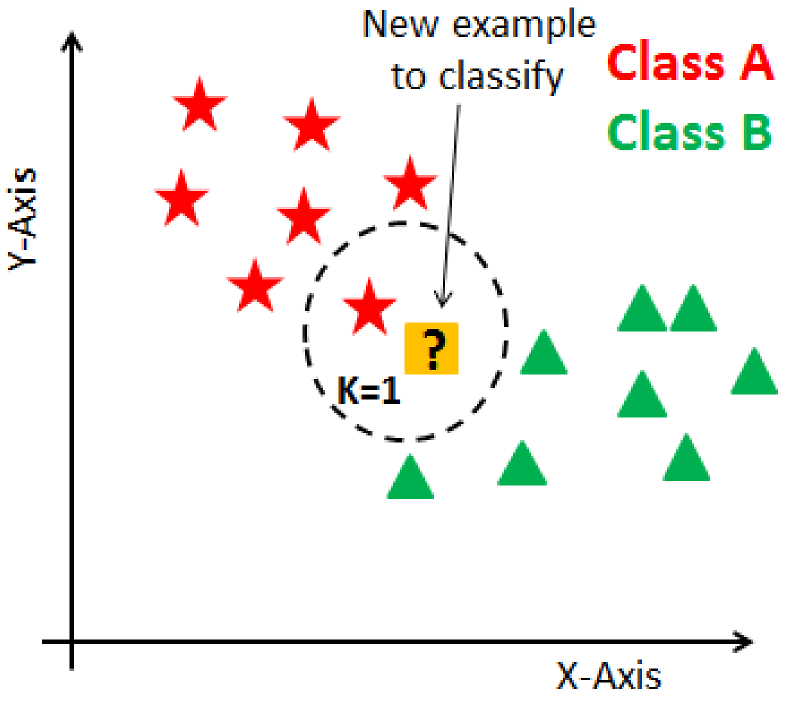

Figure 1:KNN Visualizing dada points.

In our model , K is the number of nearest neighbors. The number of neighbors is the core deciding factor. The image above visualizes how KNN works when trying to classify a data point based a given data set. It is compared to its nearest points and classified based on which points it is closest and most similar to. Here you can see the point Xj will be classified as either W1 (red) or W3 (green) based on its distance from each group of points.

4.6.2 Calculate Distances

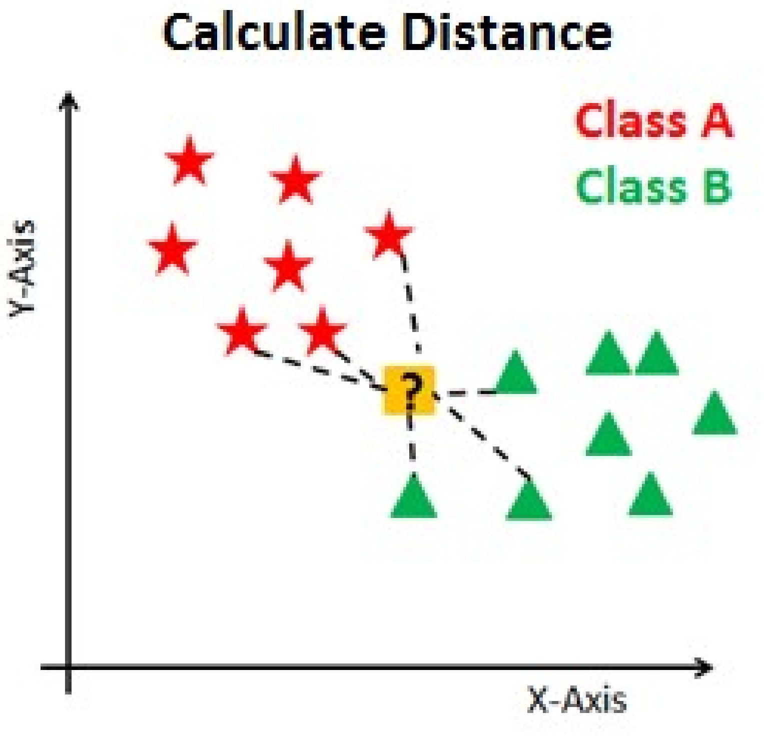

Figure 2:Calculating nearest distances.

