AQI Predictor

Final Year Project – Final Report

Session 2016-2020

A project submitted in partial fulfilment of the

COMSATS University Islamabad, Lahore Campus

Degree

of

BS in Computer Science (CUI)

Department of Computer Science

COMSATS University Islamabad, Lahore Campus

30 July 2020

Project Detail

Type (Nature of project)	[√] **D**evelopment [ ] **R**esearch [ ] **R&D**
Area of specialization	Artificial Intelligence

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	FA16-BCS-047	Waqas Ahmad	FA16-BCS-047@cuilahore.edu.pk	
(ii)	FA16-BCS-041	Hamza Sadiq	FA16-BCS-041@cuilahore.edu.pk	
(iii)	FA16-BCS-162	Ali Hassan	FA16-BCS-162@cuilahore.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I am Waqas Ahmad S/D/o Mukhtar Ahmad, group leader of FYP under registration no CIIT/FA16-BCS-041/LHR at Computer Science Department, COMSATS Institute of Information Technology, Lahore. I declare that my FYP proposal is checked by my supervisor and the similarity index is 12 % that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: ___30/07/20____ Name of Group Leader: Waqas Ahmad Signature:

Name of Supervisor: Dr Usama Ijaz Bajwa Co-Supervisor (if any): ________________

Designation Assistant Professor Designation: ____________________

Signature: ____________________ Signature: ____________________

HoD: ____________________

Signature: ______________________

Abstract

Our project, AQI Predictor, is an application to predict the quality of Air aka Air Quality Index (AQI) Level using images. The user can input the images by capturing directly from the smartphone’s camera or already captured ones from the gallery. The app uses a trained model that’s hosted on local server at the backend. The trained model is a Convolutional Neural Network that uses machine learning techniques to help us in predicting the Air Quality level (AQL) correctly. The model is trained on a relevant dataset in order to achieve the accuracy. Along with the app, a website will be created that will be synchronized with the app and the data will be collected. The data then will be used to generate a map of the areas the Air quality level has been calculated for. This will help people in deciding which areas have what kind of air quality. The quality of air will be predicted by the application providing us Air Quality Level which will further tell us more details such as if the air is harmful or good and how necessary precautions can be taken. These factors and results will help us and the authorities including the Government and relevant governmental and other bodies decide certain measures to make the air quality better.

Acknowledgement

Thanks to Allah Almighty who gave us the power and the courage to compete this project and it would be unfair and unkind not to acknowledge the guidance, intelligence and patience Dr. Usama Ijaz Bajwa has shared with us while working on this project. We are incredibly grateful to our supervisor for his vision and his persistence upon the values of hard work, helping others and not losing morale.

It’s been a beautiful journey and we couldn’t have wished for anything more than what we got.

We would also like to thank our institute, Comsats University Islamabad, Lahore Campus, for providing us a platform to work on as well as helping us in any way possible. This would not have been possible without it.

Our heartiest gratitude to Dr. Usama Ijaz Bajwa, colleagues and everybody who helped us in anyway with their thoughts, skills or any productive way.

Table of Contents

List of Tables

List of Figures

CHAPTER #1
INTRODUCTION

1 Introduction

1.1 Introduction

Air Quality is a major and serious issue in the World, which has been picking pace in getting identified in World's major cities in the past few years. A lot of International Organizations have started researches alongside independent scholars and research groups that have helped in identifying the problems pertaining to Air Quality and proposing small scale solutions as well as large scale solutions for the relevant problems.

Declining air quality index has been in progress for a few years, and major cities like Paris, Lahore, New Delhi, Beijing, and so many others have suffered from this. For a while, it wasn't acknowledged, but now it is, and it has been causing a lot of problems. Many International Organizations have worked to mitigate the issues arisen from the Air Quality Index (AQI). With all of the major problems, it takes a lot of time for the solutions, knowledge, and the problems to actually mature enough in order for it to work out.

It's only natural to consider and think of a solution that is portable, inexpensive, and easily implementable. A mobile app would certainly solve such issues and makes lives easier if it can predict the Air Quality level of a place just through a picture.

Now in Pakistan, there is no application or software available which can judge the Air Quality level of our environment and help us take specific measures. We decided to build an app that can judge the Air Quality Level of the local area just by a picture. It is not only frugal but also user-friendly as compared to hardware-based solutions. Most of the hardware devices which measure AQI need prior knowledge of the device to operate it and are expensive to purchase and operate.

The Air Quality level, AQL, is a term used to describe the categories the AQI is divided into. There are three levels that the AQI is divided into on the basis of PM2.5.

PM2.5 refers to atmospheric particulate matter (PM) that have a diameter of less than 2.5 micrometers. It is one of the major pollutants that authorities all over world monitor. PM2.5 values are used to determine Air Quality index.

Here is how AQI is divided on the basis of PM2.5 values:

1. Good: 0-74

2. Average: 75-115
3. Bad/Hazardous: >115

Our application predicts/projects the AQI of a place by categorizing it into one of three levels and then displaying the level to the user.

The front end of our main system (the app) is developed in Java using the Android SDK. It has a simple and clean interface that gives users the option to either capture a picture or use one from their gallery to process. The interface has all the features it requires and looks sophisticated. It works in conjunction with the model and the database that is linked to it at the backend. The app processes the picture to calculate the Air Quality level from the picture by handing the picture over to the model that's trained on the dataset we have given it. The data is saved in our database, increasing our dataset for the model to be trained further for higher accuracy if required. The app works on the principle of crowdsourcing. The users being the source for our constantly increasing dataset.

Coming onto the model, we trained the model using the concepts of Neural Networks. Our model extracted the features from the dataset and created feature vectors from the trained images. This set of features is used to train the model. The model trained itself on this vast dataset and then predicts the AQ level. It returns an integer value that tells us which level it is:

- 0 is average
- 1 is bad/hazardous
- 2 is good.

By getting the user's location, we can also tell the temperature as well as humidity level of that specific location by obtaining information from another source.

Secondly, we created a website that can be used to predict AQL using the same model that the app uses. The user can simply upload a picture and the website will predict the AQL and display it to the user. It works in synchronization with all the data that the database has at the backend. The site is linked to our database. It will show all the data and results that will be calculated by the app, which can be useful for the average citizen or relevant government institutes/employees. The website also embeds a map which will show the AQL of different areas that have been predicted by the app.

This accomplishment will help the Government to tackle the areas and alleviate the problems of bad air quality by doing proper plantation or initiating a drive for plantation. Govt. can also hinder the working of factories which emit harmful chemicals in that area until air quality reaches a normal level.

1.2 Objectives

There are many countries that have been affected by the poor quality of air that is the result of many external factors such as poor waste management, burning of various materials that harm the air and excessive smoke produced by vehicles and factories, etc. There are no cheap solutions as to measuring and preserving the air quality.

Our main objective was to create an android app that will measure the Air Quality level just by taking a picture. A pre-trained model is used at the backend server which will help the app in determining the air quality level.

Our project will:

- Use the phone’s camera to take pictures to predict AQL of
- Use the predicted AQL to give suggestions to the user about what precautions might be necessary
- Pin the AQL on the map for other users to see
- Provide real time AQL value of any place by a picture
- Be extremely useful for the authorities to check AQL of certain areas where sensors are unavailable or unfeasible.

1.3 Problem Statement

An AQI Predictor System is an android and web based application that predicts Air Quality Level. It is developed to make the entire process of finding air quality of any place easier, accessible, and affordable. This system works by using a trained deep learning model that predicts Air Quality Level by using the images that the user inputs through the application.

1.4 Assumptions and constraints

The assumptions and constraints of the system are as follows:

1.4.1 Assumptions

- The user must have an Android smartphone that can run the app.
- The user’s smartphone must have a functioning camera.
- The user must have an internet connection to use our system (application or website).

1.4.2 Constraints

- Our Android application will run on Android version Android 6.0.1 or higher.
- The photo that the user will use to calculate Air Quality level must have at least 40% sky in it.
- It needs daylight and it will not work during the night.
- It requires an active internet connection to function.

1.5 Project Scope

Hazardous AQI is a serious issue today especially in Pakistan. The Government and other relevant authorities haven’t been able to manage to bring down the AQI of major cities to a good level. Our system will help the Government and other relevant authorities in checking the Air Quality Level of any certain area in the picture to help them decide how the Air quality of the place is and what necessary measures can be taken in order to control the Air Quality. The information and dataset created by our system can be used by the Government of Pakistan and relevant bodies for further research and studies to reach conclusions.

CHAPTER #2
REQUIREMENT ANALYSIS

2 Requirement Analysis

2.1 Literature Review

In Pakistan there is no such mobile or web application that tells its user AQI by just clicking a photo. Here are some related works from around the globe:

2.1.1 Air Visual

Air Visual [1] is a mobile application as well as website that takes PM2.5 values from different sources. It displays to the user PM2.5 value and AQI level of a specific city. It is a platform that only displays values taken through sources rather than an AQI calculating application. These sources are sensors that help in determining PM2.5 values from the specific areas. Air Visual also sells their products in the form of AQI calculation sensors and Air Purifiers.

Screenshots of the application are provided below:

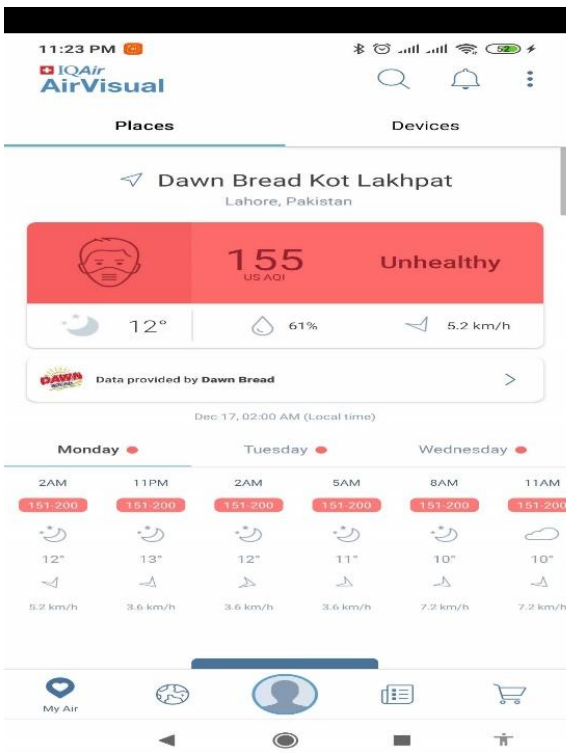

Figure 1 – Air Visual – Displaying AQI

Figure 1 shows the list of cities ranked in order of their Air Quality Index from worse to better.

2.1.2 Air Cognizer

Air Cognizer [2] is a mobile application which predicts AQI. It is limited to Indian region because it was developed by Indian developers and students. It's not available in Pakistan. It also takes a picture as input through user's phone camera or gallery and then uses an embedded model in order to determine AQI and display it to the user. It uses similar methodology with our application.

Screenshot of its main page: below:

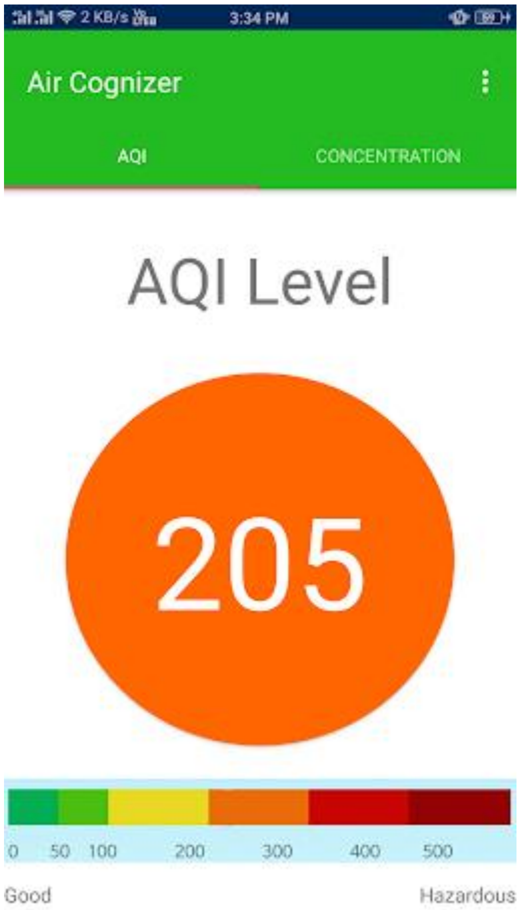

Figure 2 – Air Cognizer app

Figure 2 shows that the Air Cognizer app displays the AQI of the place in the picture.

