

**Advanced Software Engineering**

*CS 5551*

*Spring 2018*

***Project Report***

**Submitted By:**

**Aravind Sheri  
Sharath Koppu  
V Bhanu Sudheer  
Aditya Soman**

**Department of Computer Science and Electrical Engineering  
University of Missouri - Kansas City**

## **Problem Statement**

- When tourists visit different countries and places where the regional languages are different, and it would be difficult to communicate and read the local language text.
- While traveling there would be signs or boards written in the local language and it would be very difficult in understanding and follow the rules or path of the destination location.
- There would be places where MAPS is not advanced or not available so, following the signals and local language written path is the only available option to reach the destination.
- Communicating with the local people where the regional language is only the medium or dominated way of communication would be barrier to communicate.
- Health issues would be another problem for the tourists with a sudden change of weather and diet.
- Proper medication at the right time in emergency situations in remote places is highly impossible or it would take time to reach the medical help.

## **Project Plan**

- Converting the non-local language text image into English or the preferred language of the tourist using API so that the tourist can understand the local language text or road signs.
- Speech conversion from the Tourist language to the regional language of the place visited using speech conversion API.
- We can provide the weather conditions and frequent health issues occurred in the various tourist places will be provided in the app so that precautionary steps would prevent various hazardous diseases.
- We can provide tele health or live video instructions so that the primary medication would save a life. But contacting the right person or hospital details would be difficult or mobile signal would not be available.
- We can provide pre-loaded medical steps for the emergency health issues so that primary medication would save someone's life.

**Required URLs:**

1. Github Link: <https://github.com/sharathk91/Team10ASEProject>
2. YouTube Link: <https://www.youtube.com/watch?v=Clp-Xcs8A3A>
3. Zenhub Link:  
<https://app.zenhub.com/workspace/o/sharathk91/team10aseproject/boards?repos=120061974>
4. Presentations:
  - a. Increment 1:  
<https://github.com/sharathk91/Team10ASEProject/blob/master/Increment%201%20ppt.pptx>
  - b. Increment 2:  
<https://github.com/sharathk91/Team10ASEProject/blob/master/iteration%202.pptx>
  - c. Increment 3:  
<https://github.com/sharathk91/Team10ASEProject/blob/master/Iteration%203.pptx>

INCREMENT 1

**Project Goal and Objectives: Motivation:**

Traveling the world or visiting an unknown place is always exiting and very refreshing, because of the various local languages communication is the primary problem and traveling is very hard and it makes life difficult over there. Our Application provides a medium to communicate in any language and various other functionalities to make a tourist's life easy.

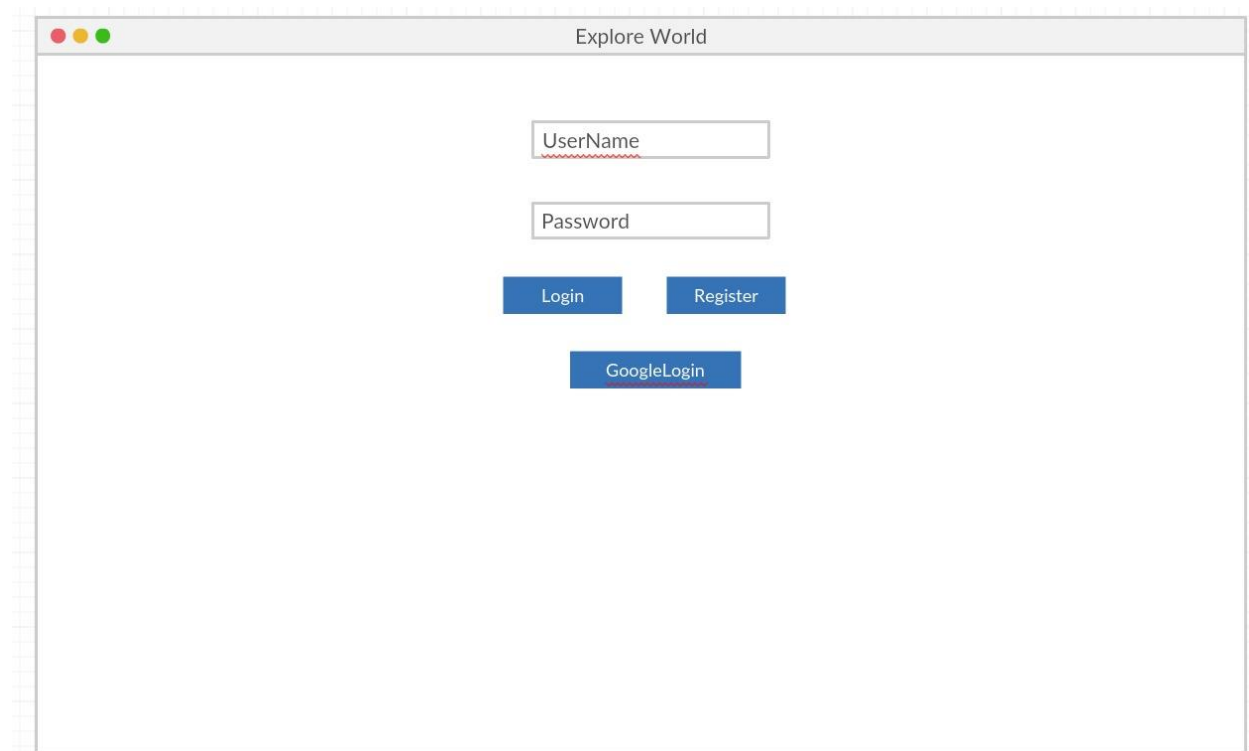
**Significance/uniqueness:** This Application has the important functionality so called a medium to communicate with local people through text to text conversion and text to speech conversion. In this application Augmented reality is used to convert any image text to normal text and navigation is very easy so that it wouldn't be any difficulty in navigating the world

**Objective:** Our Application will reduce the various barriers of a tourist to travel around the world and it will run on any platform to make life easy **System Features:** •Application has the main feature as language translation that is either text from user language to text of local language of various regions •User can register or can sign in through google API so that it wouldn't take more time to get into application and use the services

**First Increment Report: Services and API Used:** •IBM Text to Text language translation.  
•Google API for user sign in

**Detail Design of Features: Wireframes and Mockups: Below are the screenshots for the wireframes and Mockups:**

1.



2.

The image shows a registration form window titled "Register". The window has a standard macOS-style title bar with three colored buttons (red, yellow, green) on the left. The form contains six text input fields stacked vertically, each with a label above it: "Firstname", "Lastname", "Username", "Password", "Email", and "Phone No". Below these fields is a blue "Submit" button. The entire form is centered within the window.

Register

Firstname

Lastname

Username

Password

Email

Phone No

Submit

3.

Explore World

Home

About

Contact

Logout

Source Language

Target language

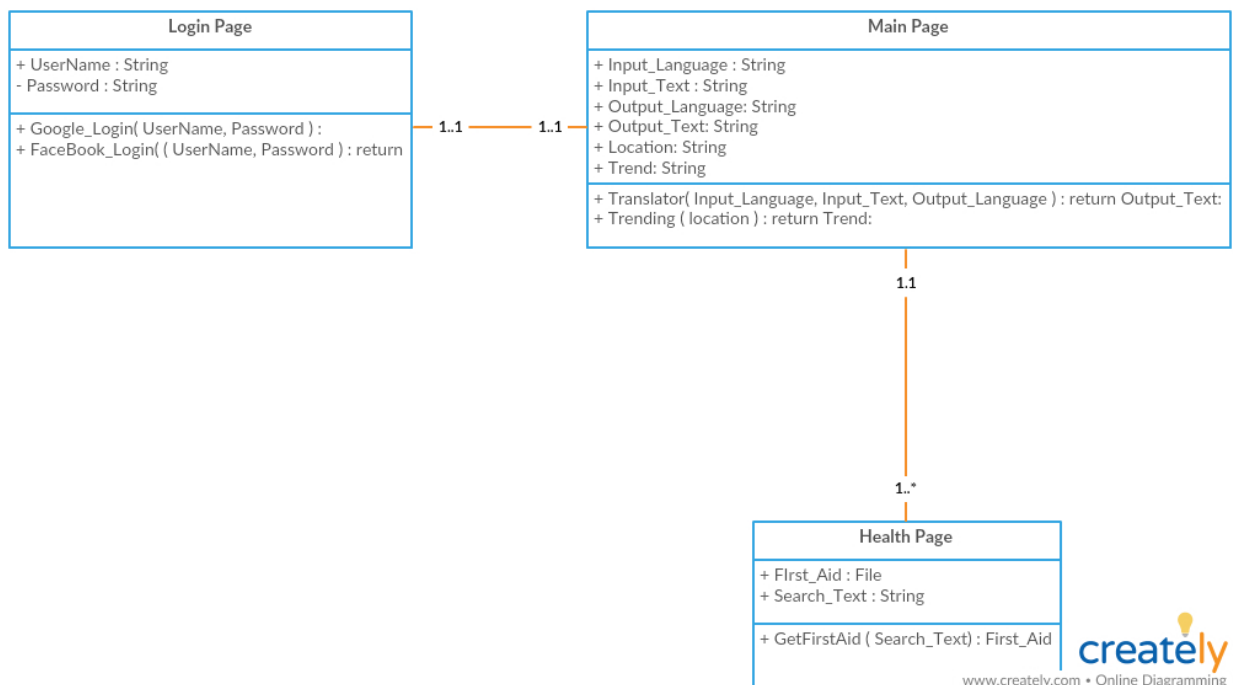
Enter source text

Converted Target text

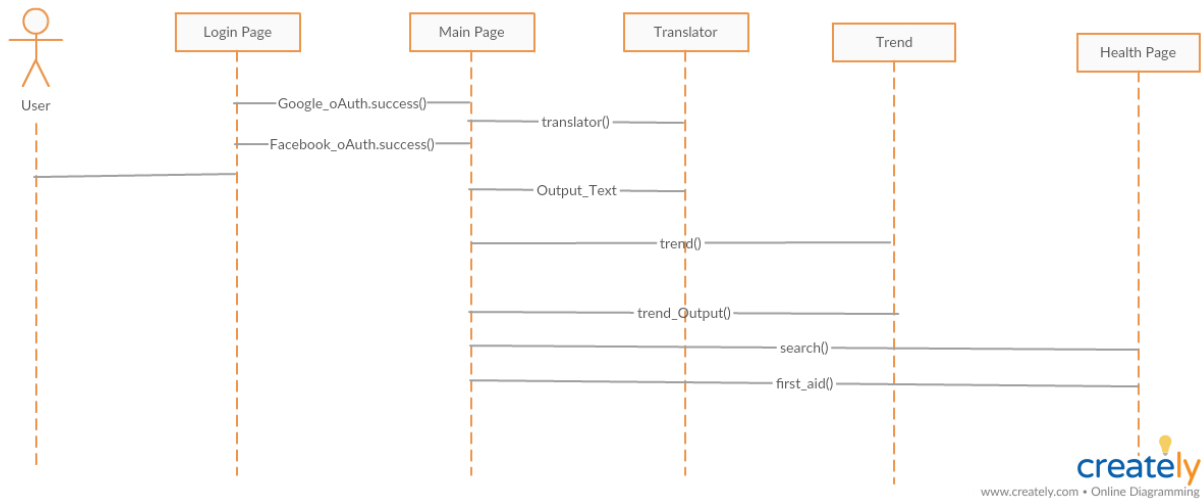
Translate

## Architecture Diagrams:

1.