In this phase, the model work on calculating the nearest point distances according to number of K. Calculate the distance between test data points and each row of training data, Using any of those methods : Euclidean, Manhattan or Hamming distance.

4.6.3 Calculate Distances

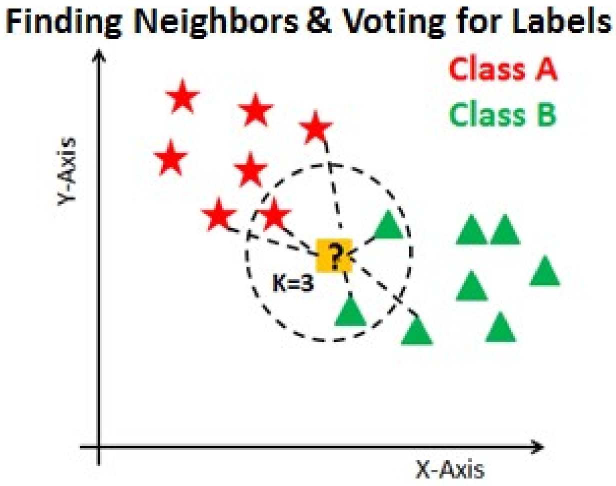

Figure 3:Vote on labels.

4.7 Machine Learning Model

Final phase at which the model is generated through suitable times of training for the model through different algorithms to predict the most accurate place for the new cell tower to be placed to support the network.

5 Human Interface Design

5.1 Overview of User Interface

Our system Cell Tower Placement user interface is very usable and clear. You can login whether you are an admin or user . The system navigates you to different screens depends on your type. System admins have some duties such as approving the the new pin on the map . While user can only pin an antenna then the admin approve it . In fact, illustration for the whole system will be shown the upcoming sections ().