2.1.3 Research papers

There are a few research papers available on the internet which are relevant to our project.

2.1.3.1 Air pollution forecasting from sky images with shallow and deep classifiers

This research paper [3] talks about how to extract information from images that can be used to detect or predict Air Quality Index of the location/atmosphere in that image. In this paper, a method was proposed that predicted the level of air pollution of any location by using an image that was taken from a smartphone's camera. At the end there were two classifications methods that were implemented in order to gather air pollution information from an image. The second method uses a Convolutional Neural Network (CNN) which is designed to get an image that has sky in it and provides the results in form of level of air pollution. There were some evaluation tests conducted and acceptable accuracy was found through these methods.

2.1.3.2 Air Quality Measurement Based on Double-Channel Convolutional Neural Network Ensemble learning

This research paper [4] proposes a solution to solve the single channelled convolutional neural networks that are used to train images. While the Single channelled CNNs take one feature at a time, the double channel CNNs can take the combined feature extracted from different parts of an image. The "double-channel weighted convolutional network ensemble learning algorithm" was proposed in this paper. In this double channel CNN, each channel is used to train different parts of the image to extract different features. And afterwards, there's a feature weights self-learning method proposed which weighs and combines the extracted feature vectors to calculate the AQI.

2.1.3.3 Image-based Air Quality Analysis Using Deep Convolutional Neural Network

This research paper [5] discusses how to classify and categorize images into different levels based on their PM2.5 concentrations. The proposed method tells how to implement this in model and get results based on different categories of PM2.5 values which can be used to determine AQI.

2.2 Stakeholders list

The stakeholders list of our system is as follows:

2.2.1 Admin

The admin will be the top level manager who will manage the whole system, any issues and the data that comes in and goes out.

2.2.2 Users

The system will help users to predict the Air Quality level using pictures and helping them in suggesting measures.

2.2.3 Government of Pakistan and relevant authorities

This system will help the Government of Pakistan and the Climate Change ministry in checking the Air Quality level of places and then taking certain measures. Currently they have to use sensors for AQI measurements which are both expensive and rarely available.

2.2.4 Any private organization or students

Lastly, any other organizations, students or researchers who want to use our application and data for further research and studying can benefit from our system.

2.3 Requirements elicitation

The requirements of the system from the stakeholders' point of view are as follows:

2.3.1 Functional Requirements

The functional requirements of the system are mentioned below:

2.3.1.1 FR01 – Predicting the AQL

Table 1 - FR01 - Calculating AQL

FR01	The application will take the picture through the camera app on the user's android phone or through the gallery, then it will predict its AQL.

Table 1 shows as the app takes a picture through user's input (camera or gallery), the app will calculate the Air Quality level of the picture.

2.3.1.2 FR02 – Showing the AQI and information to the User

Table 2 - FR02 - Showing the AQL and information to the User

FR02	Using the trained model embedded at the backend server, the app sends the image to the server and the Air Quality level is predicted. Then the Air Quality level is shown alongside any other information or suggestion to the user.

Table 2 shows that the Air Quality level has been calculated, then the Air Quality level and additional information are shown to the user on their mobile's screen.

2.3.1.3 FR03 – Checking out data and information

Table 3 - FR03 - Checking out data and information

FR04	The user can view and use the information that will be uploaded on the server through the website and through app as well.

Table 3 shows that after the data is uploaded on the server, the user can view it any time they want through the website and the app.

2.3.2 Non-functional requirements

2.3.2.1 NFR01 – User Friendly

Table 4 - NFR01 - User Friendly

NFR 01	The GUI of the application and website will be very user friendly.

Table 4 shows that our app and other system will be user friendly and very easy to understand.

2.3.2.2 NFR02 – Simple and minimalistic design

Table 5 - NFR03 - Simple and minimalistic design

NFR 02	The design of the application will be simple and minimalistic making it extremely easy to check out all the options and use them.

Table 5 shows that our app will have an ultra-simple and minimalistic design which will help in transparency and understanding.

2.3.2.3 NFR03 – Portability

Table 6 - NFR03 - Portability

NFR 03	The application will be usable on any device with an Android version of 6.0.1 or higher.

Table 6 shows that our app will be usable on any android device which has an android version of 6.0.1 Marshmallow or higher.

2.3.2.4 NFR04 – Documentations and help

Table 7 - NFR04 - Documentations and help

NFR 04	A complete documentation and how to use of the application will be provided.

Table 7 shows that there will be a complete documentation about the application and the system which shall be provided.

2.3.2.5 NFR05 – Responsive and fast

Table 8 - NFR05 - Responsive and fast

NFR 05	The prediction of the Air Quality level will be done within seconds, even on low end devices. The application will be quick to generate results.

Table 8 shows that our app will be highly responsive and quick to generate results as well as showing them to the user. There'll be no unnecessary waiting.

2.3.2.6 NFR06 – Accuracy

Table 9 - NFR06 - Accuracy

NFR 06	The application will generate results that will be reasonably accurate.

Table 9 shows that our app will generate results that are accurate up to a very good percentage so that the user can trust the results and the app.

2.4 Use case descriptions and designs

2.4.1 App Side

Table 10-Use Case-App Side

Use case ID: 01
Use case name: AQI Predictor app
Actors: User, Server
Use Case Summary: The Air Quality level is determined by the user's input image through the device's camera. The Air Quality level will be displayed on the app.
Pre-condition: Must have app installed, must have android phone with android version 6.0.1 or higher
Course of events: The user captures a picture, the app will send the picture to the server and the server will send predicted Air Quality level from the picture as an output. This AQI level will be displayed by the app.
Post condition: Air Quality level is determined by the model and displayed on the app.
Includes: Information, settings, take picture, predict Air Quality level.
Extends: No GPS Permission, no internet connection.

Table 10 shows us that the use case "AQI Predictor app" with the use case ID "01" explains the workflow of the application side of the project. This application side is the one that the user is going to deal with. There are two actors in this use case, the user and the server. The user performs majority of the tasks here including taking a picture. Then the picture is run through the app to the model which then predicts the Air Quality level and shows it back to the user on the screen.

2.4.2 Design

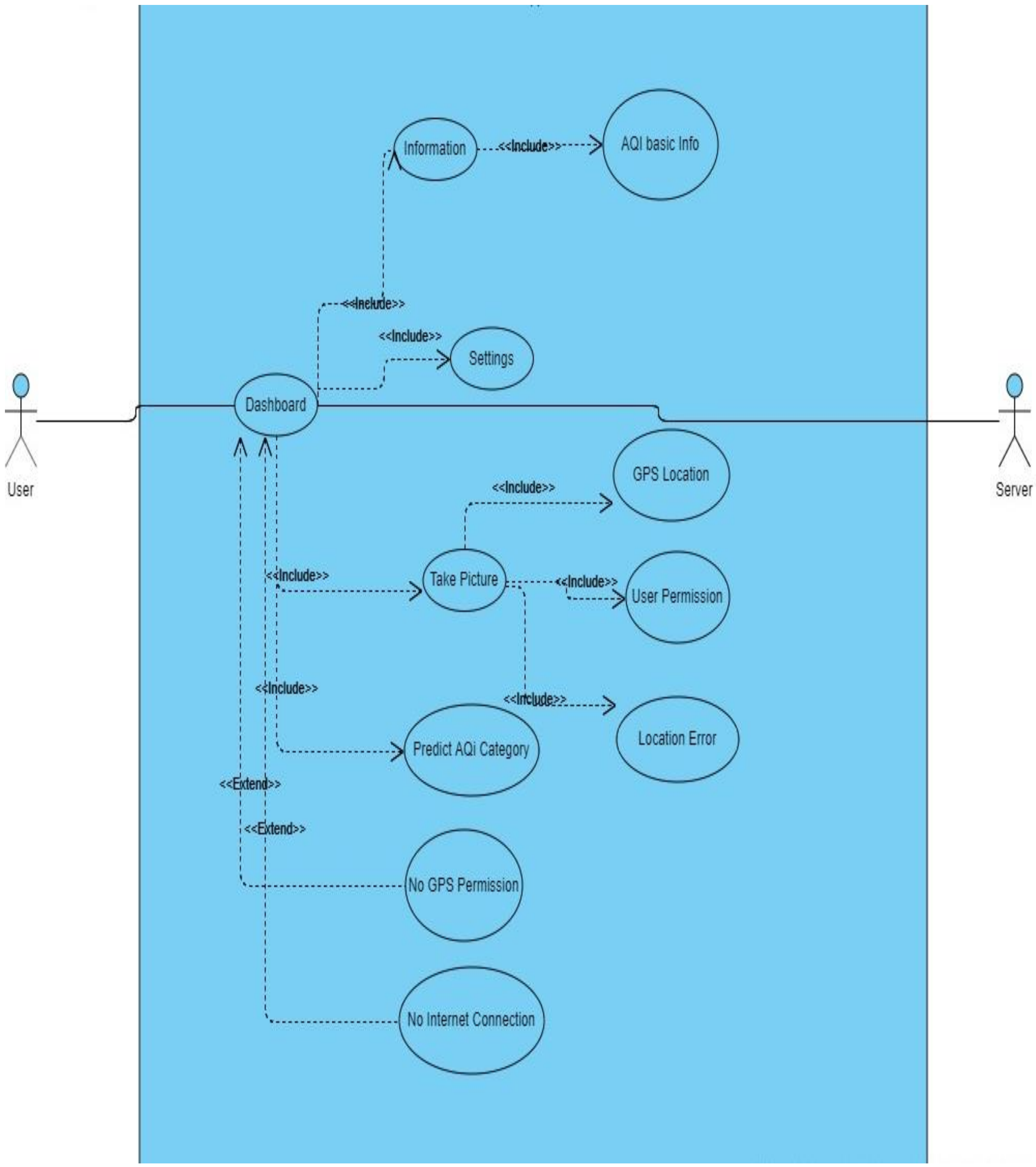

Figure 3 – Use case: App Side

Figure 3 shows that the actors in the use case are the user and the server. The user is the main actor here who performs the tasks while the server is connected to the backend. The user takes a picture, the picture goes through the model and then the Air Quality level is predicted. The Air Quality level is then shown to the user on the screen. If there's no internet connection, the application will not run. Then the data is uploaded on the server.

2.4.3 Server side description

Table 11-Use Case-Server Side

Use case ID: 02
Use case name: AQI Predictor Server
Actors: Admin, App, User, Database
Use Case Summary: Admin will have access to modify and add into the dataset, the model and the application at any time to make any changes or increase efficiency.
Pre-condition: The admin must have access to the internet/server in order to perform the tasks.
Course of events: Admin opens the console, goes into the server and increases the images in the dataset. Then the admin trains the model through the increased images which results in increased efficiency. The admin can also update the application through this.
Post condition: None
Includes: Connection, training, server, updation, result verdict.
Extends: Location error, internet error

Table 11 shows us that the use case with ID 02 titled "AQI Predictor server" explains how everything works at the server end of the application. The dataset can be increased, the model can be retrained and redeployed and the app can be updated through the server side.