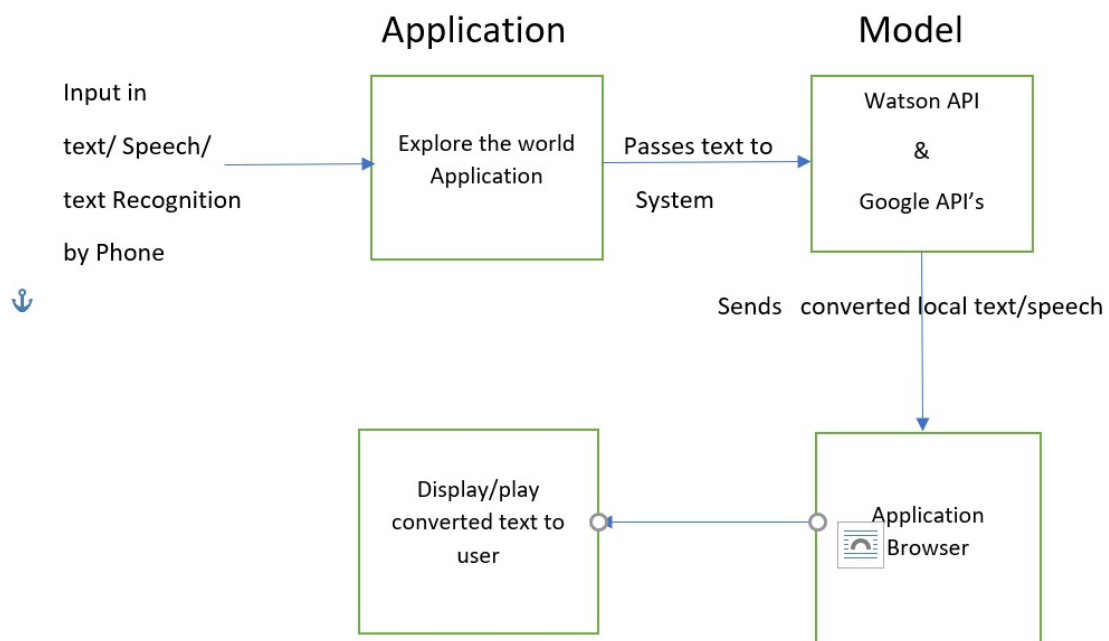
2.



3.

## Explore The World

### Architecture





**Testing:** Tested the application with the below use cases

Test case Type	Test case Description	Expected Output	Application Output
Login Page	Check with wrong username/password	It should throw an Error and should not go to home screen	Validated login screen and throwed the error
Registration Page	Give password less than 6 characters	Shows the alert message password is less than 6 characters	Alert message came up with password length count
Text to Text Conversion	Given the source text with English and target text as french	It should convert text from English to French	Converted the text and shown in the UI

**Implementation:** Created application in WebStorm which has login page and registration page where user can register to use the services provided in the application and also used Google API for sign of user with ease and IBM text to text language conversion.

**Below are the steps involved in creating the application**

1.Created a project in WebStorm and created Google and IBM text to text language conversion API keys so that we can use in the project to use various services

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Explore World</title>

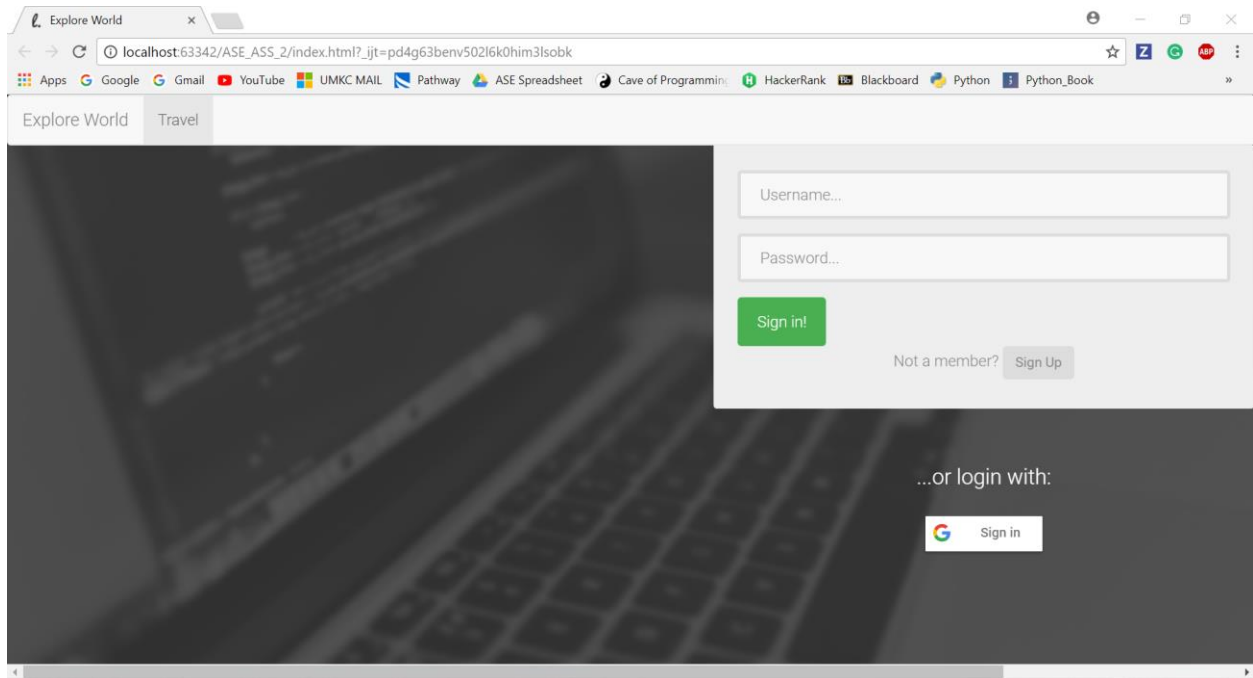
  <link rel="stylesheet" href="http://fonts.googleapis.com/css?family=Roboto:400,100,300,500">
  <link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">
  <link rel="stylesheet" href="assets/font-awesome/css/font-awesome.min.css">
  <link rel="stylesheet" href="assets/css/form-elements.css">
  <link rel="stylesheet" href="assets/css/style.css">
  <link rel="stylesheet" href="css/cssforlogin.css">

  <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
  <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>

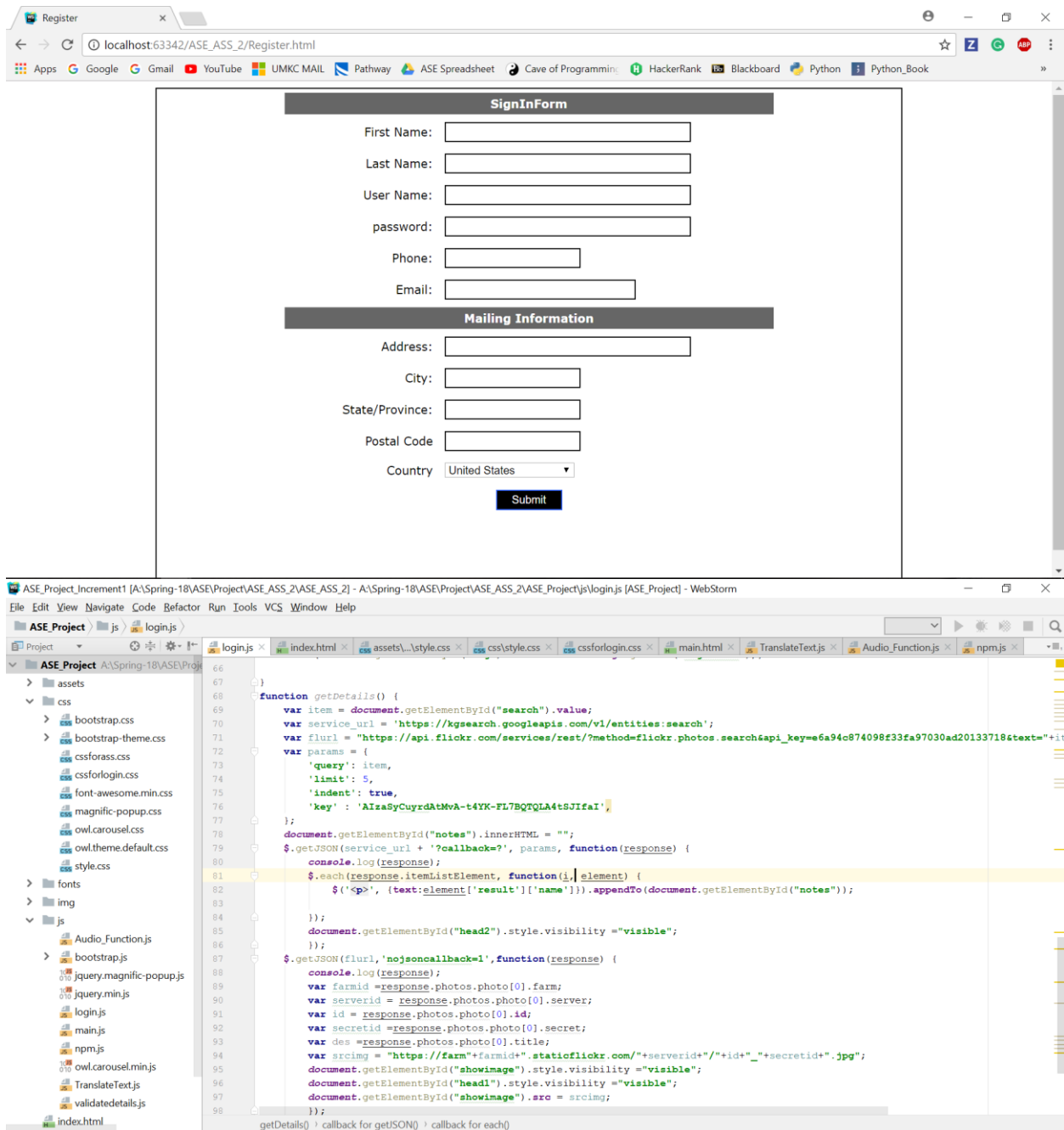
  <!-- Favicon and touch icons -->
  <link rel="shortcut icon" href="assets/ico/favicon.png">
  <link rel="apple-touch-icon-precomposed" sizes="144x144" href="assets/ico/apple-touch-icon-144-precomposed.png">
  <link rel="apple-touch-icon-precomposed" sizes="114x114" href="assets/ico/apple-touch-icon-114-precomposed.png">
  <link rel="apple-touch-icon-precomposed" sizes="72x72" href="assets/ico/apple-touch-icon-72-precomposed.png">
  <link rel="apple-touch-icon-precomposed" href="assets/ico/apple-touch-icon-57-precomposed.png">