5.2 Screen Images

5.3 Login Screen

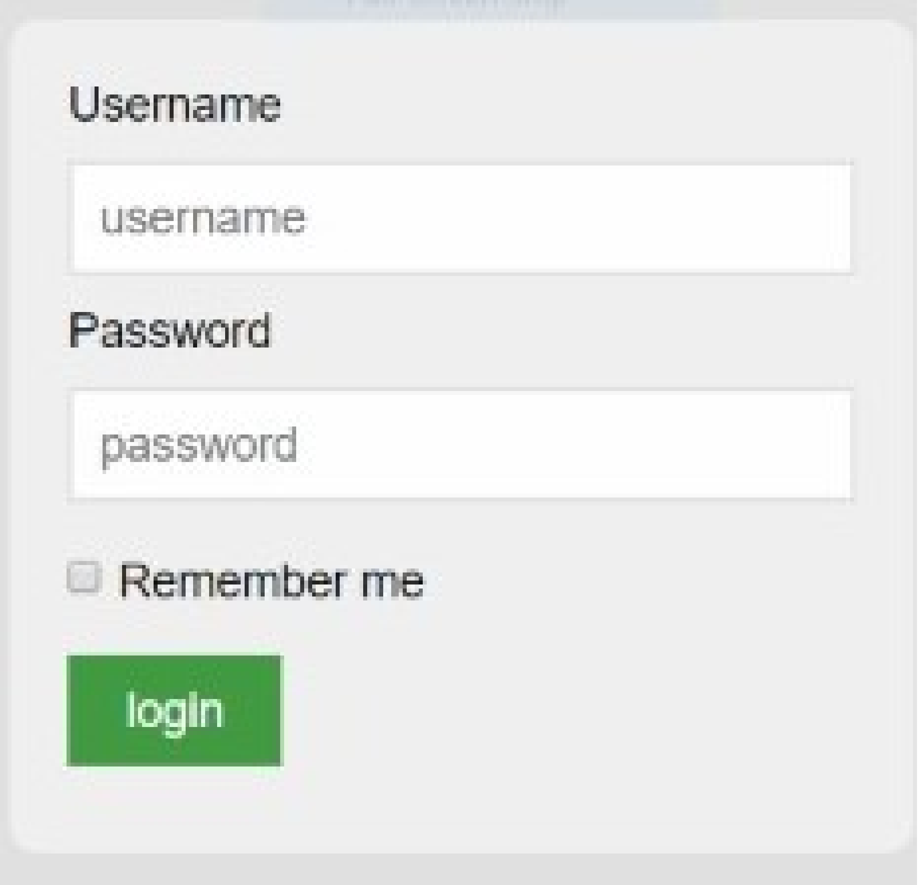

Figure 1 : Login Form

5.4 Home Screen

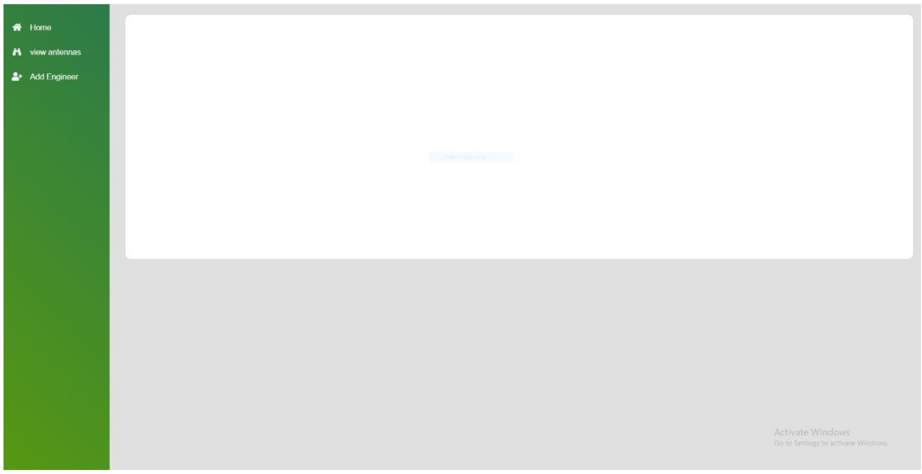

Figure 2 :Home Screen

5.5 Map Screen

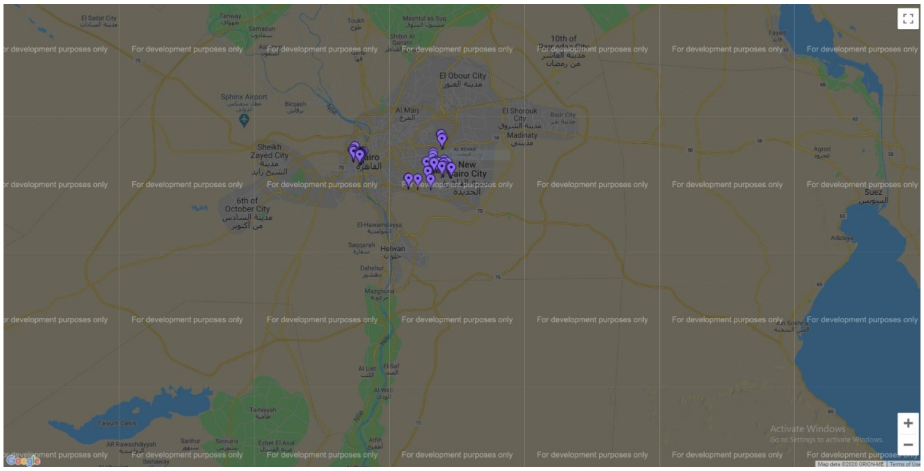

Figure 3 :Map Screen

5.6 Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

6 References

6.1

https://towardsdatascience.com/k-nearest-neighbors-knn-algorithm-bd375d14eec7