2.4.4 Server side design

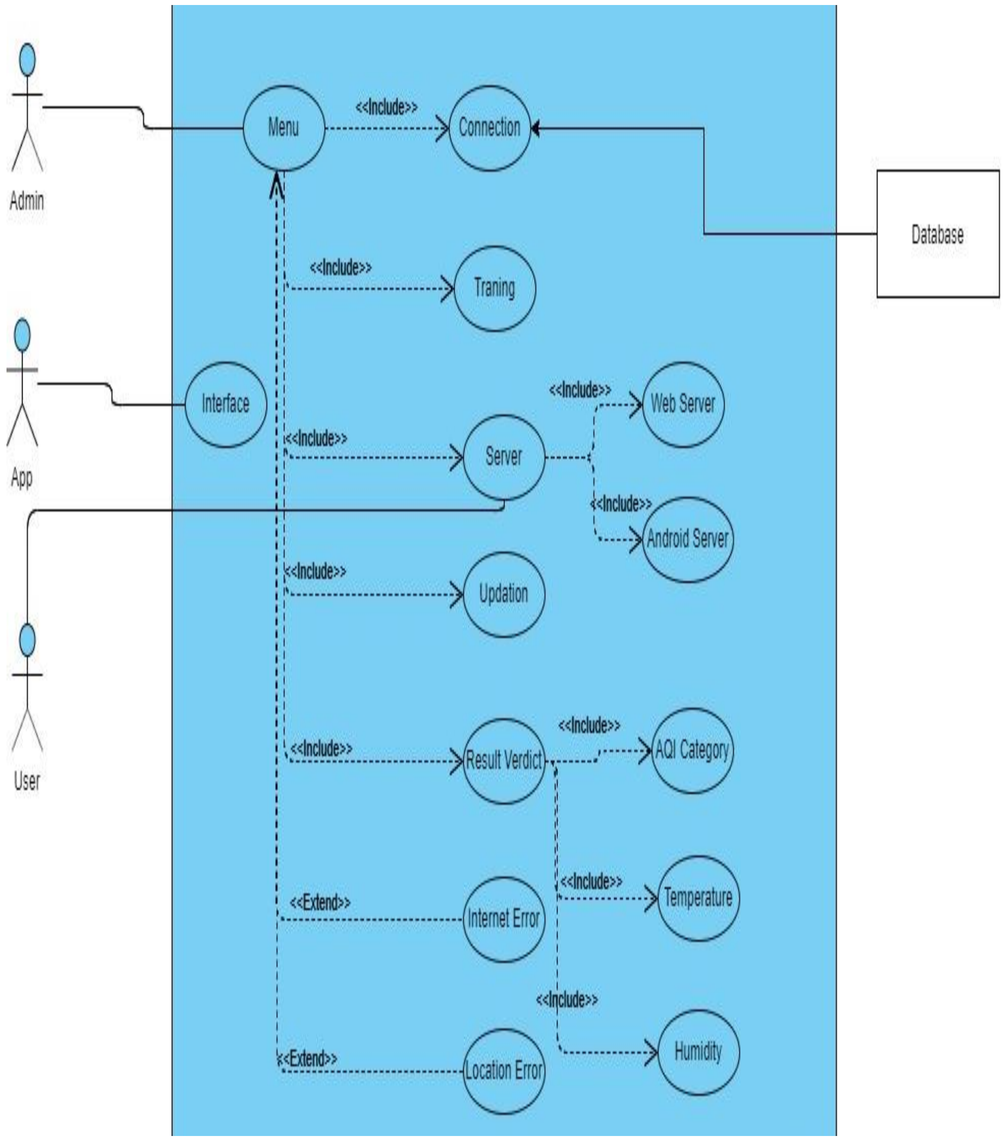

Figure 4 - Use case: Server side

Figure 4 shows us the use case of Server Side. There are three actors in this use case:

1. Admin
2. App
3. User

This figure explains us that the main actor is the admin. It explains to us the working and how the admin can work through the server side. The admin can manage things such as managing the dataset, increasing the dataset, retraining the model and updating the app. All of this is done in connection with the database.

2.5 Software development life cycle model

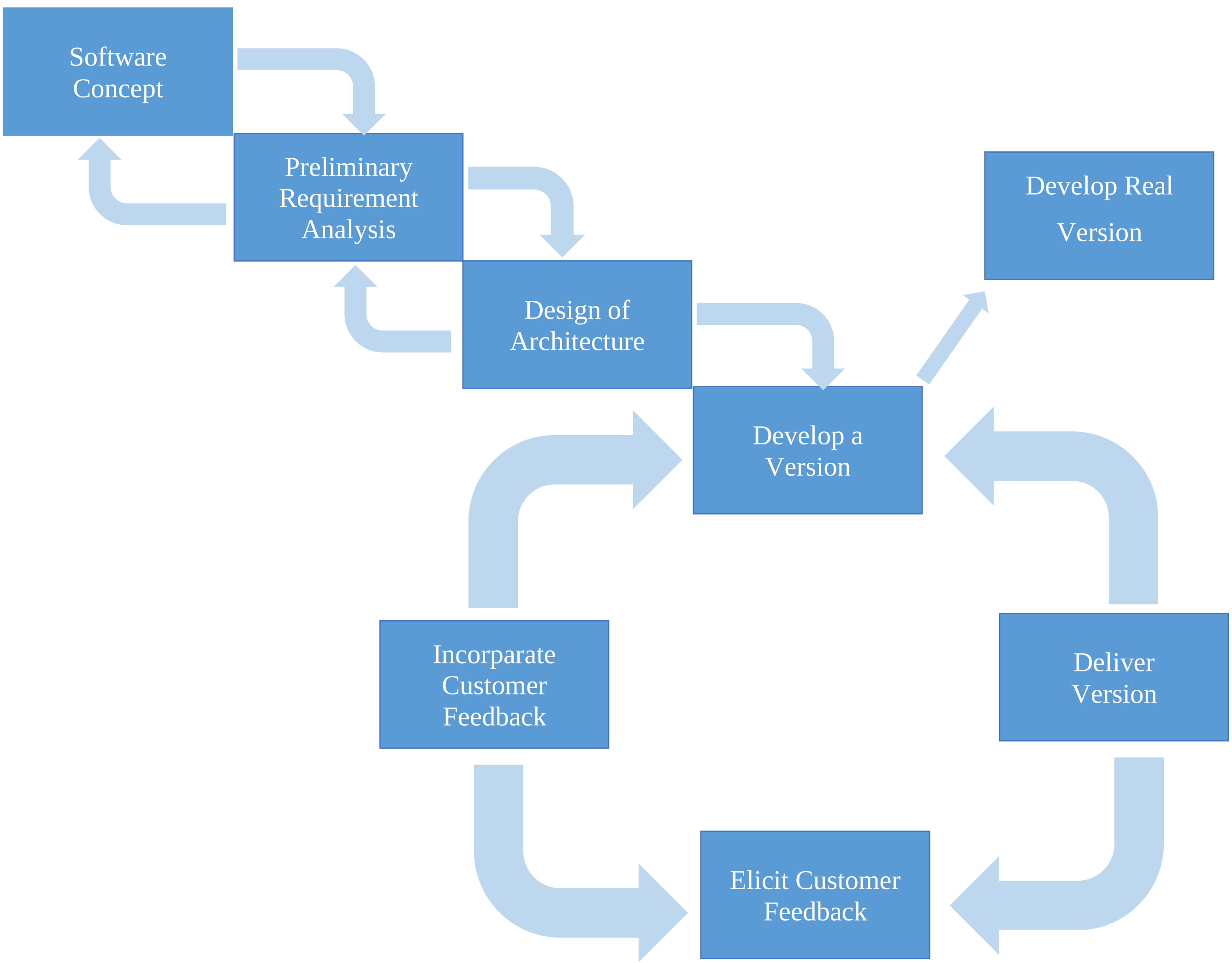

Figure 5 – SDLC Model

Figure 5 shows the Software Development Life Cycle Model we have chosen for our project which is the Evolutionary model. The reason behind choosing this model is that we need to redefine our features and functions as we keep developing. Our project is based on modules which are completed in sequence and then integrated into the whole project. The evolutionary model combines iterative and incremental model. We have divided our project into smaller, incremental developments. Then those small developments were tested and checked if there could be any issues. Then we developed on top of these small developments which in the end gave us our completed project with updated and latest features and functions as well as efficiency while addressing our issues along the way. We updated our project as we gained more knowledge on how this project could be made better.

CHAPTER #3
SYSTEM DESIGN

3 System Design

Chapter 3 shows all the designs that were created of this project:

3.1 Work breakdown structure

The work breakdown structure is as follows:

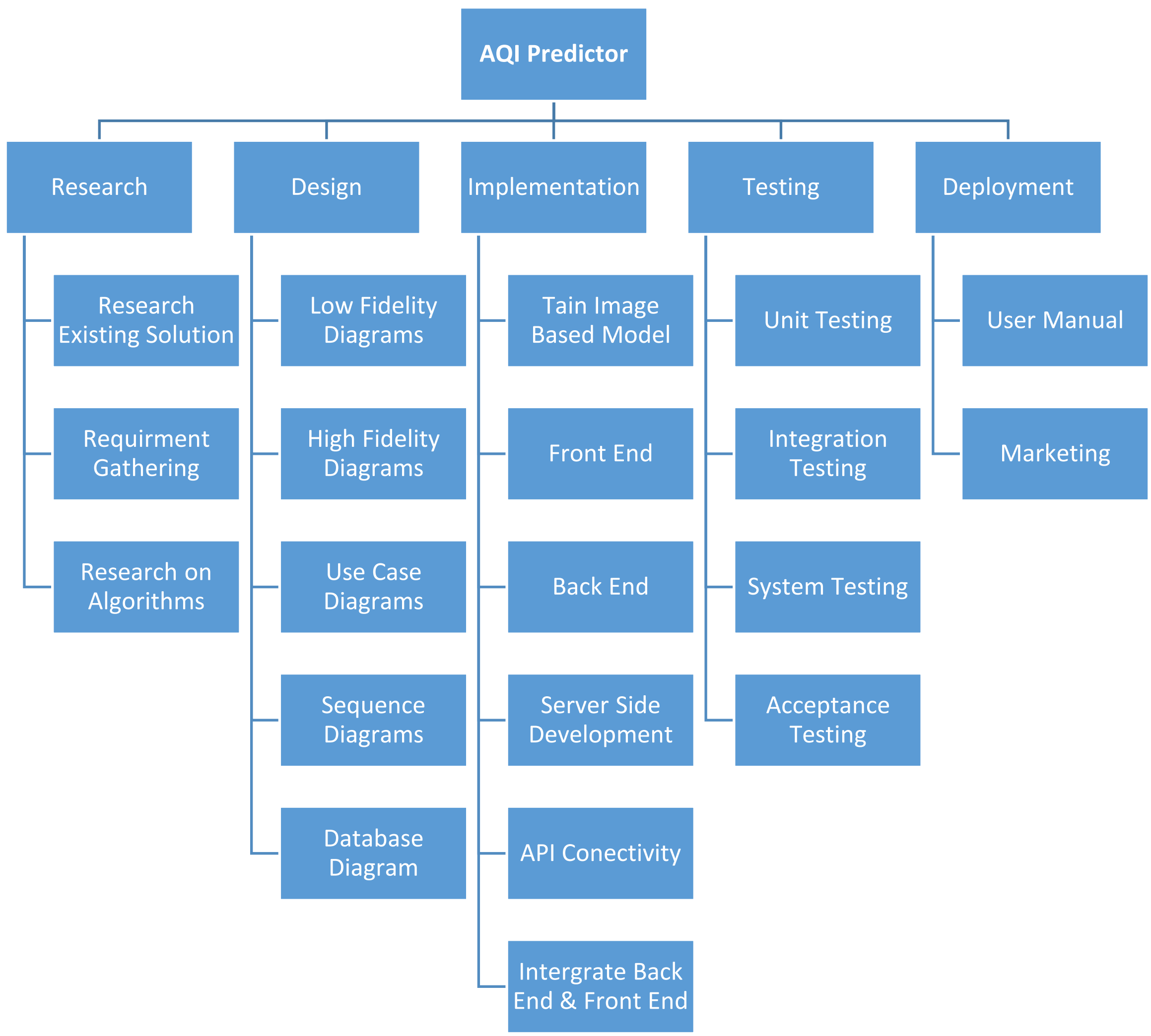

Figure 6 - Work BreakDown Structure Diagram

Figure 6 shows us the work breakdown structure of the whole project starting from research till deployment. It explains all the working of the whole project.

3.2 Sequence Diagram

Sequence diagrams explains the whole working of the system.

3.2.1 AQL Prediction from Mobile App:

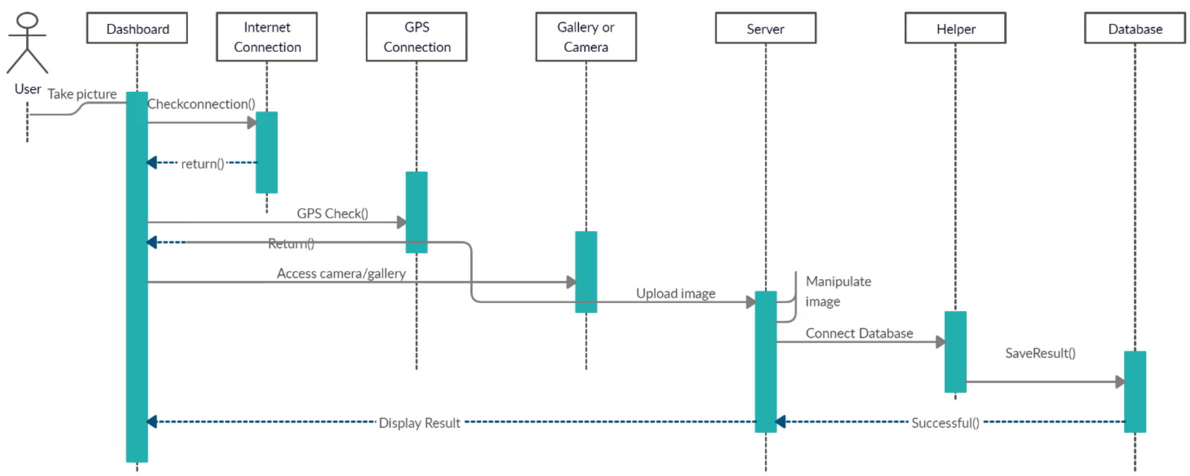

Figure 7 - Sequence diagram – Taking Picture

Figure 7 explains the workflow of the entire sequence of how a user takes a picture through the camera of his/her phone, then the picture is sent with the user's location to the model in the app, then the results are calculated and shown back to the user. After all that, if the user's device is connected with the internet, it connects with the database at the server end and stores data there.