  <script src="js/login.js"></script>
  <script src="https://apis.google.com/js/platform.js?onload=onLoad async defer"></script>
  <meta name="google-signin-client_id" content="1088887293043-2btu5h68vociupkd57fj3vo01qavik.apps.googleusercontent.com">
</html>
```

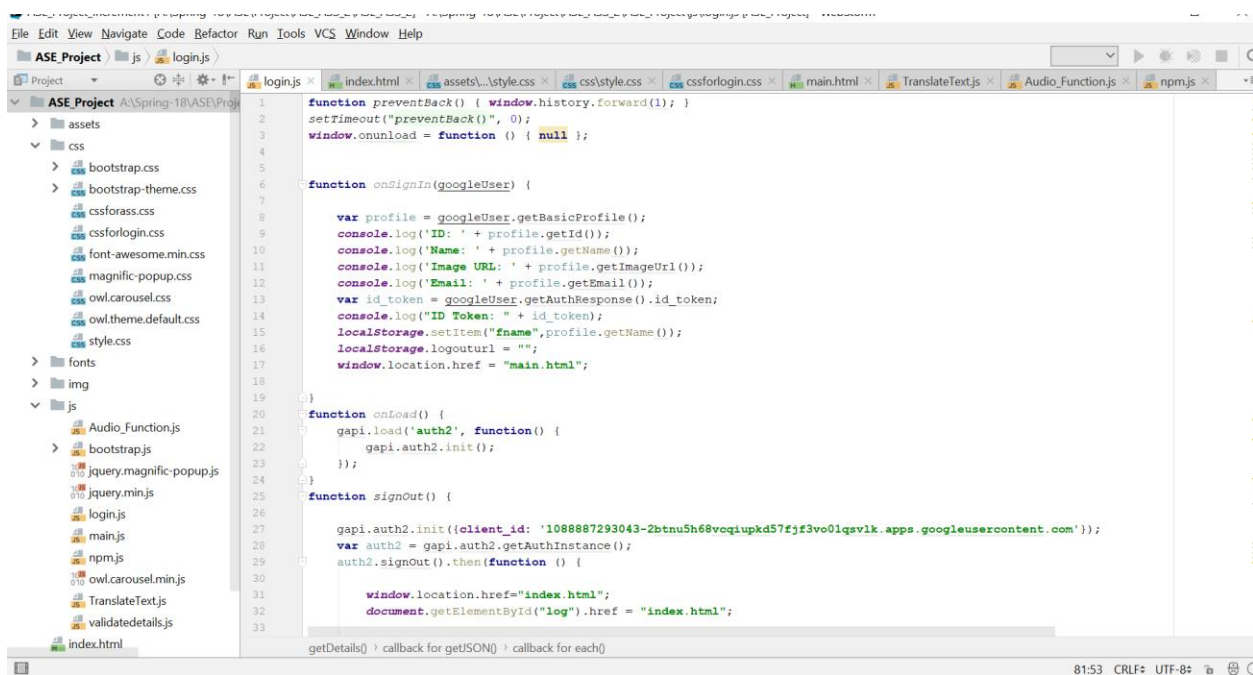
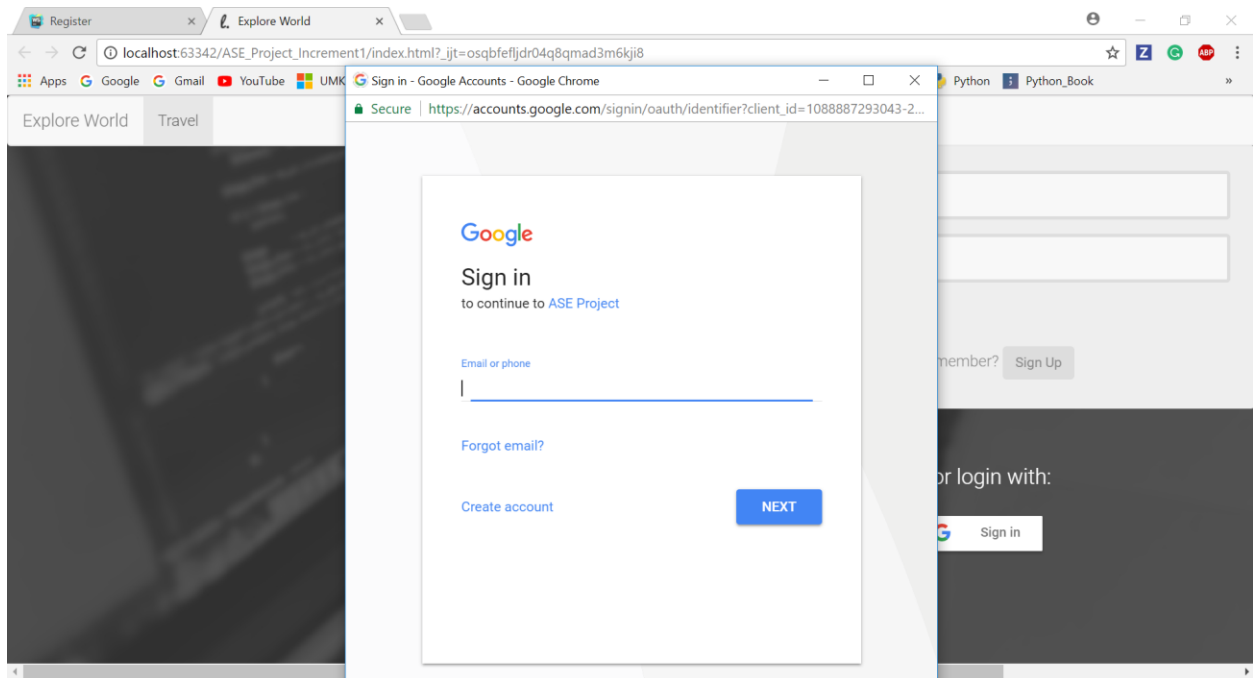
2.Created a login page for the user to login and use the various services provided in the application



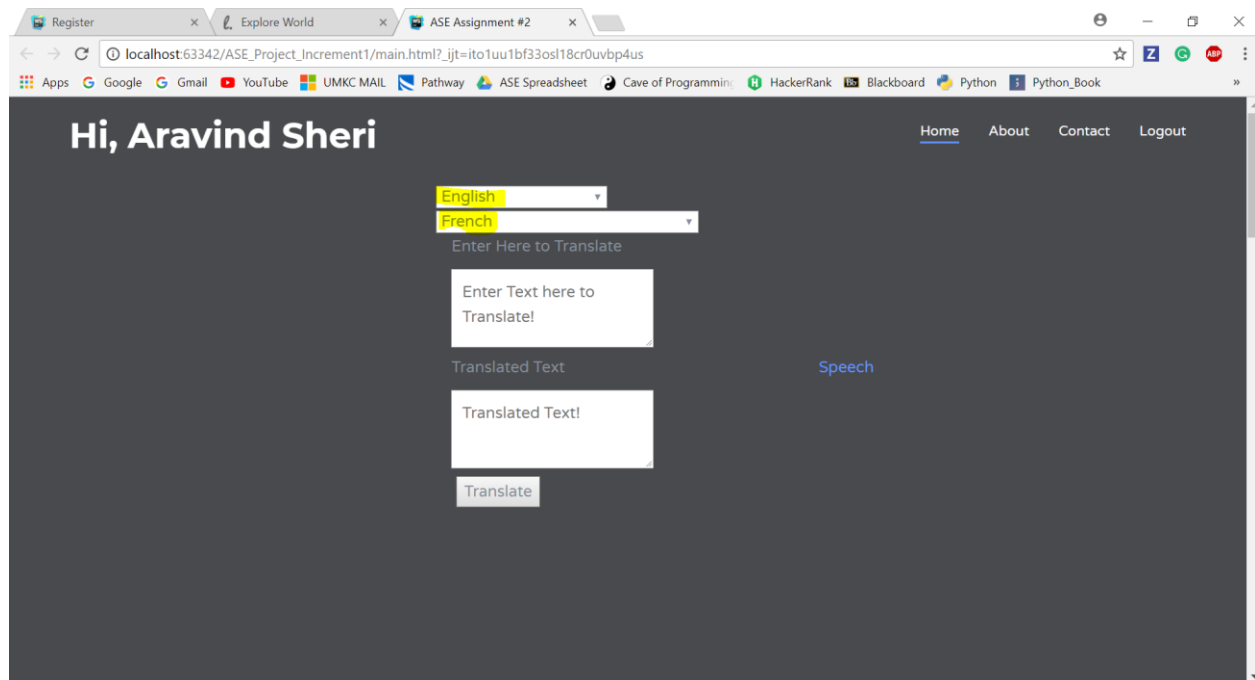
3.If the use is new to the application or never used the application he should register using registration page or user can login using Google account



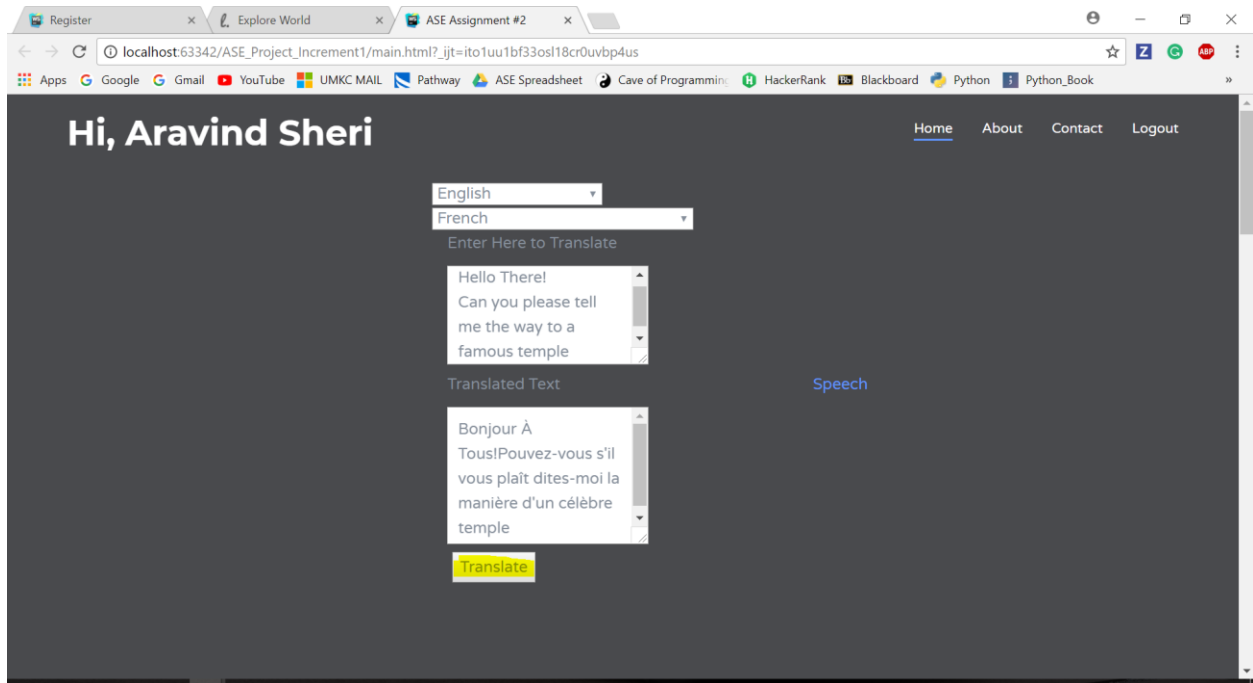
**Google Login:** We used Google API for using login into the application using Google username and password to make life easy.



4. After user login into the application he can use all the services that are available in the application. After login the user will redirects to the main page where user has to select language drop down to enter the text and the language he wants to translate the entered text.

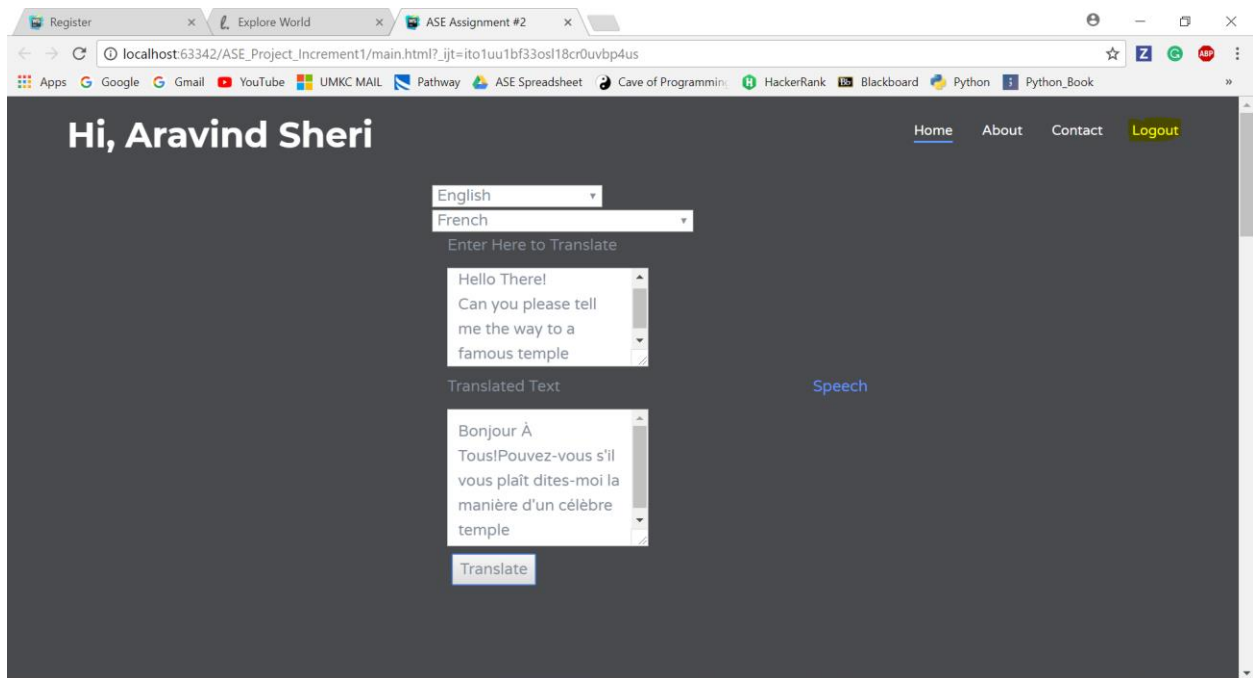


5. After selecting the languages user will enter the translate button to translate the text which he selected so that the local people can read and understand the situation or the intent of tourist is trying to explain. The text boxes will expand automatically as the user enters the text so that it wouldn't be confusing for both of them.

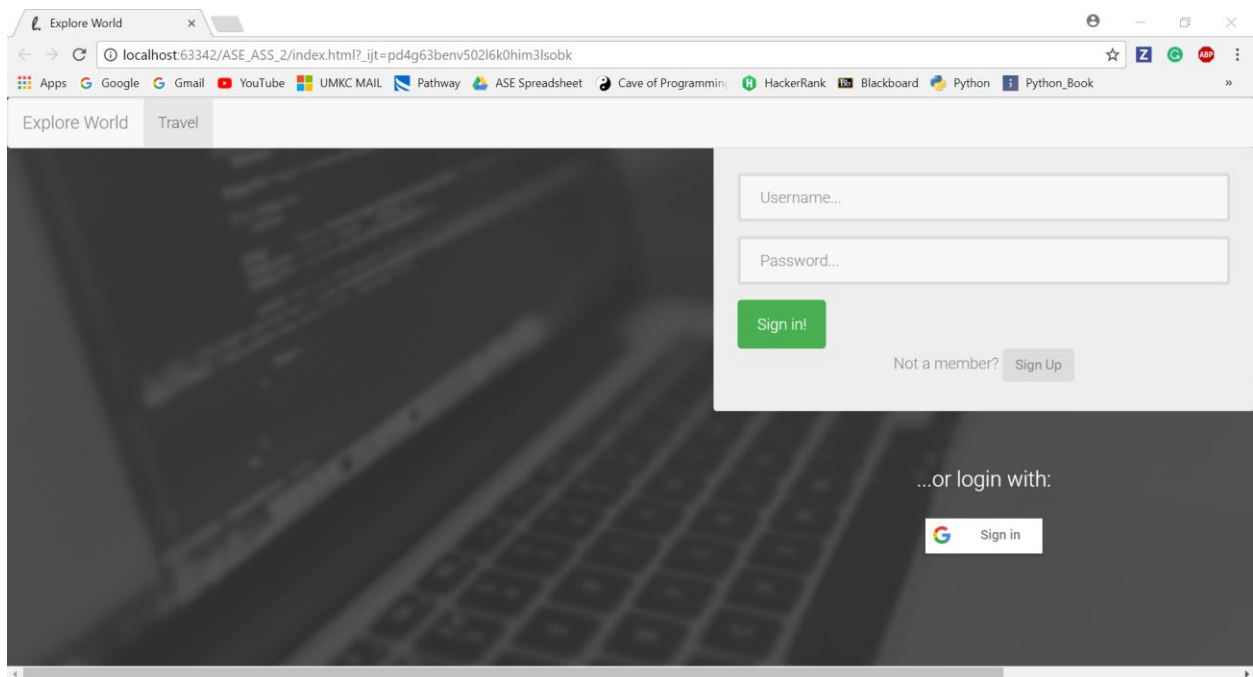


6. User can contact for any technical help using contact tab and if he wants to know more details of the application and the details of the development he can use the About tab.

7. After the use of various services in the application user can logout and he will be redirected to the login page so that he can login again if he wants to.



8. User will be redirected to login page after the logout



**Project Management: Implementation Status Report: Work Completed:**



1.Login Page – Login Page is the first page of the application where user will be able to login with his username and password and if the user is new to the application he can register with register page or he can login using Google. **Responsibility and Time Taken:** • Aditya – Designed the Login Page (5 days) • Sharath - Implemented the login page (8 days) • Bhanu – Tested the login page (5 days) 2.Registration Page – New Users will be able to register using the register page where he will he should enter the various details and the basic validations are performed using JavaScript.

**Responsibility and Time Taken:** • Aravind – Design and Implementation of the registration page (5 days) • Bhanu – Tested the registration page with use cases (5 days)

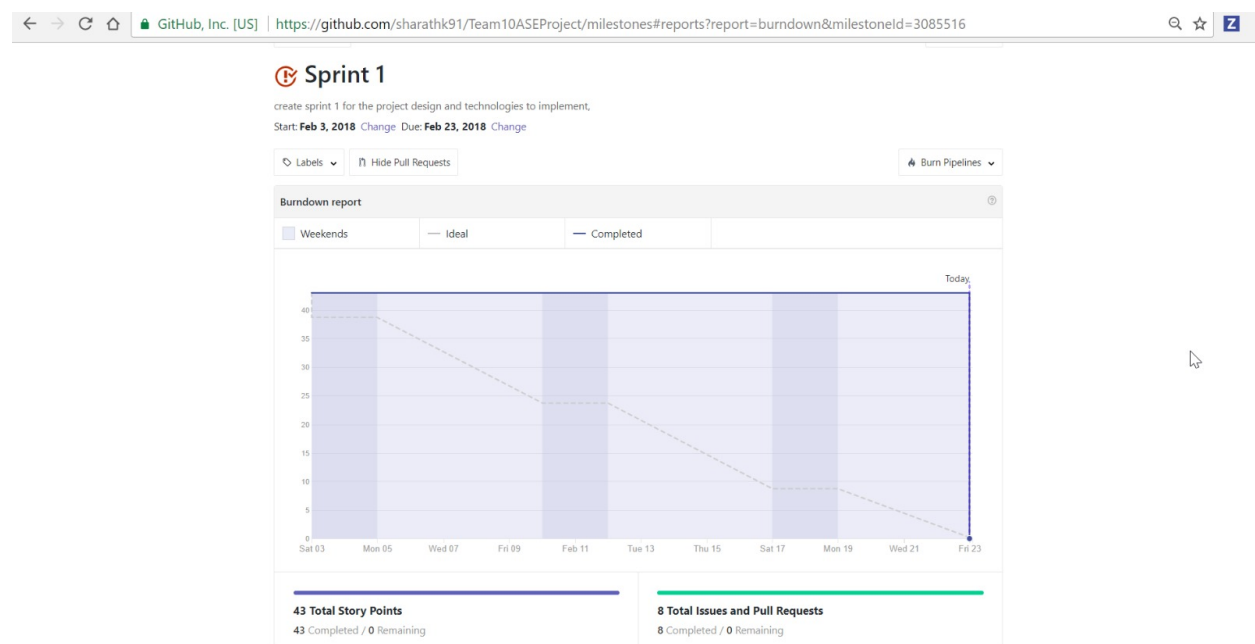
3. Text to Text Language Translation: User will selects the source language where he enters the text in the source language selected and he will select the language in which the entered text to be converted. Responsibility and Time Taken: • Aravind – Design, Implementation and Testing of the translation page (5 days).

4.Architecture and Wireframe Diagrams: Describes the various functionality of the application.

**Responsibility and Time Taken:** • Sharath – Designed wireframes, architecture and blueprint of the project (5 days).

5.Class and Sequence Diagrams: Explains the functionality of the application. • Aditya - Designed the class and sequence diagrams.

### Sprint Burn Down Chart:



## INCREMENT 2

### Services and API Used:

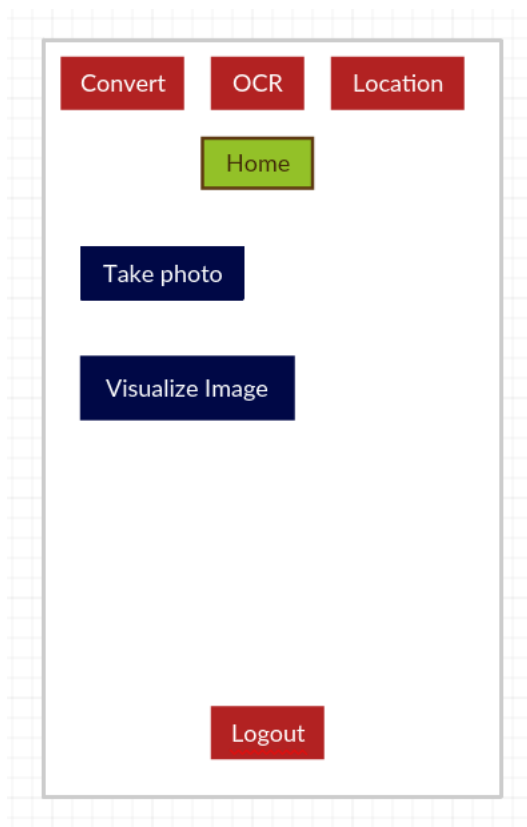
1. Yandex API for Text to Text language translation.
2. Firebase for storing the details of the user and email authentication.
3. OCR plugin for extracting text from image.
4. Google maps API for maps and user location and navigation.
5. Geolocation for getting the user current location.
6. Clarifai for visual detection and getting insights from an image.
7. Google API for text to speech of the translated text.

### Detail Design of Features:

### Wireframes and Mockups:

Below are the screenshots for the wireframes and Mockups:

#### 1. Home Page



## 2. OCR Plugin

Diagram illustrating the OCR Plugin interface. The interface consists of a top navigation bar with three buttons: 'Convert', 'OCR' (highlighted in green), and 'Location'. Below the navigation bar, the main content area contains three buttons: 'Take photo', 'Choose photo', and 'Analyse text'. At the bottom of the interface is a 'Logout' button.

## 3. Convert

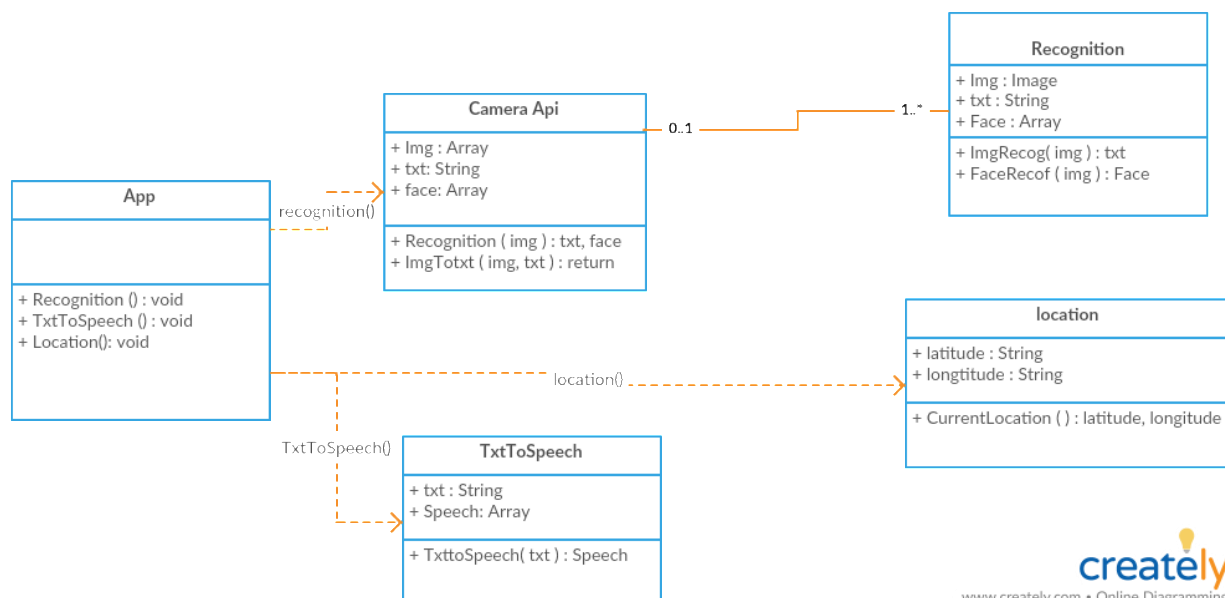