3.2.2 Training Sequence Diagram

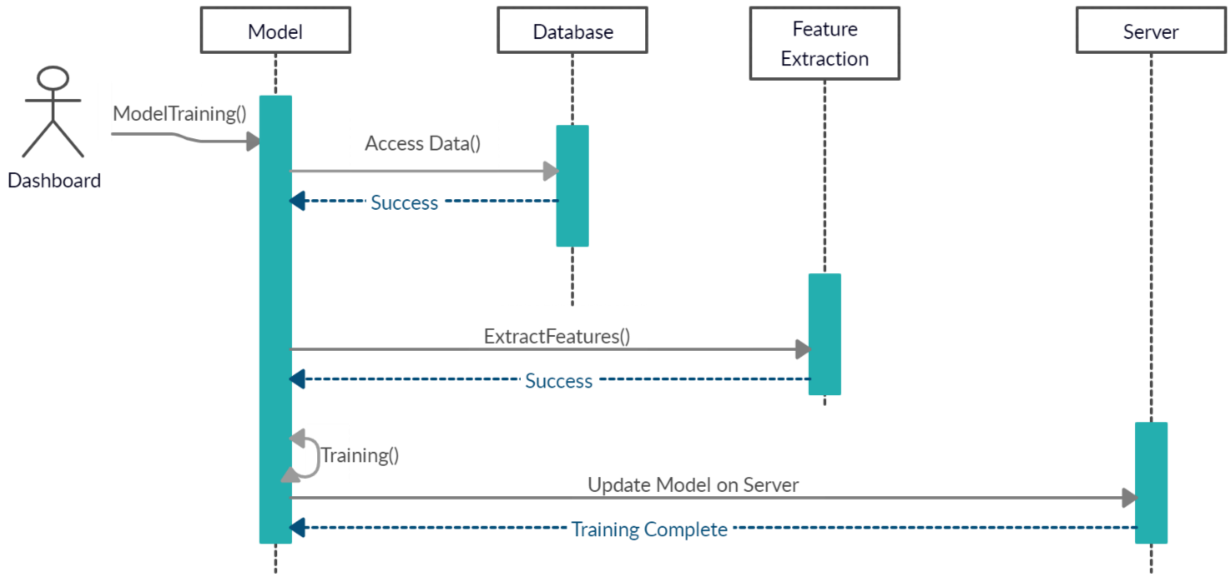

Figure 8 - Sequence Diagram – Training

Figure 8 shows us the training sequence of the model. Only the admin can initiate training or retraining of the model embedded in the app. When the admin initiates the training sequence, the model is connected with the database, the data is picked up and fed to the model and then the training is performed. After the training is done, the model file is updated and embedded in the application and the training is complete.

3.2.3 AQL Prection from Web App

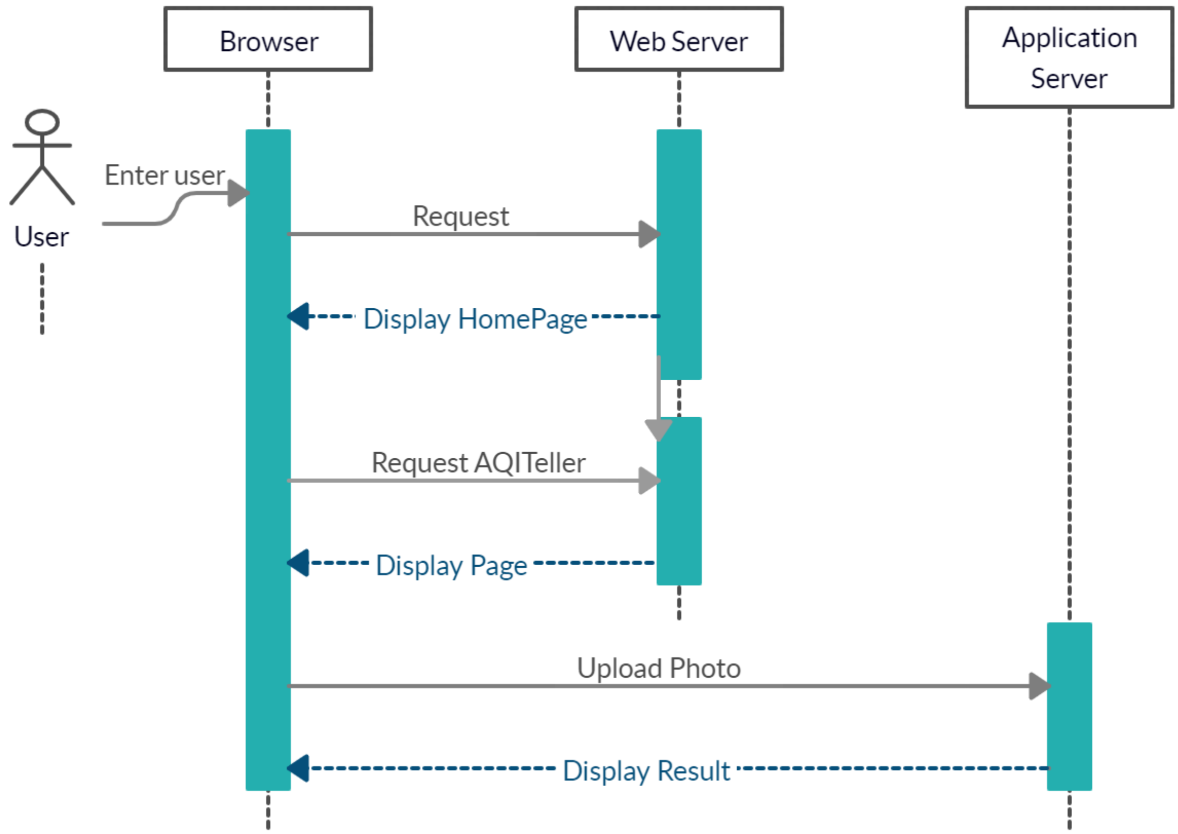

Figure 9 - Sequence Diagram - Web Server

Figure 9 shows us the Web Server Sequence Diagram which explains the workflow of the web side of the project. When the user opens the website through the designated URL, the browser requests for access for the page and the data that is available on the database. Upon this, the results are generated from the database and it generates dynamic web pages that are displayed to the user in their browsers.

3.2.4 AQL Map

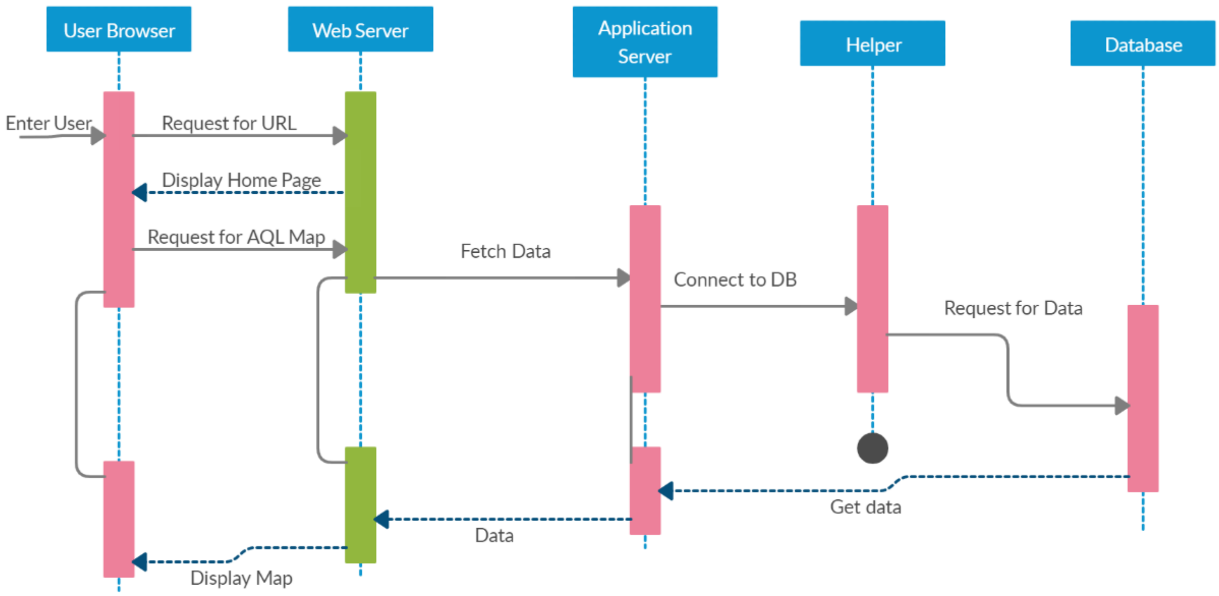

Figure 10 - Sequence Diagram – AQL Map

Figure 10 shows us the AQL Map Sequence Diagram which explains the workflow of the AQL Map of the project which is present both in the android application and on the website. When the user opens the website through the designated URL or the android application, the application (in case of app) or the browser (in case of website) requests for access for the page and the data that is available on the database. Upon this, the results are generated from the database and it generates AQL map that is displayed to the user in their browser or the application.

3.2.5 Web Pages

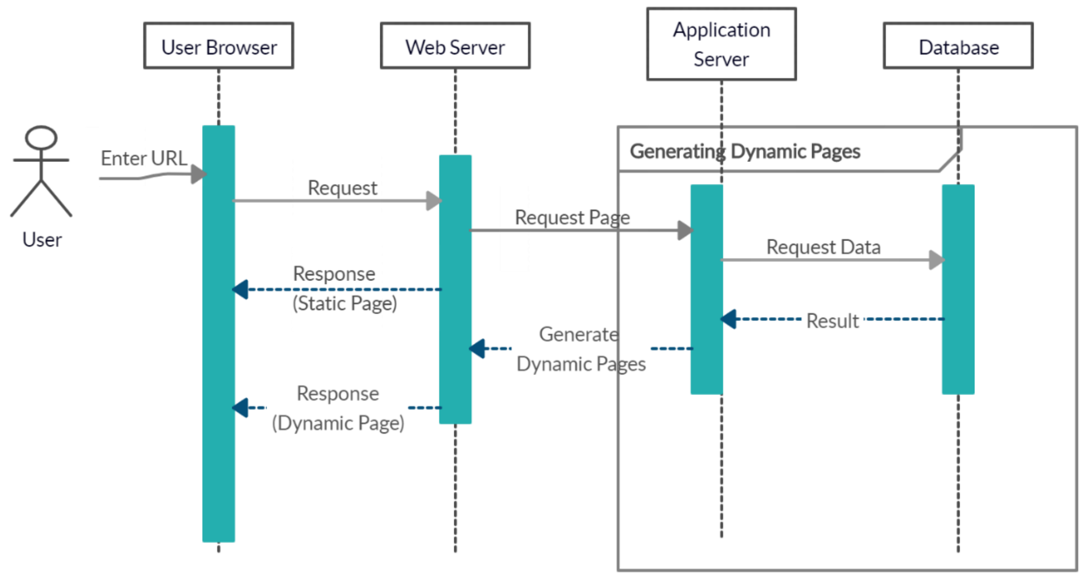

Figure 11 - Sequence Diagram - Web Pages

Figure 11 shows us the Web Pages Sequence Diagram which explains the workflow of the website of the project. When the user types in the URL in the browser, there's a request at the web server which provides response in terms of static pages. Our website is a composition of static and dynamic web pages and this sequence diagram explains how both of these work. In case of dynamic pages, the web server requests page from the application server which then requests data from the database. The database then returns the request with data, which the application server uses to generate dynamic web pages and send back to web server. The web server then displays it to the user in the browser.

3.3 High Fidelity Diagrams of the System

The high fidelity diagrams of the whole project are as follows:

3.3.1 Application:

3.3.1.1 Guide Page

Figure 12 - Guide Page

Figure 12 shows us the welcome page of our application which is our guidelines for our users to know. The welcome screen shows all the guidelines for our users.