Diagram illustrating the Convert interface. The interface consists of a top navigation bar with three buttons: 'Convert' (highlighted in green), 'OCR', and 'Location'. Below the navigation bar, the main content area contains four input fields: 'Source text', 'Source lang', 'Target lang', and 'Converted text'. Below the input fields are two buttons: 'Submit' and 'Logout'.

## 4. Maps

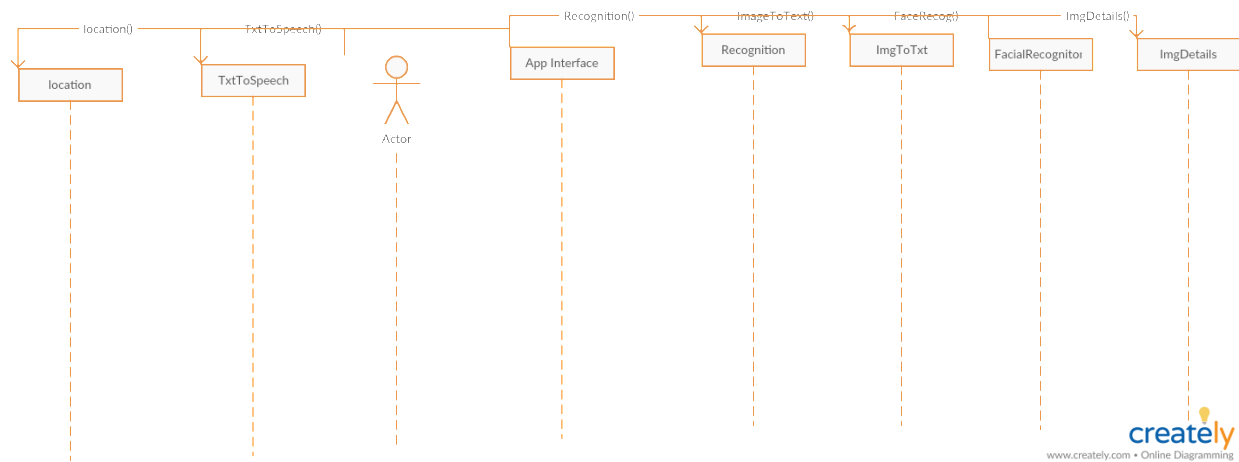


### Architecture Diagrams:

#### 1. Class Diagram:



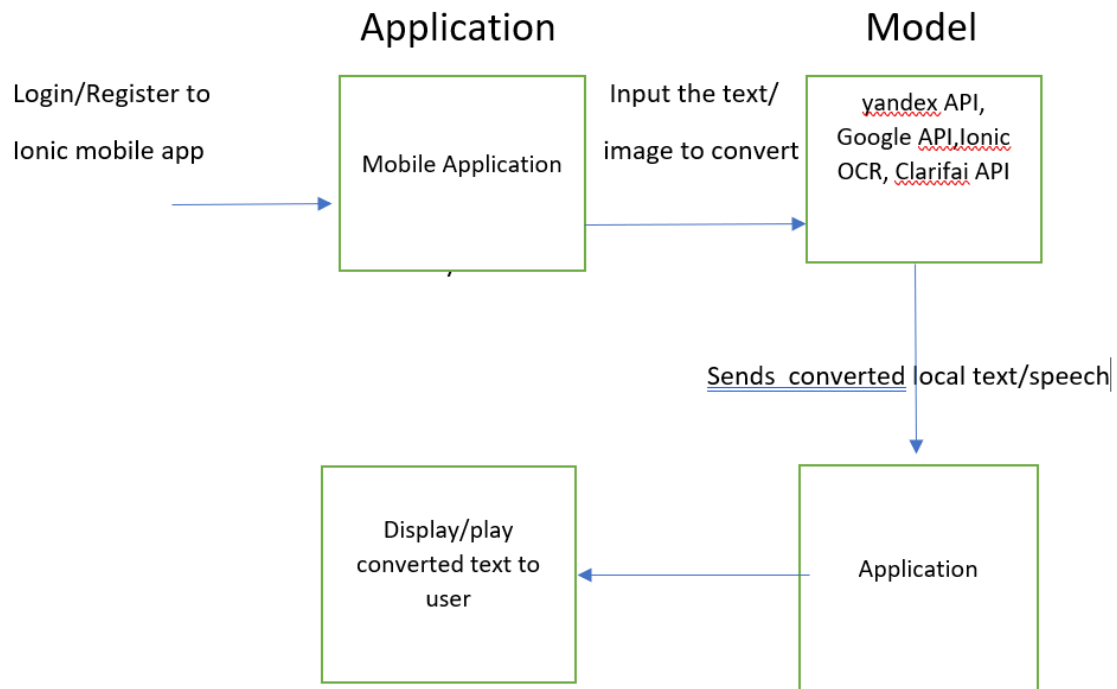
## 2. Sequence Diagram:



## 3. Iteration-2 Architecture Diagram:

Explore The World

### Architecture



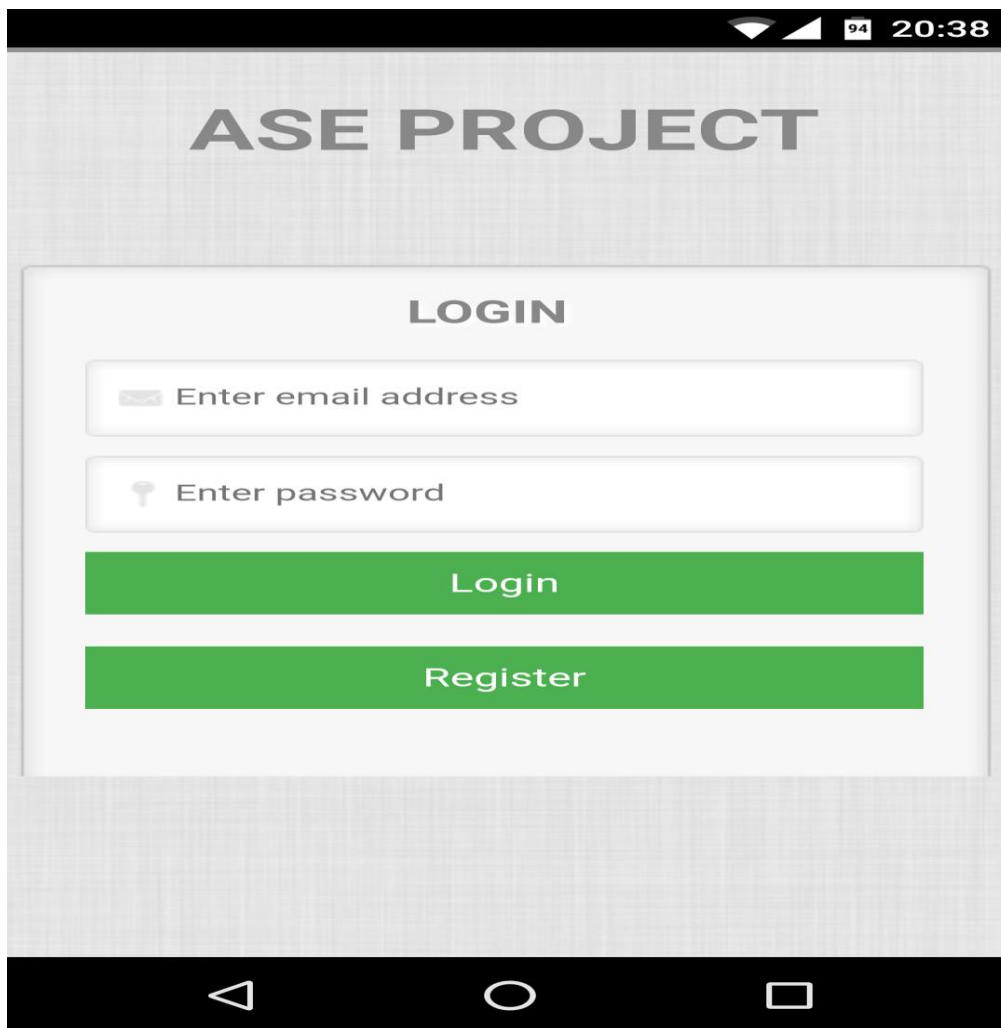
## Implementation:

Created an application using Ionic framework so that the application is runnable in any platform and user friendly. Application has a login page where user will be able to login and use the services provided in the application and user will be able to register and the details are stored in Firebase and email authentication is used for logging into the application.

After login into the application tourist will be able to use various services like text to text translation, image prediction, Google Maps and text to speech.

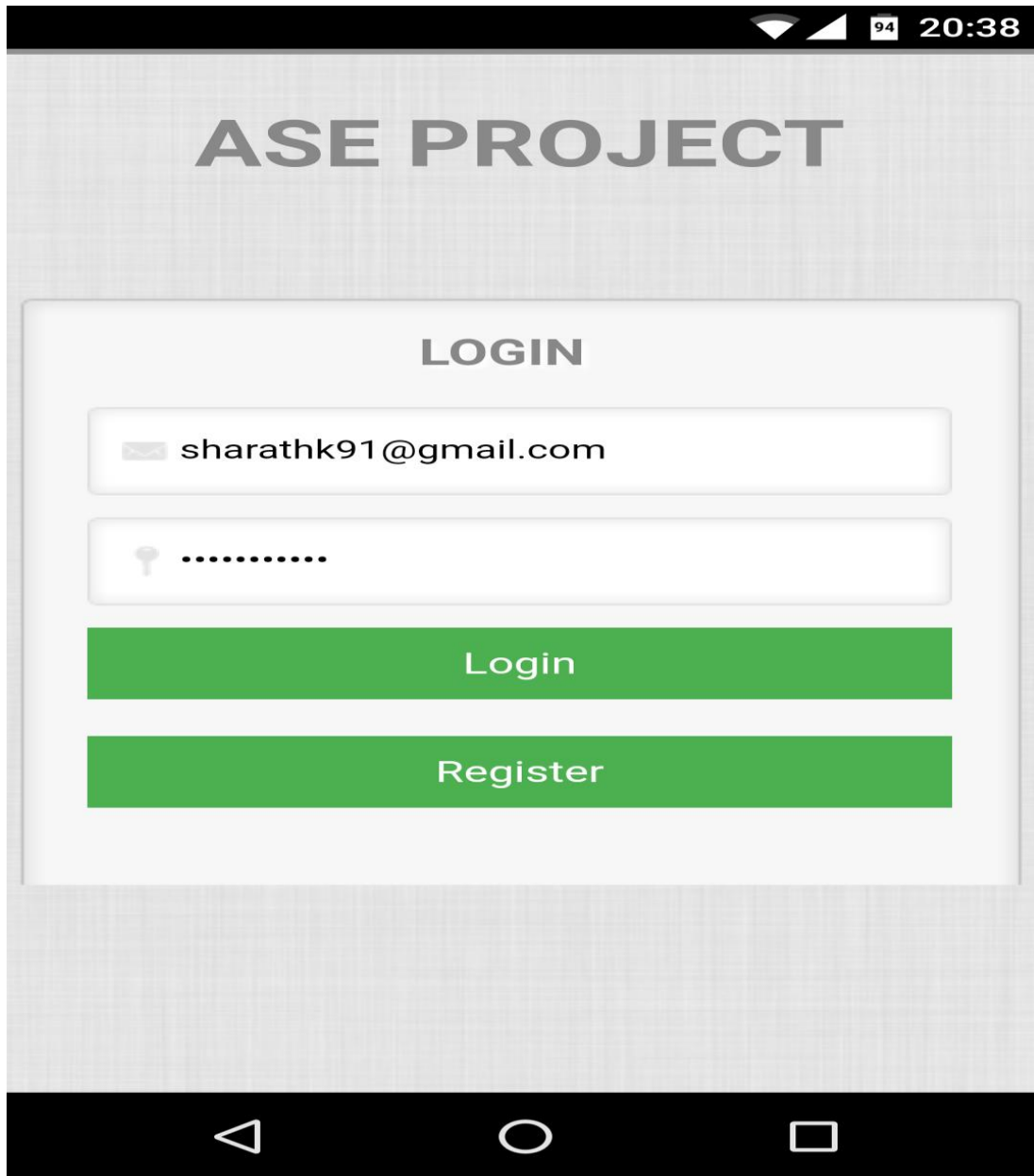
### 1. Login Screen:

User will be able to login into the application using email authentication provided by Firebase API.



## 2. Registration Screen:

Registration of a user is done by using by the same login screen, when user provides email and password and clicks register the details will be saved in the Firebase database and the data is fetched when user logins using the email and password set at the time of registration.



The screenshot shows a mobile application interface for 'ASE PROJECT'. At the top, there is a status bar with a Wi-Fi icon, a signal strength icon, a battery level of 94%, and the time 20:38. Below the status bar, the text 'ASE PROJECT' is displayed in a large, bold, grey font. In the center, there is a white rectangular box with a thin grey border. Inside this box, the word 'LOGIN' is written in bold, black, uppercase letters. Below 'LOGIN', there are two input fields. The first input field contains an email address 'sharathk91@gmail.com' preceded by an envelope icon. The second input field contains a password represented by a key icon and a series of dots. Below the input fields, there are two green buttons with white text. The top button is labeled 'Login' and the bottom button is labeled 'Register'. At the bottom of the screen, there is a black navigation bar with three white icons: a triangle pointing left, a circle, and a square.