3.3.1.2 The main page of the application:

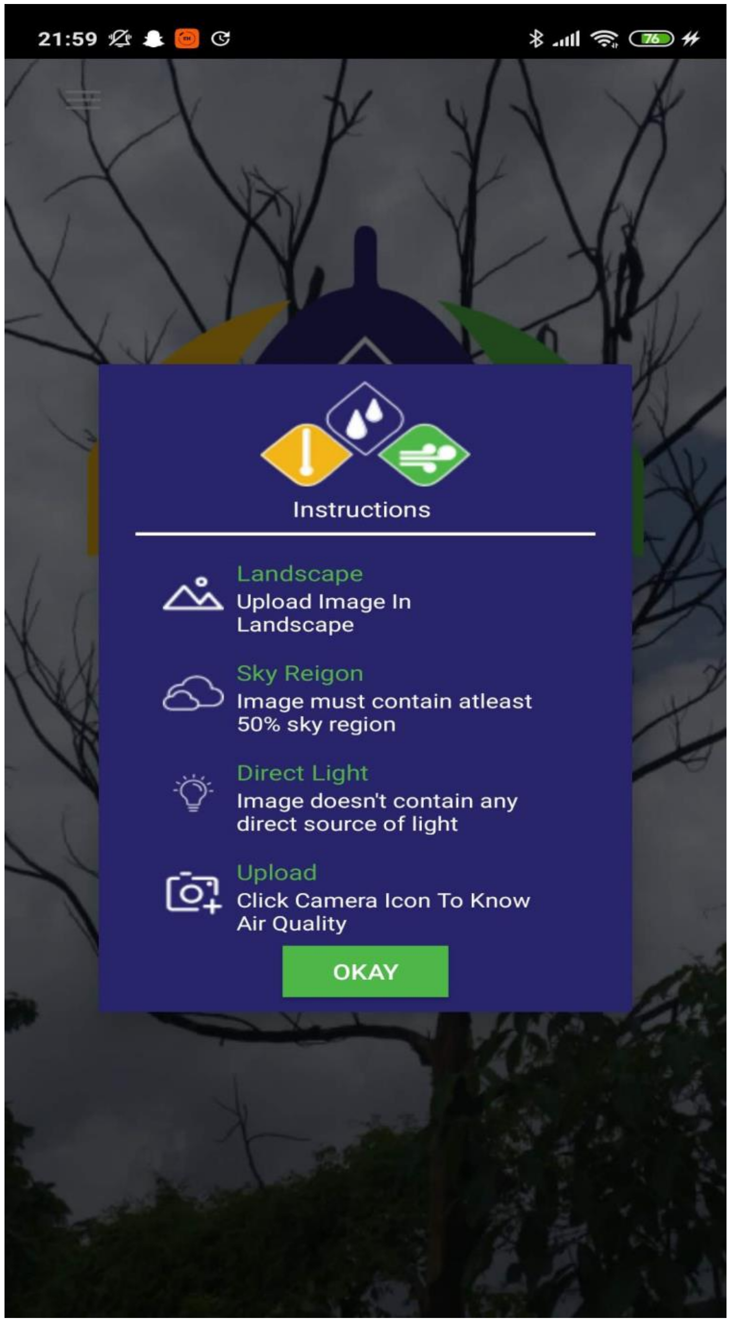

Figure 13 - Welcome Page

Figure 13 shows us the welcome screen of our application. The welcome page shows the instructions about how to use the application.

Figure 14 - Home Page

After the welcome screen, it directly takes you the home screen. Figure 14 shows us the home screen which is straightforward and simple. It gives you the option of taking a picture and predicting AQL which the is the main purpose of the system. Or you can swipe left to see more options.

3.3.1.3 Side Drawer/Navigation Drawer Page

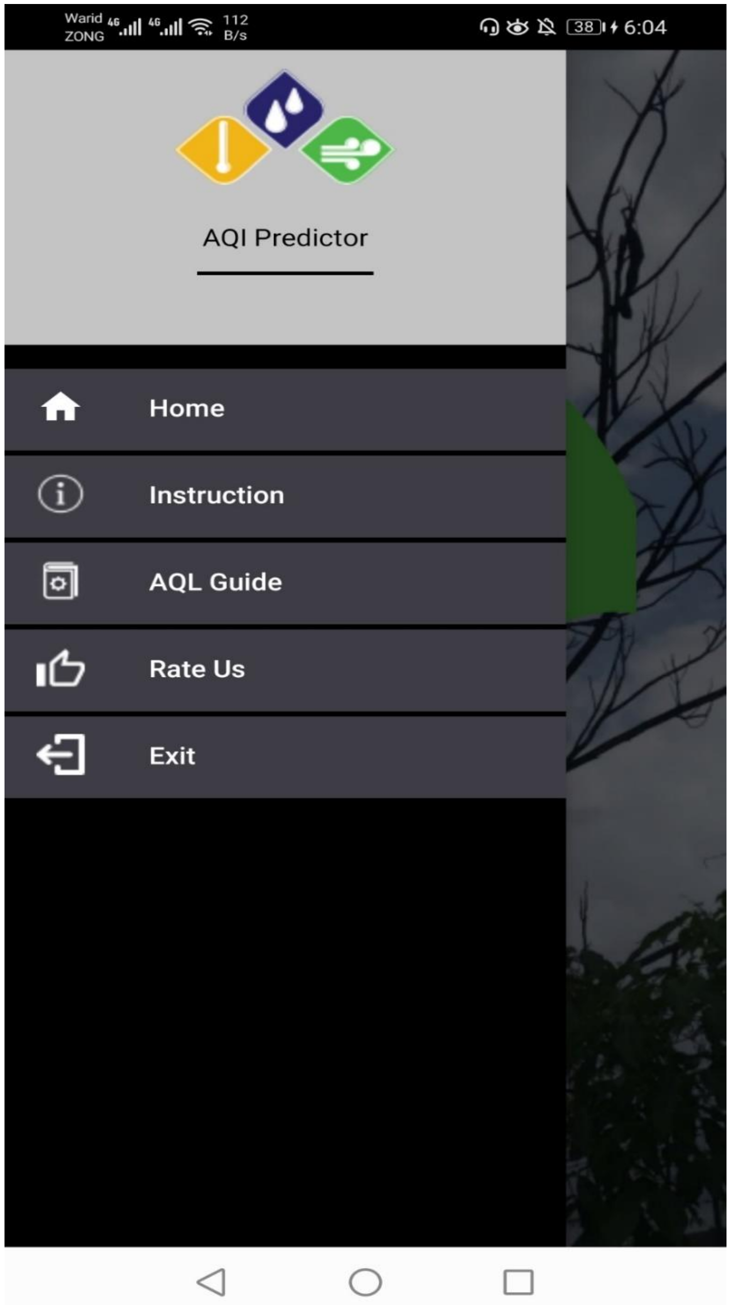

Figure 15 - Side Drawer/Navigation Drawer

Figure 15 shows the navigation drawer which can be accessed by swiping left on the home screen. It gives access to further options.

3.3.1.4 Results Page

Figure 16 - Results Page of the application

Figure 17 - Results Page with AQL Map

Figure 16 and 17 show the results page of the application. When the results are generated and shown the user in terms of Air Quality level along with some additional information such as health recommendations, humidity level, maximum and minimum temperature which can be seen in the interface. It also shows the AQL map that is built with the data our system has concerning AQL of different places.

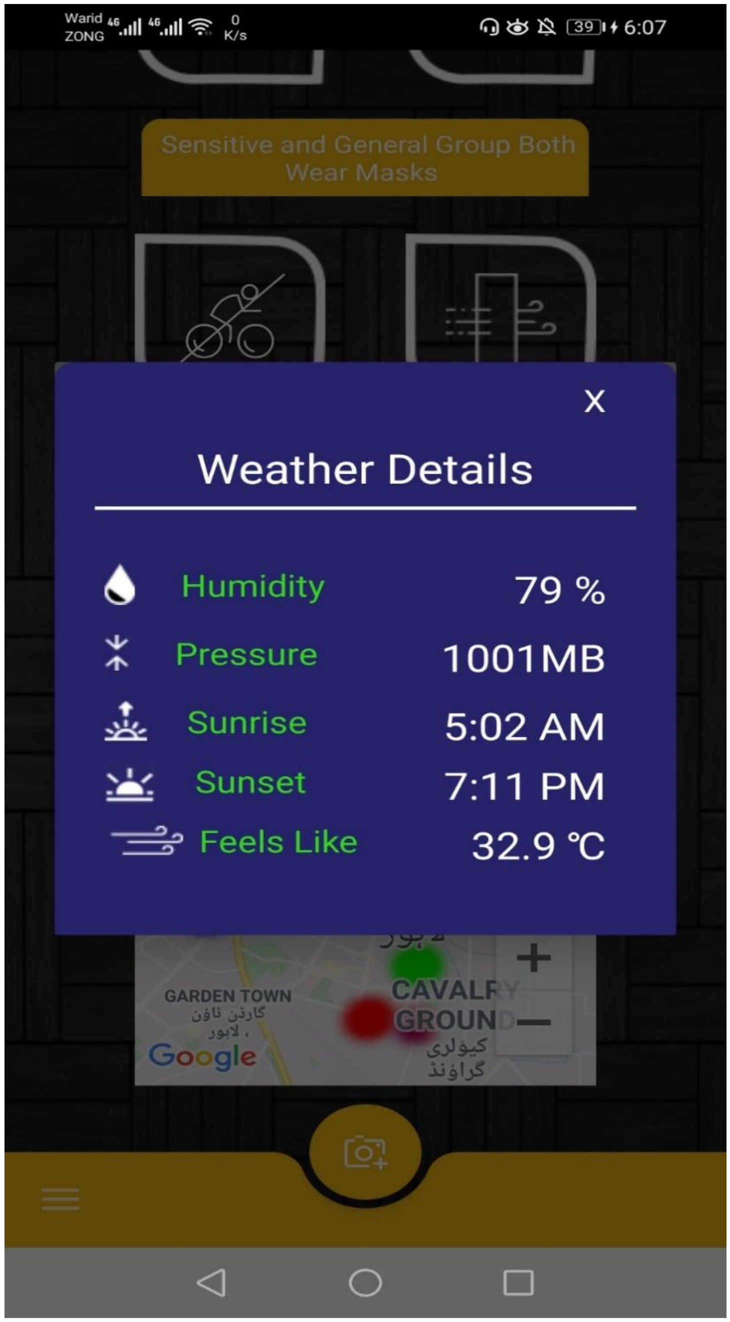

Figure 18 - Results Page - Weather details

Additionally, figure 18 shows us the weather details which can be accessed by tapping on the weather. This page shows us extra details about the weather.

3.3.2 Website

3.3.2.1 Home Page of website

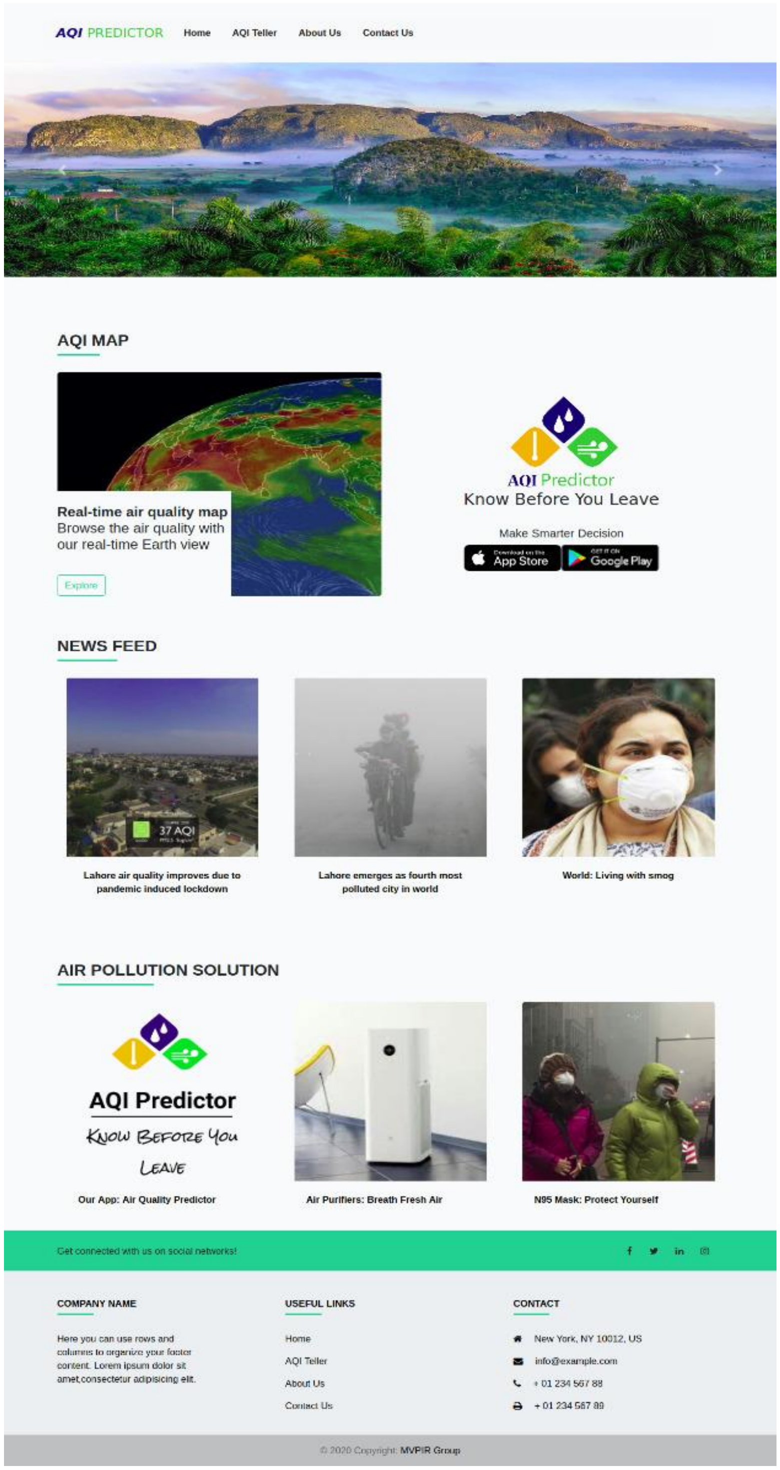

Figure 19 – Website Home Page

Figures 19 show the home page of the initial website designed for the project. The home page houses the header, the footer and the following features:

1. AQL Map which shows the AQL marked of the cities on the map using Google Maps API
2. A news feed that houses relevant news stories
3. The solutions to Air Pollution

3.3.2.2 AQL Map

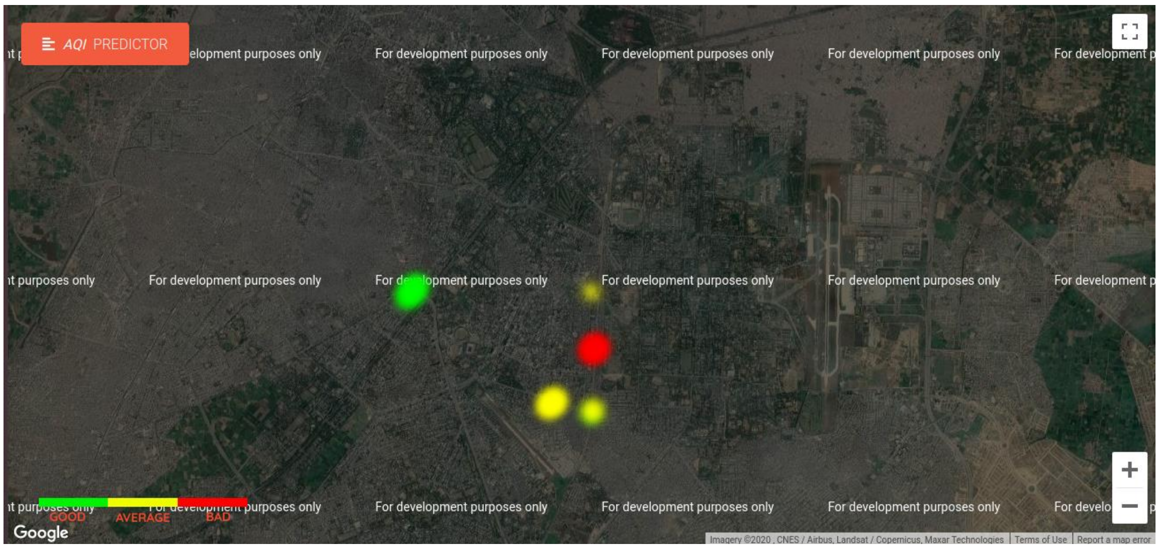

Figure 20 - AQL Map on Website

Figure 20 shows the AQL Map on the website.

3.3.2.3 AQI Teller Page

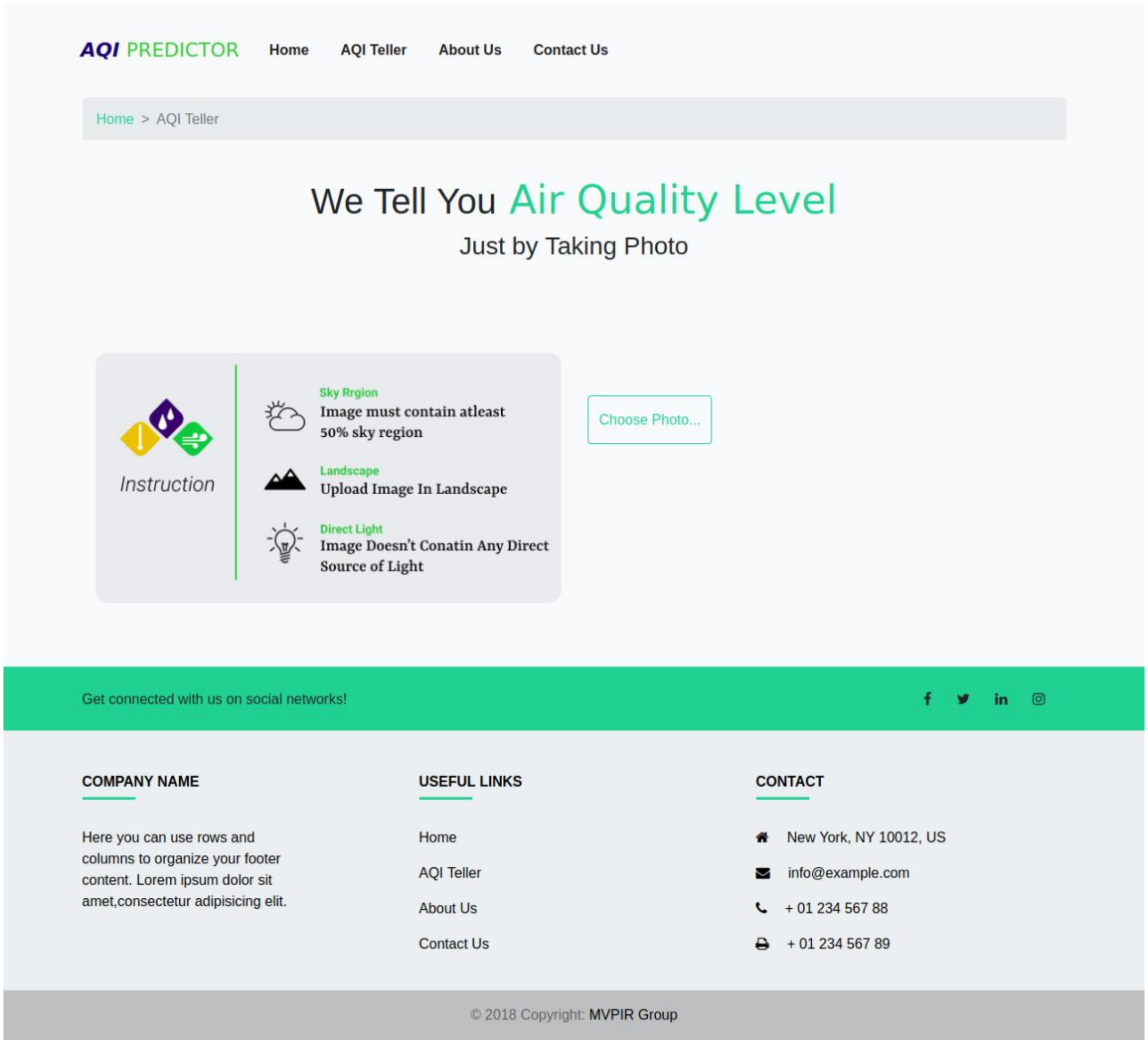

Figure 21 - AQI Teller Page of website

Figure 21 shows the AQI Teller Page of the website. The user can select the choose file option and input a picture to the website which will use the trained model to calculate the Air Quality level of the image. This will be one of the most important features of the website.

3.3.2.4 Results Page

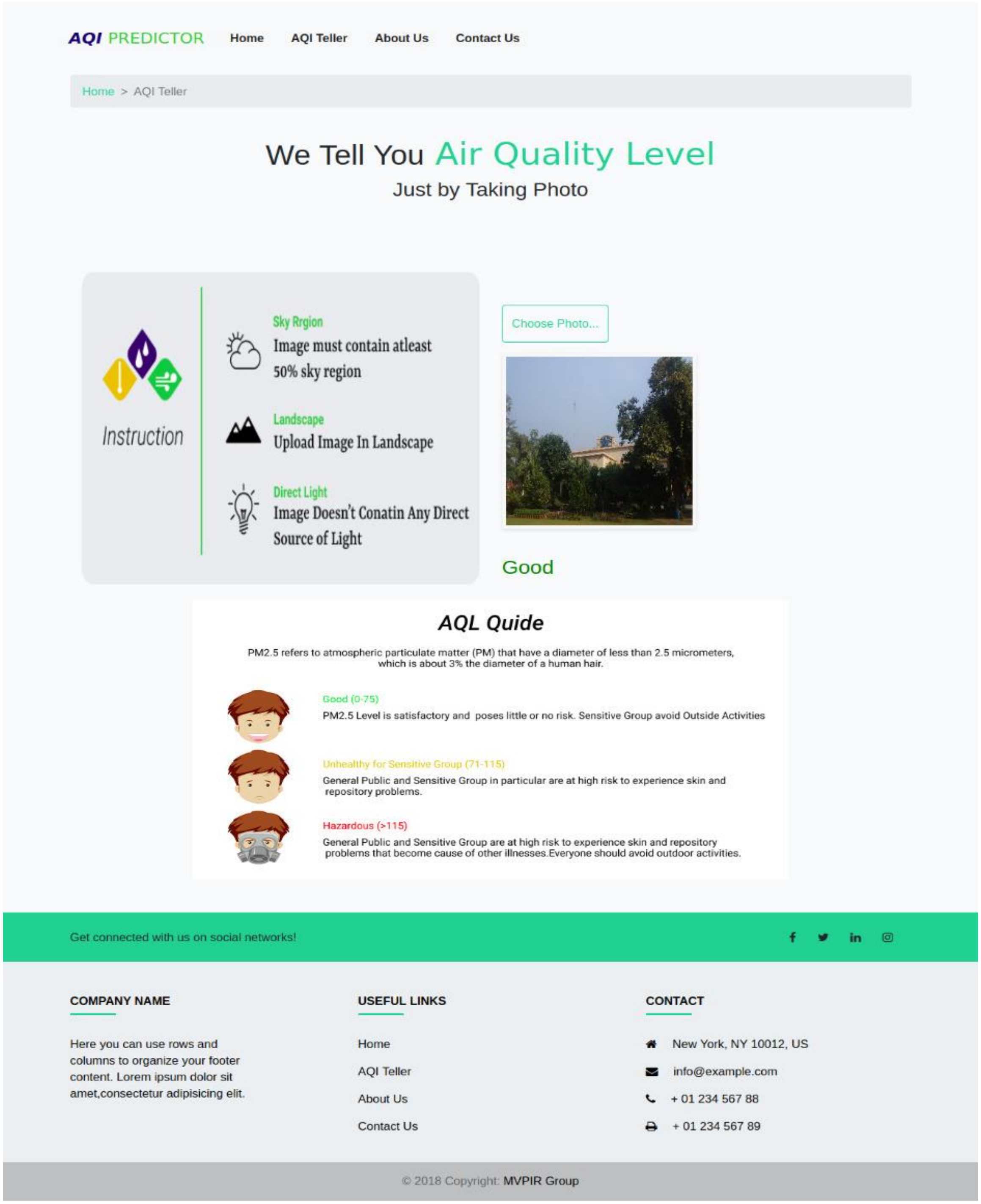

Figure 22 - Results Page (Web)

Figure 22 shows us the results page that is displayed after the air quality level of a photo is predicted.