## 3. Home Page:

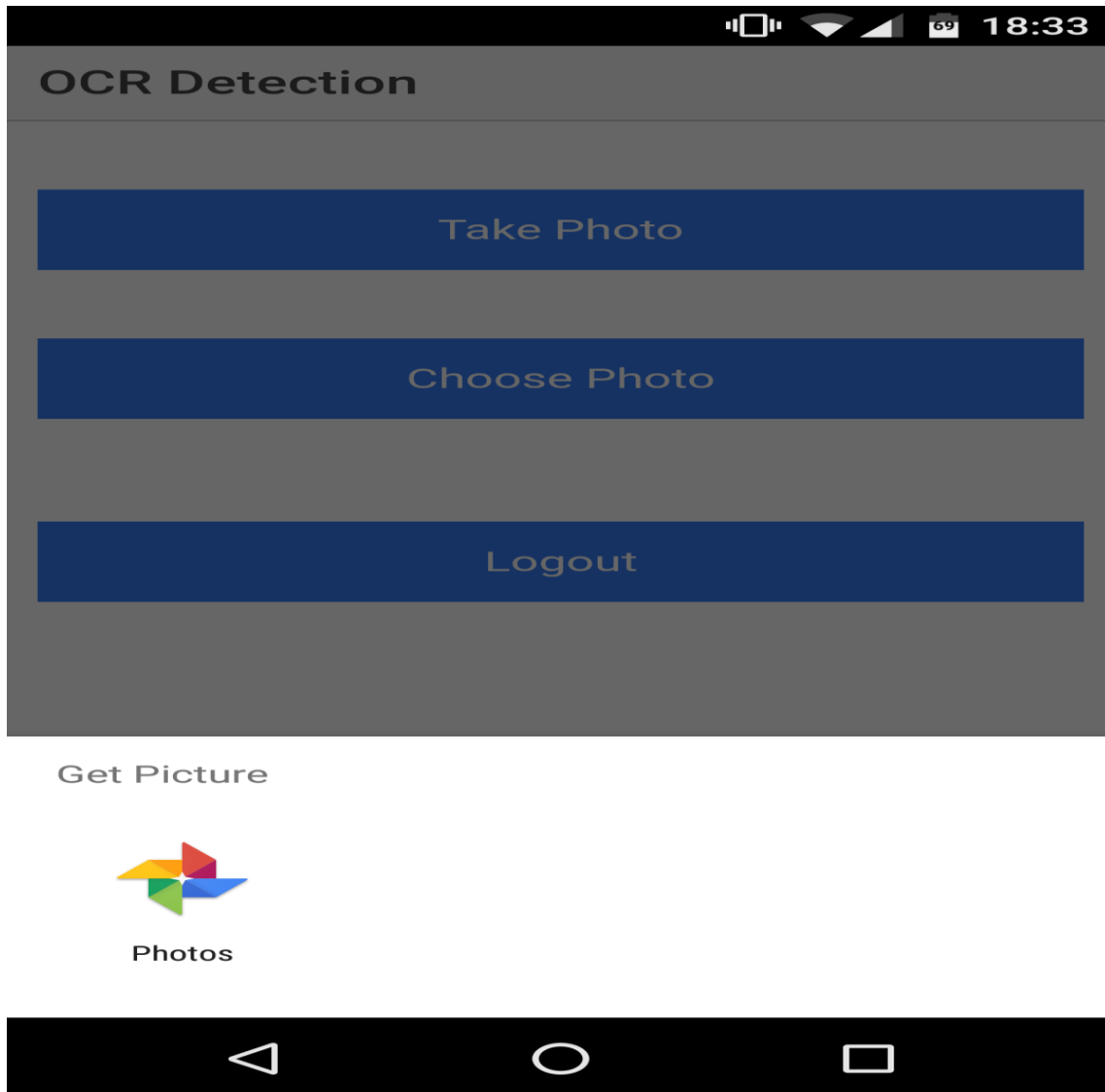


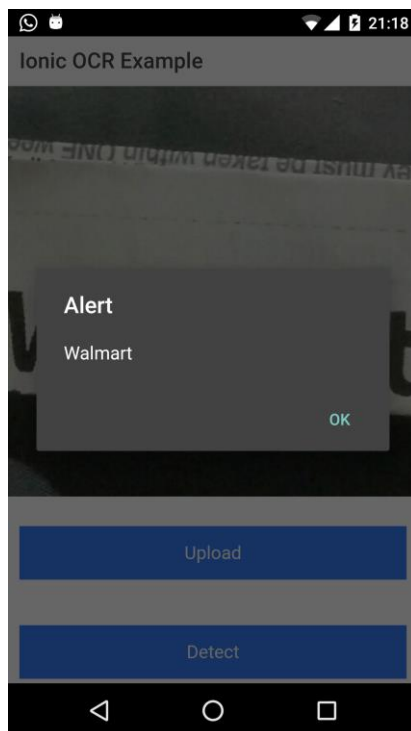
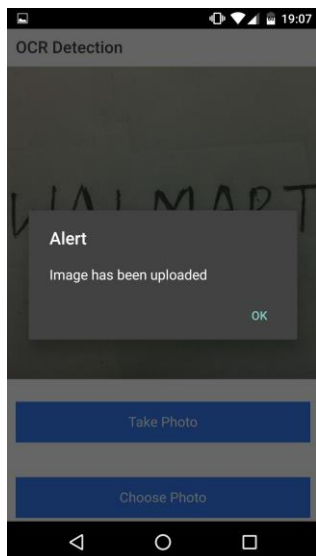
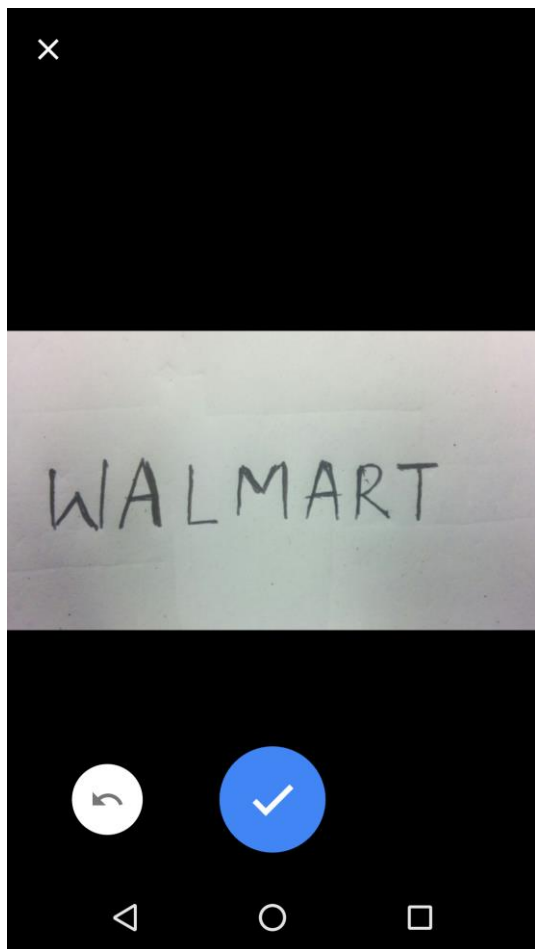
Home Page contains all the services of the application and tourist can use all the services that are required.



#### 4. OCR Plugin:

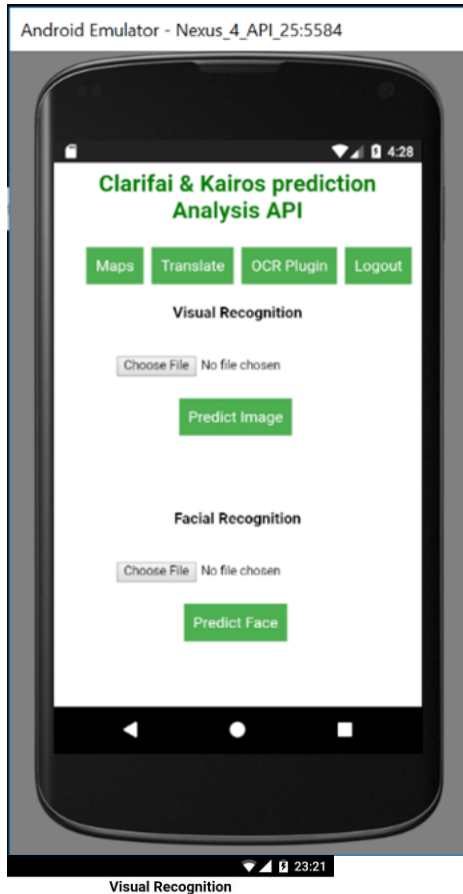
OCR plugin is used to extract text from an image so that tourist can take a picture of an image and convert the text in the image to editable text. User can take a picture using camera of the device and select an existing image so that he can extract meaningful text from the image.





## 5. Visual Recognition:

Visual Recognition is used to get meaningful insights from an image so that tourist will be able to get useful data from an image like a traffic signal which would be useful for better understanding of various situations. User has to choose a file from the device so that he can get insights from the image.



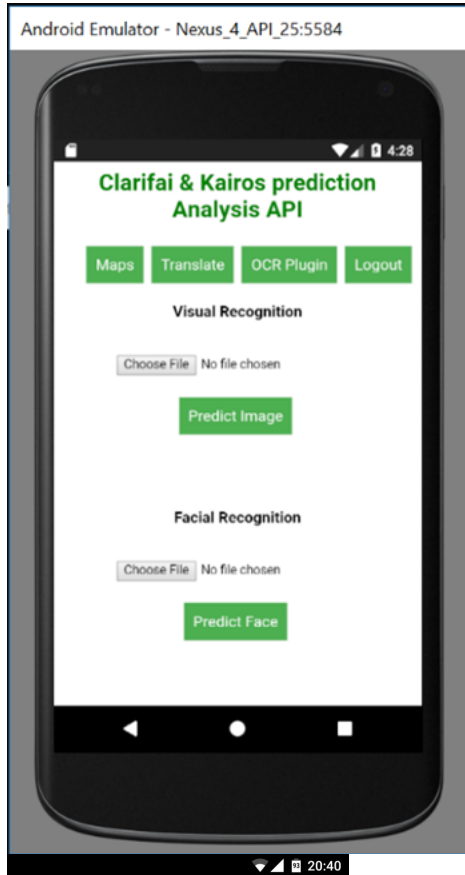
## 6. Translation

This tab is used to translate the text from tourist native language to the preferred language so that the barrier of communication can be overcome with this service in the application.



## 7. Maps:

Maps is used to locate the location of the tourist so that he can find and track the movements without opening any other application for his location.



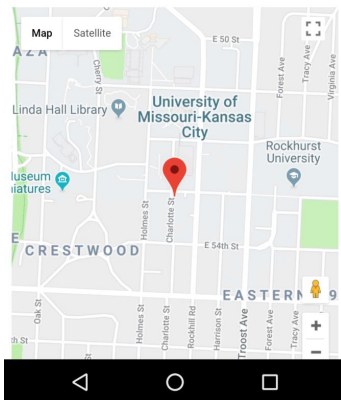
Location Details

Logout

Latitude: 39.0307332

Longitude: -94.5769098

Address: 803 E 53rd St, Kansas City, MO 64110, USA



## 8. Speech:

Speech tab is used to translate the text that is translated so that the local people can understand.



### Project Management: Implementation Status Report: Work Completed:

**1. Login Page:** Login Page is the first page of the application where user will be able to login with his username and password and if the user is new to the application he can register with Firebase email authentication.

#### Responsibility and Time Taken:

Aditya – Designed the Login and Registration Page (5 days) Sharath - Implemented the login and Registration page (5 days) Bhanu – Tested the login and Registration page (5 days)

**2. Text to Text Language Translation.** User will select the source language where he enters the text in the source language selected and he will select the language in which the entered text to be converted.

#### Responsibility and Time Taken:

Aravind – Design, Implementation and Testing of the translation page (5 days).

**3. Architecture and Wireframe Diagrams:** Describes the various functionality of the application.

**4. OCR Plugin: To extract editable text from an image.** Aditya – Designed the OCR and Camera plugins (5 days) Aravind - Implemented the OCR plugin (5 days) Bhanu – Tested OCR Plugin (5 days)

**5. Image Visualization:** Aditya – Designed the Visualization of Image Page (5 days) Sharath - Implemented and tested Image Visualization using Clarifai (13 days)

**6. Maps:** Aditya – Designed the Google maps page (5 days) Aravind - Implemented the Google Maps (5 days) Bhanu – Tested the Maps API (5 days)

**7. Language Conversion API's:** Sharath - Implemented, Designed and Tested Language Conversion API's (5 days).

**8. Text to Speech:** Aravind - Implemented, Designed and Tested text to speech API(8 days).

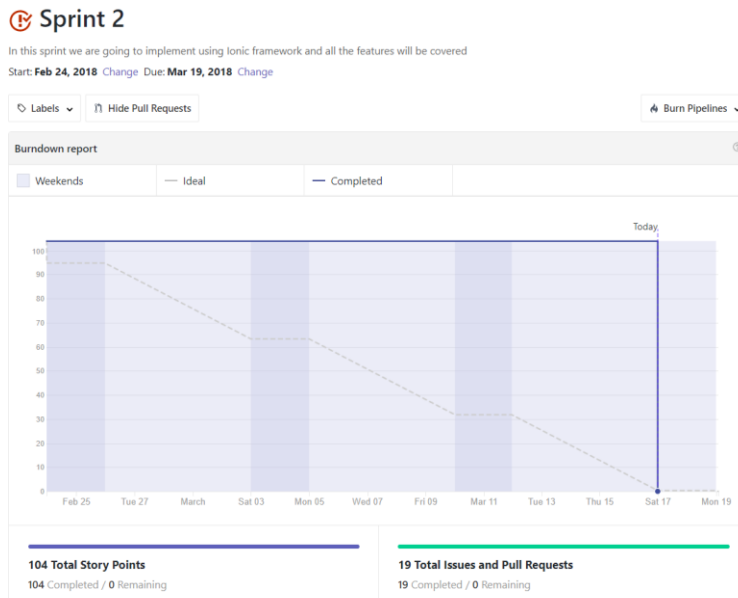
### Responsibility and Time Taken:

Sharath – Designed wireframes, architecture, blueprint and wiki page of the project (5 days)

Aravind – Created project Documentation.

**9. Class and Sequence Diagrams: Explains the functionality of the application.** Aditya - Designed the class and sequence diagrams.

### Sprint Burndown Chart:





## INCREMENT 3

### Services and API Used:

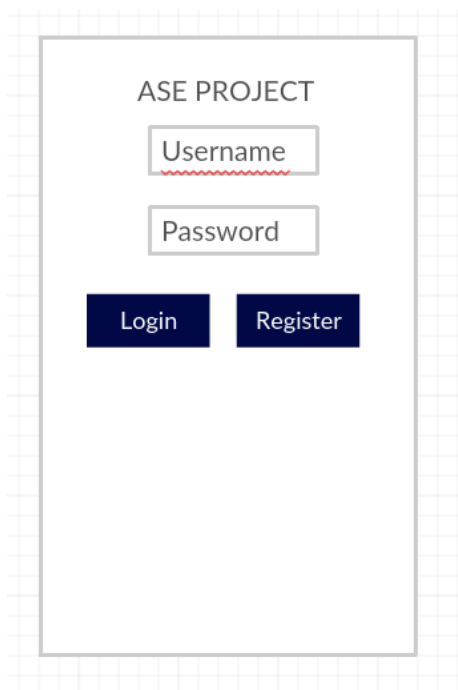
1. Yandex API for Text to Text language translation
2. Firebase for storing the details of the user and email authentication.
3. Google OCR plugin for extracting text from image.
4. Google maps API for maps and user location and navigation.
5. Geolocation for getting the user current location.
6. Clarifai for visual detection and getting insights from an image.
7. Google API for text to speech of the translated text.

### Detail Design of Features:

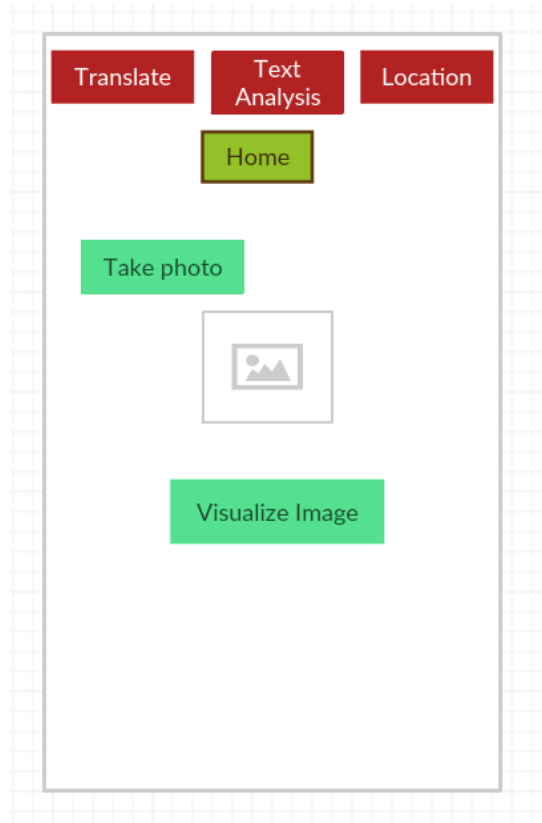
#### Wireframes and Mockups:

Below are the screenshots for the wireframes and Mockups:

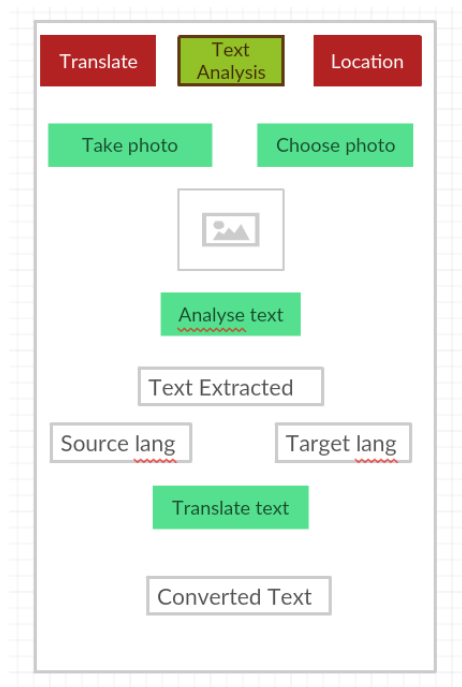