3.3.2.5 About us

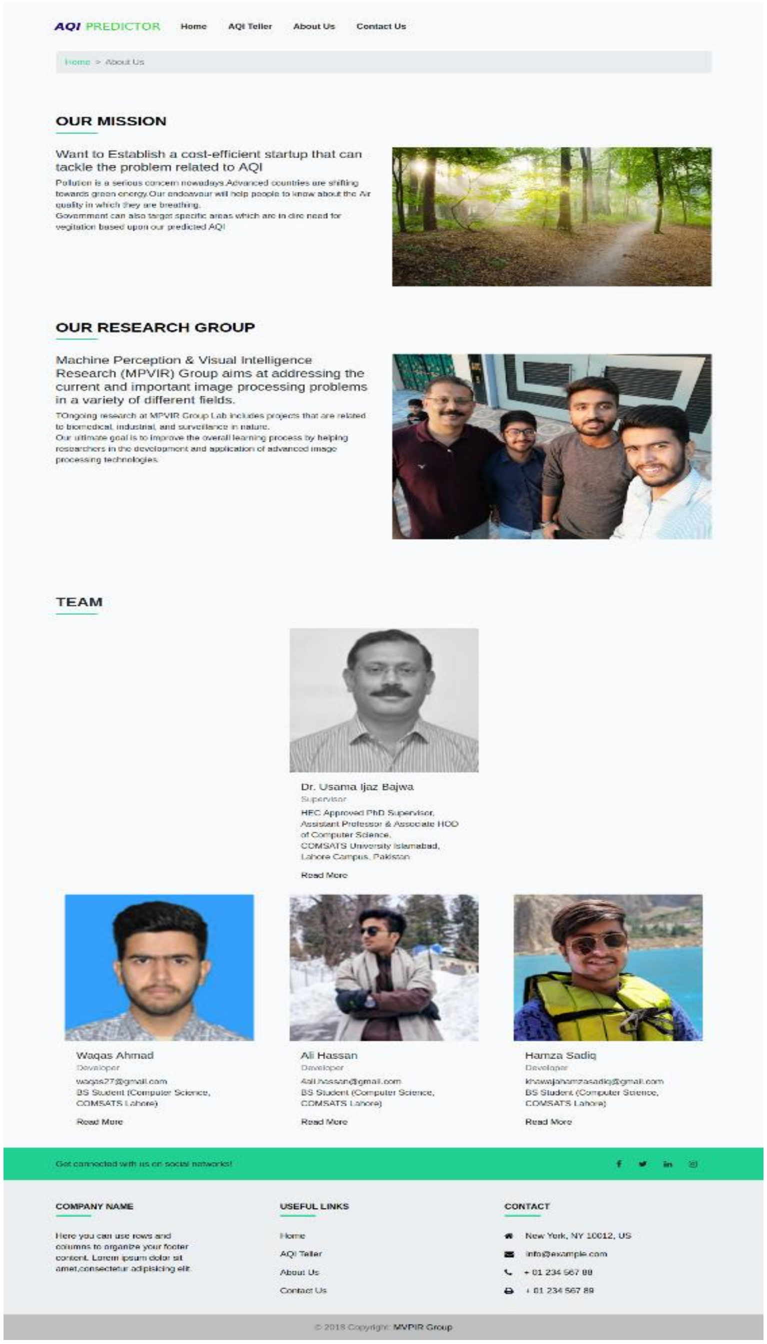

Figure 23 - About us page of website

Figures 23 show the about us page of the website. It houses all the information about what this project is, who are the creators and who is the supervisor in this project.

3.3.2.6 Contact us page

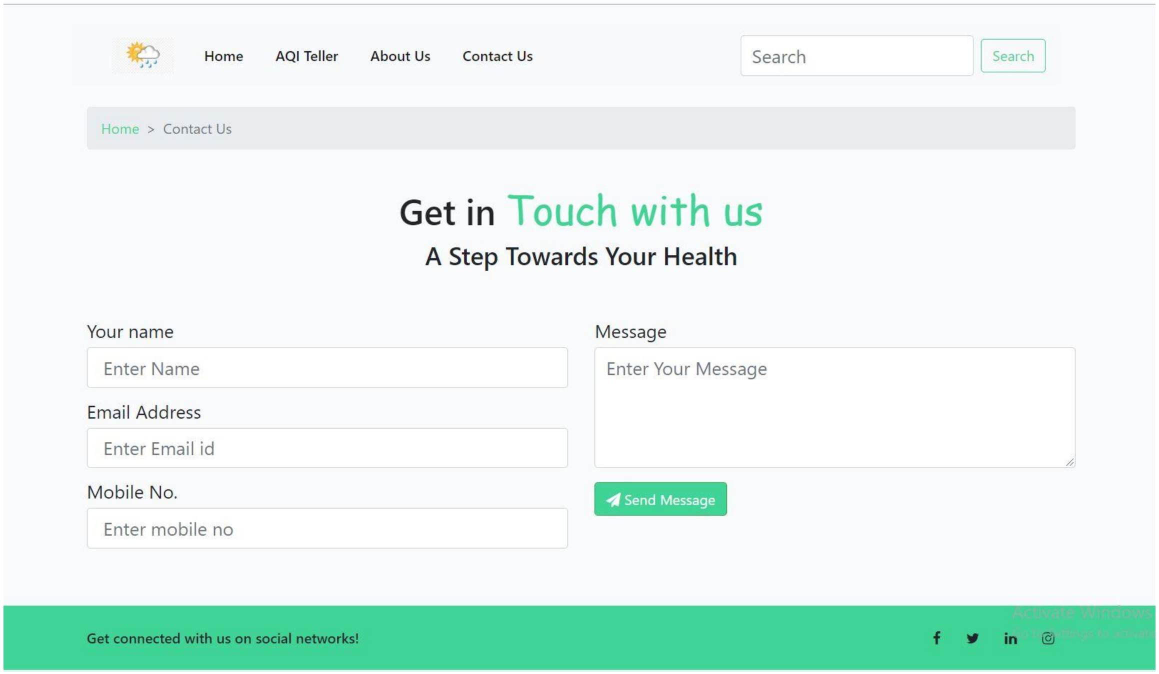

Figure 24 - Contact us page Website

Figure 24 shows the contact us page of the website. It's pretty straightforward. If anybody needs to get in touch or needs some help regarding relevant issues, they can contact us through this simple form.

3.4 Database diagram

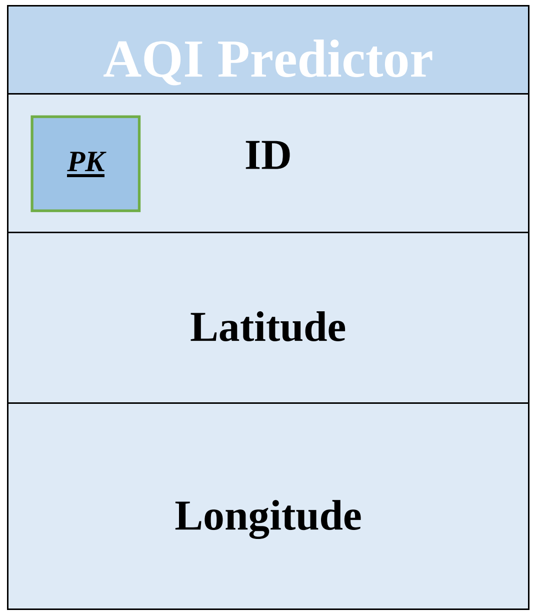

Figure 25 - Database Diagram

Figure 25 shows the database diagram of the project. The database that is used in this project has the following attributes and constraints. There will be location data of the image and the image's link in the form of URL.

3.5 Software Architecture

The following diagram, Figure 26 displays the software architecture that is used in our system.

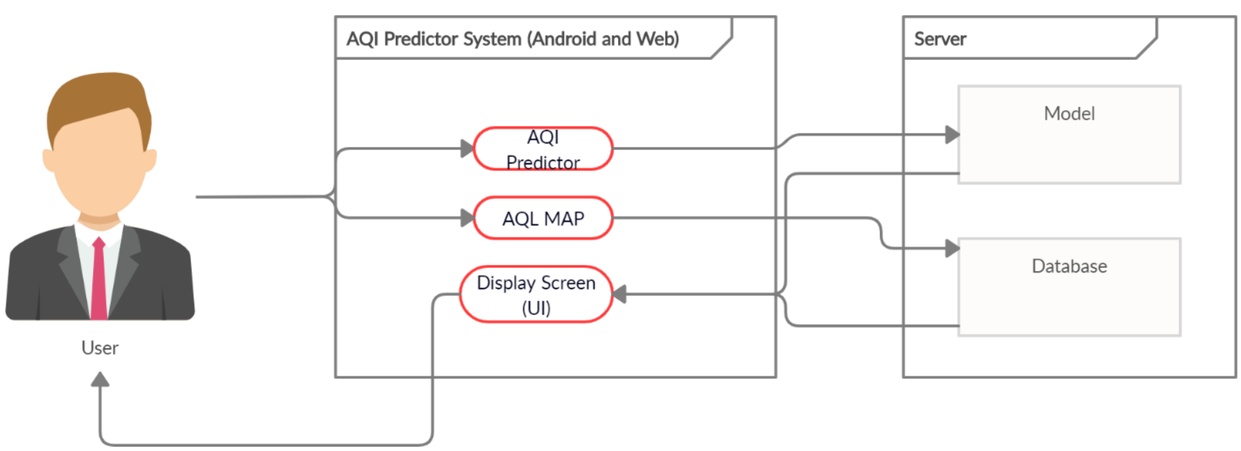

Figure 26 - Software Architecture

3.6 Network Diagram (Gantt chart)

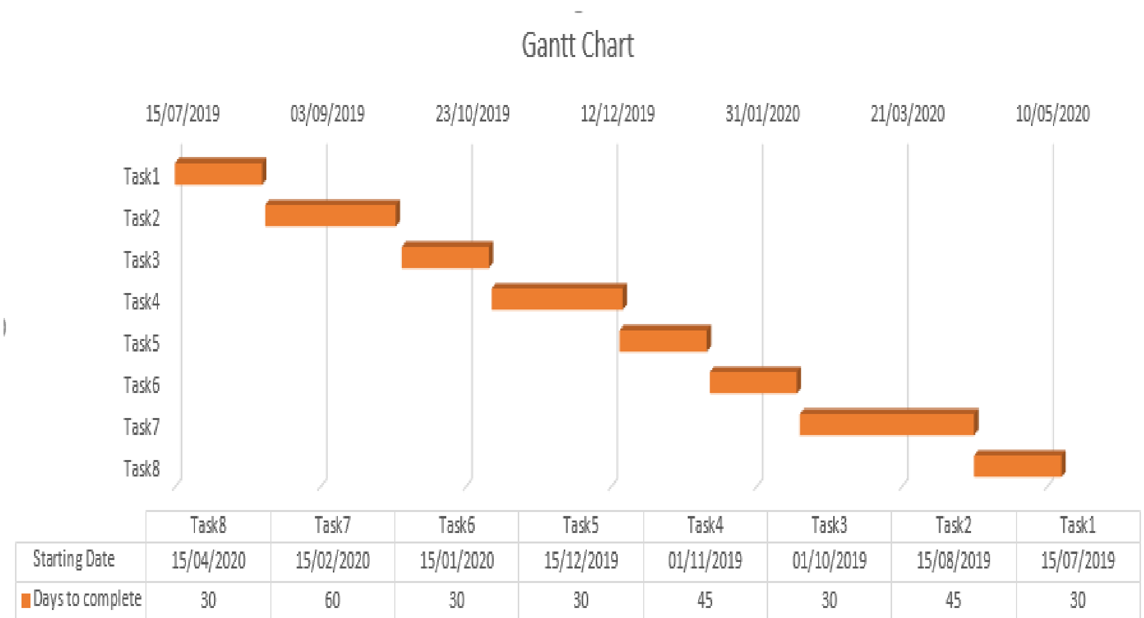

Figure 27 - Gantt Chart

Table 12 - Gantt Chart

Sr No.	Task	Starting Date	Dates to Complete
1	Research existing solutions, Requirement gathering, Prioritizing the requirements	15-July-19	30
2	Documentation (Use cases, Sequence diagrams)	15-August-19	45
3	Gathering the data to train the model, Designing the Front-end	1-October-19	30
4	Development of interfaces	1-November-19	45
5	Training of the model	15-December-19	30
6	Integrating the front-end and backend	15-January-20	30
7	Formal testing of application and the website	15-Feburary-20	60
8	Deployment	15-April-20	30

Figure 27 and table 12 shows the Gantt chart of the project which tells the progress of the whole project with respect to time and explains the timeline of the project.

CHAPTER #4
TESTING

4 Testing

4.1.1 Air Quality Predictor (Android application)

Table 13 – How the user can view the application

Test Case ID: TC-01	
Application Name:	AQI Predictor
Use Case(s):	AQI Predictor Android application
Input summary:	The user will open the application. The user will tap on the image icon to select an image from either gallery or live through camera.
Output summary: Success: The application will take user to the desired page.	
Pre-conditions: User must have turned their location on. User must have an android smartphone with a working camera.	
Post-conditions: Air quality level prediction process will start.	

Table 13 describes how the user views the application upon launching it. The users can choose any option from two available options i.e. open image from gallery or capture from camera.

Table 14 – How the AQ Level is predicted

Test Case ID: TC-02	
Application Name:	AQI Predictor
Use Case(s):	AQI Predictor Android application
Input summary:	After user has provided the picture through either gallery or camera, the application will process the picture and provide Air Quality level and resultant page.
Output summary: Success: Air Quality level has been predicted.	
Pre-conditions: User must have turned their location on. User must have an android smartphone with a working camera.	
Post-conditions: Air quality level will be predicted and shown on the screen alongside other information.	

Table 14 describes how our android application will process the picture and predict Air Quality level of the place in the picture using either the image from gallery or through camera.

4.1.2 Air Quality Predictor (Website)

Table 15 – How the user views the website

Test Case ID: TC-03	
Application Name:	AQI Predictor
Use Case(s):	AQI Level Prediction using the website
Input summary:	On our website, the user can tap on "AQI Teller" page and open it. Then the user clicks on "Choose photo" to upload a photo. The website starts processing the image.
Output summary: Success: The website starts processing.	
Pre-conditions: User must have a working internet connection. User must have a device on which the website can be opened in a supported browser.	
Post-conditions: Air quality level prediction process has started.	

Table 15 describes how the user can view our website to choose a photo of which AQ level they want to check. It can be any photo with a good amount of light and sky in it.

Table 16 – How the AQ Level is predicted on website

Test Case ID: TC-04	
Application Name:	AQI Predictor
Use Case(s):	AQ Level Prediction using the website
Input summary:	After user has provided the picture through the option on website, the application will process the picture and provide Air Quality level and resultant page.
Output summary: Success: Air Quality level has been predicted.	
Pre-conditions: User must have a working internet connection. User must have a device with a supported internet browser.	
Post-conditions: Air quality level will be predicted and shown on the screen alongside other information.	

Table 16 describes how air quality level is predicted on our website. The user chooses the "Choose picture" option on the page of "AQI Teller" and inputs a photo which the website processes and predicts Air Quality level.

4.1.3 Showing notifications to the user

4.1.3.1 Showing results to user on Android application

Table 17 – Showing user the results

Test Case ID: TC-05	
Application Name:	AQI Predictor
Use Case(s):	Showing results to user on Android application
Input summary:	After user has done his part, the application processes the photo and shows the user the results.
Output summary: Success: Air Quality level has been predicted and shown to the user on the android screen in a neat, understandable manner.	
Pre-conditions: AQ Level has been predicted.	
Post-conditions: The user can view the results alongside additional information and precautions.	

Table 17 shows the results of the Air Quality level prediction performed by the application. It shows results alongside additional information such as precautions.

4.1.3.2 Showing results to the user on website

Table 18 – Showing results to User on website

Test Case ID: TC-06	
Application Name:	AQI Predictor
Use Case(s):	Showing results to user on AQI website
Input summary:	After user has done his part, the application processes the photo and shows the user the results.
Output summary: Success: Air Quality level has been predicted and shown to the user on the web page in a neat, understandable manner.	
Pre-conditions: AQ Level has been predicted.	
Post-conditions: The user can view the results alongside additional information and precautions.	

Table 18 describes how the user sees the results from the website after it processes the picture.

4.1.4 Admin/Server End

4.1.4.1 Saved images on drive

Table 19 - Admin can view saved images on Drive

Test Case ID: TC-07	
Application Name:	AQI Predictor
Use Case(s):	Admin/Server End
Input summary:	The admin can view and control saved (by application) images on Google Drive.
Output summary: Success: Admin can view images saved on Google Drive.	
Pre-conditions: Saved images on Drive.	
Post-conditions: AQ Level values are saved in Database.	

Table 19 describes how the admin can see the images saved on drive.

4.1.4.2 Saving values in Database

Table 20 - Admin can see saved values in Database

Test Case ID: TC-08	
Application Name:	AQI Predictor
Use Case(s):	Admin/Server End
Input summary:	The admin can view and control saved values in database.
Output summary: Success: Admin can view values saved in database.	
Pre-conditions: Saved values in database.	
Post-conditions: None	

Table 20 describes how the admin can see AQ level values saved in Database.

CHAPTER #5
CONCLUSION

5 Conclusion

5.1 Problems faced

During the course of the entire project, we faced a lot of problems. We have discussed the problems and their possible solutions below.

5.1.1 Dataset Collection and Annotation

Today, the fields of Machine Learning and Deep Learning are far more advanced and successful than other fields of Computer Sciences. The major reason behind the success of this field is the growth of the value of data. Data could be said to be the pillar upon which this field is built. Because machines don't sense and think like humans, they require data to learn, operate, and perform any actions. Hence, our first requirement for the project was data.

Collecting the dataset was incredibly difficult as we needed a dataset that was either of Pakistan or was similar to Pakistan in terms of atmosphere, climate and sky color. We researched and tried to find datasets from different possible sources of Pakistan and the relevant authorities in the country but we couldn't find any. Therefore, we decided to use the dataset of China that was found after constant efforts. It was found to be appropriate to use as Chinese climate is similar to Pakistan's.

As we found our dataset, it was found that the dataset was not annotated. Therefore, manual labelling had to be done. We annotated our dataset with the help of the timestamp that was found on every picture. We matched the timestamp with the data US embassy provides online to label it.

5.1.2 Selection of Technique

The next problem we faced was how to select our technique to approach our main problem. It was either to use Machine Learning or to use Deep Learning or use a combination of both. First we applied regression. To apply regression, we extracted five features from images:

1. Transmission
2. Blue Colour of the Sky
3. Gradient of Sky Region
4. RMS Contrast
5. Entropy

We extracted these features from images on MATLAB. Using these features, we trained a linear regression model but that model didn't predict right on any given picture from the same dataset.

After this, we tried the Deep Learning approach but for that the dataset wasn't enough as Deep Learning techniques require large datasets. For this, we performed augmentation on the images in our dataset. The results were substantial as the dataset increased to 18,000 images. The techniques used to perform augmentation were:

- Shear
- Random Zoom
- Flip Left-Right
- Translation
- Random Crop

After that we applied transfer learning techniques on our model, and then we trained the mode. The results were good enough to go ahead and we deployed this model.

5.1.3 Model deployment

The biggest issue that we had to face on our way to completing the project was how to port the application on a smartphone as the model was too large to run on such platforms. We found two solutions, either to transform the model into another model which was either light or supported on such platforms, or to put the model at the backend server and receive requests from the mobile application and respond. Both choices have their advantages and disadvantages that we discuss below:

5.1.3.1 Port to Android

The model we trained was large in size. This adds to the task that is to deploy it on any mobile platform (iOS/Android). If we want to deploy the model on the mobile application, we would have needed to convert the keras model (original model) into tensorflow and then the tensorflow model into tensorflow lite model as the lite model is lighter and flexible enough to work on mobile devices. Unfortunately, that also accounts for the loss in accuracy.

The process to convert the model to tensorflow and then to tensorflow lite holds a lot of challenges and tricky situations. Add that to the cost of accuracy loss and the need to test it each time we converted it, this method became unfeasible.

5.1.3.2 Serve The Model

The other option, which was the chosen solution to the problem we faced in model deployment was to serve the model. The following steps were used to establish this:

1. Establish a Server

First, we created a server. This was done by using the OS of Ubuntu on our PC through tutorials found on the internet. The server used flask framework to establish it.

2. Communication between server and applications

Once the model is placed on the server, we created an android application and a website that communicated with the server.

We chose this option to solve our problem as we were able to obtain our required results using this method. Volley framework was used to connect our android application to the server.

5.2 Project summary

In conclusion, our project, AQI Predictor, consists of an app and a website. The functionality that predicts Air Quality level and the model embedment at the backend server will be shared across the app and the website. The app and website will use a deep learning model that will help in predicting the Air Quality level of the place in the picture. Both platforms will be synchronized in terms of data through a mutual database that is connected with both website and android application. All the results and data will be stored in the database.

The data collected through the app will populate our dataset which can later be used for further purposes. One of these purposes will be to show a map on the website that will have the AQI of the places that were calculated by the users in the app. This will help in checking Air Quality level through the website as well as the app, whichever the users prefer.

It can further tell people about more details including if the air is harmful/good and what the necessary precautions should be depending upon the AQI. The AQI can further help authorities and other relevant bodies in deciding and gathering data which is usually collected through expensive sensors.

5.3 Future work

As of now, our project is going to be of intermediate professional level which can predict AQI with good accuracy that will be sufficient for our usage and will only use photos with natural light (day time) to predict results correctly. Currently, our system doesn't calculate exact AQI value. It calculates the level of the Air Quality based.

In the future, it can be further enhanced by providing the model with a much larger dataset that includes various countries to train it further so it can work for those places too. It can also be updated if there's a way found such that the model can predict the AQI value. The model can also be enhanced in a way that the accuracy can be increased and can work at night time as well. This will increase the efficiency and effectiveness of this project in the future. Furthermore, as our website grows in functionality and data that will be constantly collected, it will reach a much wider audience that can help us grow into a much larger and effective society helping us open the doors for many more possibilities.

5.4 Github Respority Links

The code and dataset of project is available on followings links:

5.4.1 For Android App Code:

https://github.com/waqas2727/AQIPredictorApp

5.4.2 For Website Code:

https://github.com/waqas2727/AQI-Predictor-Web-App

5.4.3 For Web Application:

https://drive.google.com/file/d/1oOoP3vU5sremb13PsBO-GHFkEyd8xt4B/view?usp=sharing

5.4.4 For Dataset:

https://drive.google.com/drive/folders/1m14IyFM466uotXo9fy-J4ZqwbV4z0wdp?usp=sharing

5.5 Project Description Link

Our Work is cited at Machine Preception & Visual Research Group Website.

It can be seen at the following link:

https://sites.google.com/view/mpvir/projects/aqi-predictor

CHAPTER #6
REFERENCES

6 References

[1] AirVisual, “Air Quality || Air Visual,” Air Visual, 2019. [Online]. Available: https://play.google.com/store/apps/details?id=com.airvisual&hl=en. [Accessed May 2020].

[2] AirCognizer, “Air Cognizer,” 2019. [Online]. Available: https://play.google.com/store/apps/details?id=com.bvpiee.com.airmirror&hl=en_US. [Accessed May 2020].

[3] M. S. Vahdatpour, H. Sajedi and F. Ramezani, “Air pollution forecasting from sky images with shallow,” *Earth Science Informatics,* vol. 11, no. 3, pp. 413-422, 2018.

[4] Z. Wang, W. Zheng, C. Song, Z. Zhang, J. Lian, S. Yue and S. Ji, “Air Quality Measurement Based on Double-Channel Convolutional Neural Network Ensemble Learning,” *IEEE Access,* vol. 7, pp. 145067-145081, 2019.

[5] A. Chakma, B. Vizena, T. Cao, J. Lin and J. Zhang, “Image-based air quality analysis using deep convolutional neural network,” in *2017 IEEE International Conference on Image Processing (ICIP),* Beijing, 2017.

[6] R. Li, X. Liu and X. Li, “Estimation of the PM2.5 Pollution Levels in Beijing Based on Nighttime Light Data from the Defense Meteorological Satellite Program-Operational Linescan System,” *Atmosphere 2015,* vol. 6, pp. 607-622, 12 May 2015.

[7] C. Liu, F. Tsow, Y. Zou and N. Tao, “Particle Pollution Estimation Based on Image Analysis,” *PLOS ONE,* vol. 11, no. 2, 1 February 2016.

Appendix A

1st page of Turnitin Report

Turnitin Originality Report

CS15-AQI_Predictor-Final_Thesis.docx by Anonymous

From Research Papers 2 (Research Papers 2)

- Processed on 23-Jul-2020 6:23 AM PDT
- ID: 1361184235
- Word Count: 7916

Similarity Index
12%
Similarity by Source

Internet Sources:
4%
Publications:
3%
Student Papers:
10%

sources:

1 — 1% match (student papers from 20-May-2016)
Submitted to Higher Education Commission Pakistan on 2016-05-20

2 — 1% match (publications)
M. M. Samsami, N. Shojaee, S. Savar, M. Yazdi. "Classification of the Air Quality Level based on Analysis of the Sky Images", 2019 27th Iranian Conference on Electrical Engineering (ICEE), 2019

3 — 1% match (student papers from 15-May-2018)
Submitted to Higher Education Commission Pakistan on 2018-05-15

4 — 1% match (student papers from 16-May-2018)
Class: OS
Assignment: Assignment 1
Paper ID: 964510345

5 — < 1% match (student papers from 11-Jun-2019)
Submitted to Higher Education Commission Pakistan on 2019-06-11