#### 1. Login Page



## 2. Home Page



## 3. OCR Plugin



#### 4. Translation:



A diagram of a translation interface. At the top, there are three buttons: 'Translate' (green), 'Text Analysis' (red), and 'Location' (red). Below these are two input fields: 'Source lang' and 'Target lang', both with red wavy lines underneath. In the center is a 'Source text' input field. Below that is a red 'Translate' button. At the bottom is a 'Converted text' output field.

#### 5. Maps

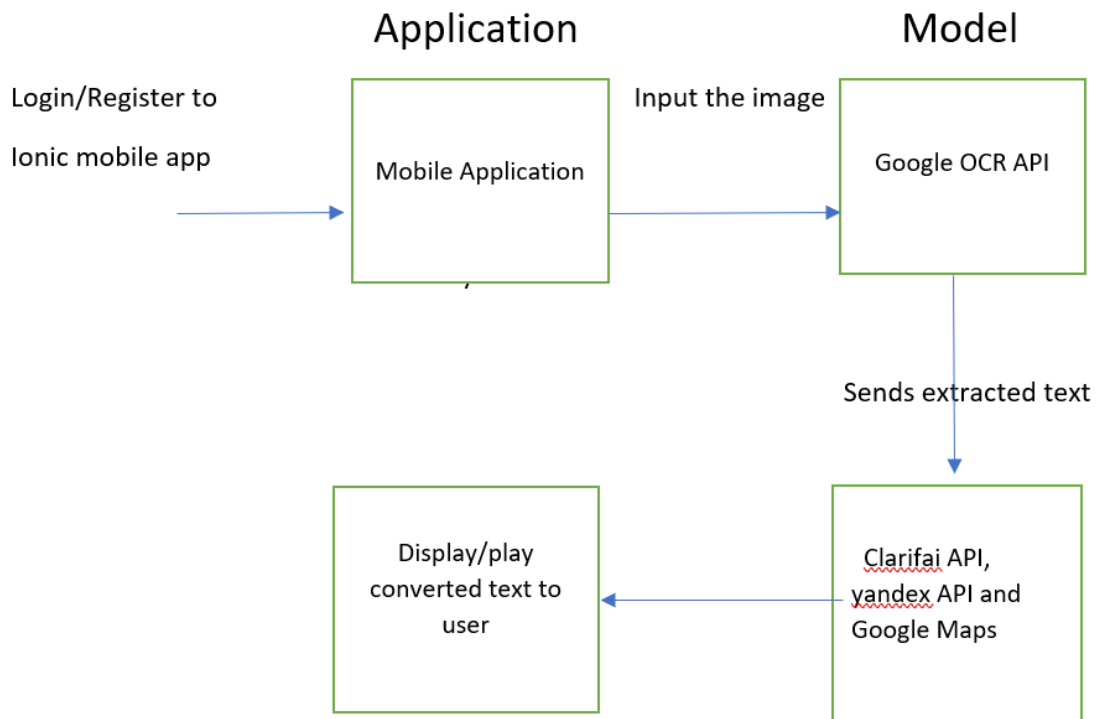


A diagram of a maps interface. At the top, there are three buttons: 'Translate' (red), 'Text Analysis' (red), and 'Location' (green). Below these are three input fields: 'Latitude', 'Longitude', and 'Street Address'. Below the input fields is a label 'Map' and a large empty rectangular box for the map. At the bottom is a red 'Logout' button.

## Architecture Diagrams

Explore The World

### Architecture

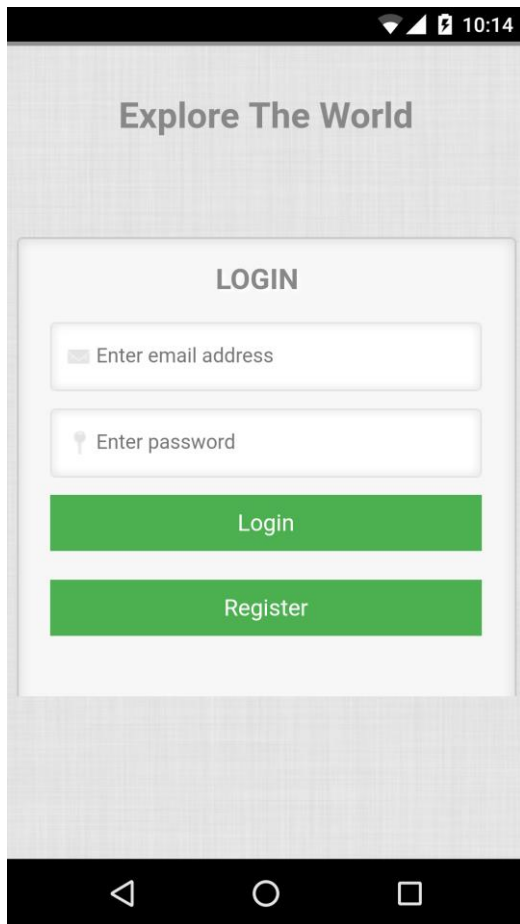


### Implementation:

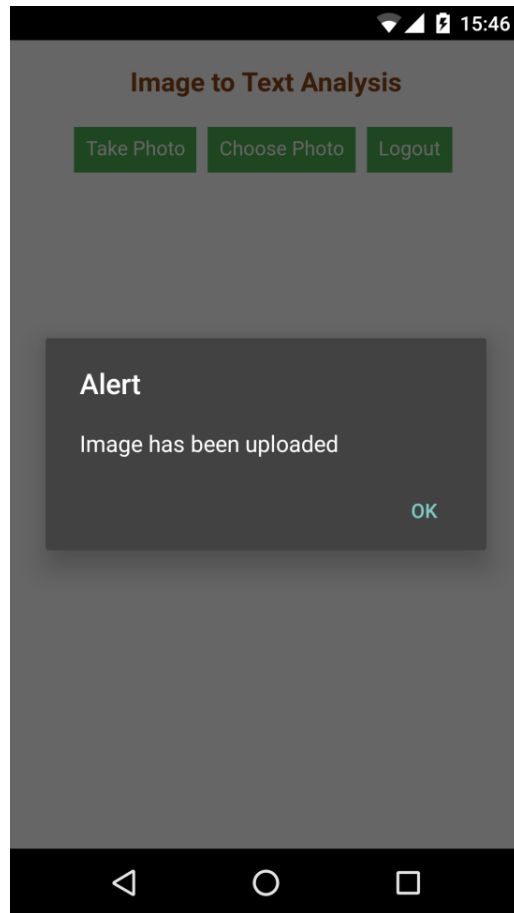
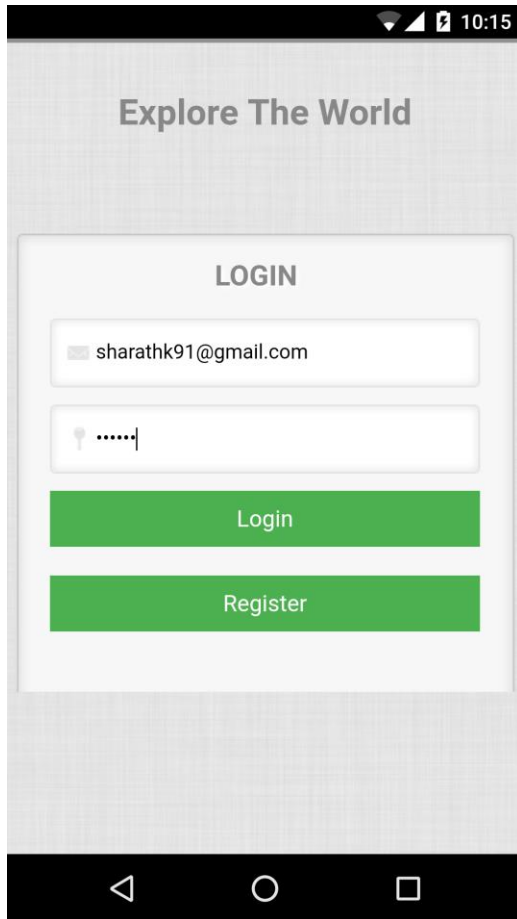
Created an application using Ionic framework so that the application is runnable in any platform and user friendly. Application has a login page where user will be able to login and use the services provided in the application and user will be able to register and the details are stored in Firebase and email authentication is used for logging into the application.

After login into the application tourist will be able to use various services like text to text translation, image prediction, Google Maps and text to speech

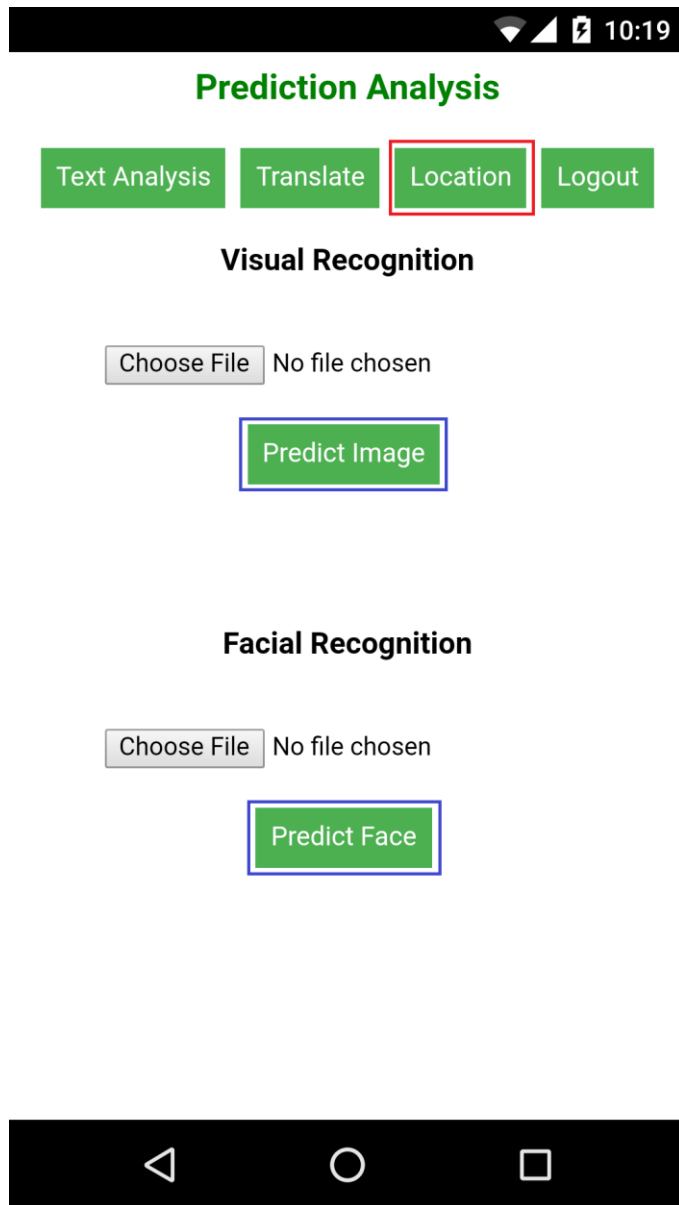
**1. Login Screen:** User will be able to login into the application using email authentication provided by Firebase API.



**2. Registration Screen** Registration of a user is done by using by the same login screen, when user provides email and password and clicks register the details will be saved in the Firebase database and the data is fetched when user logs in using the email and password set at the time of registration.

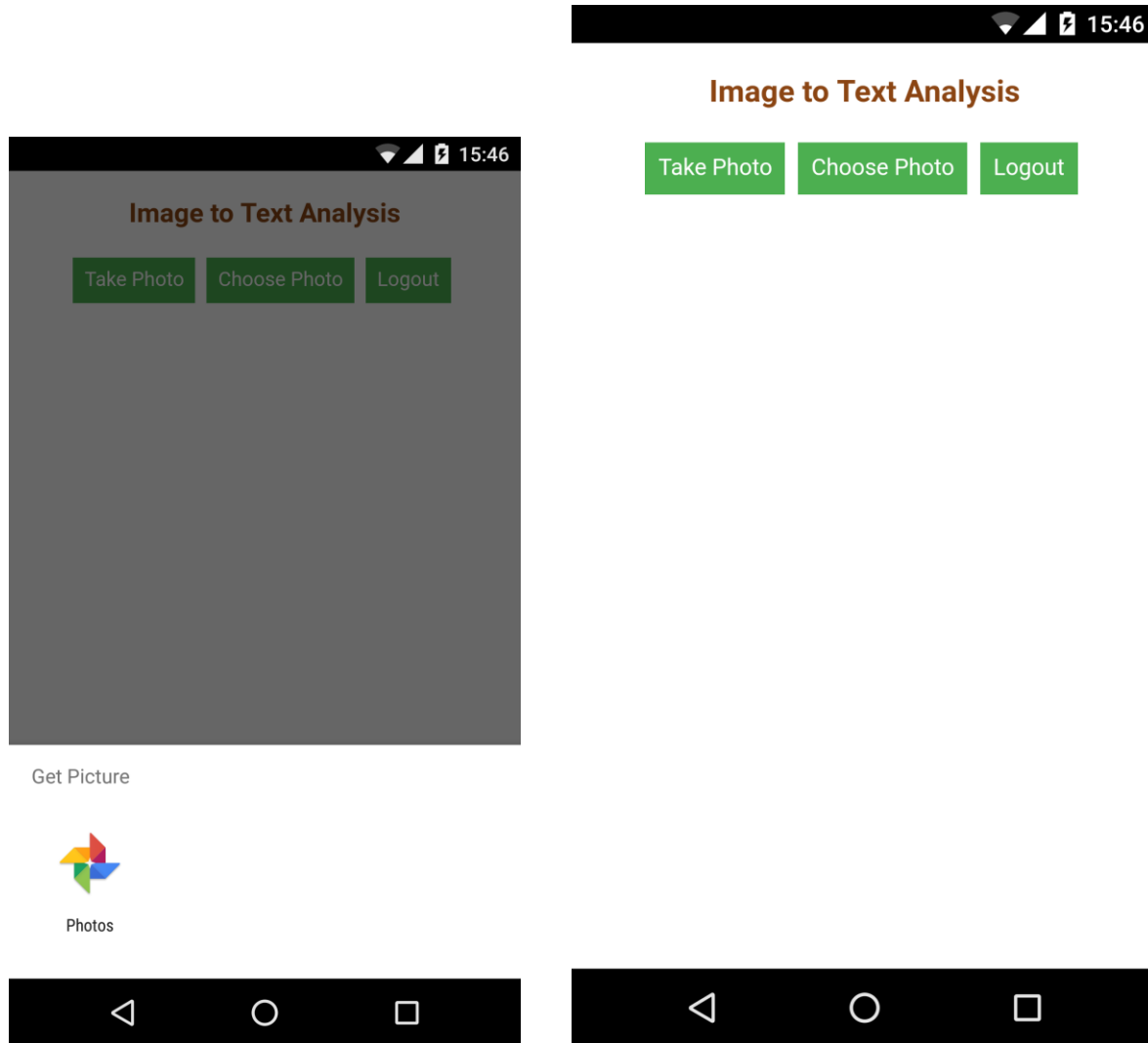


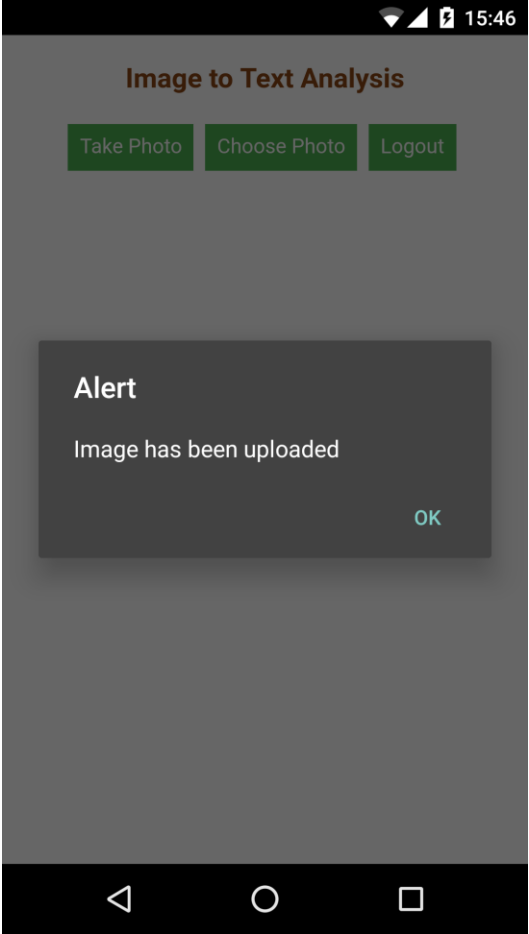
**3. Home Page** Home Page contains all the services of the application and tourist can use all the services that are required.

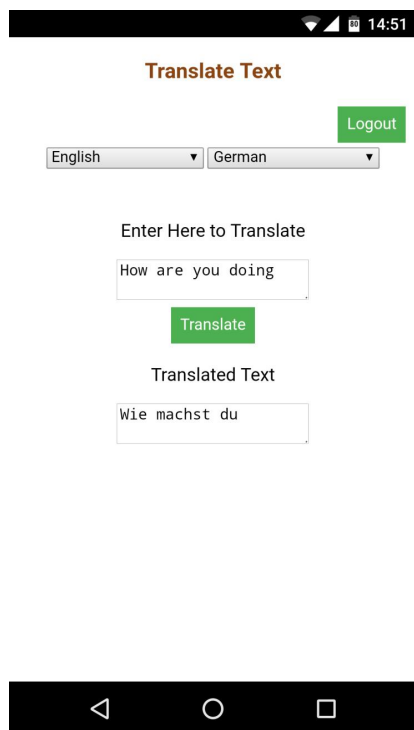
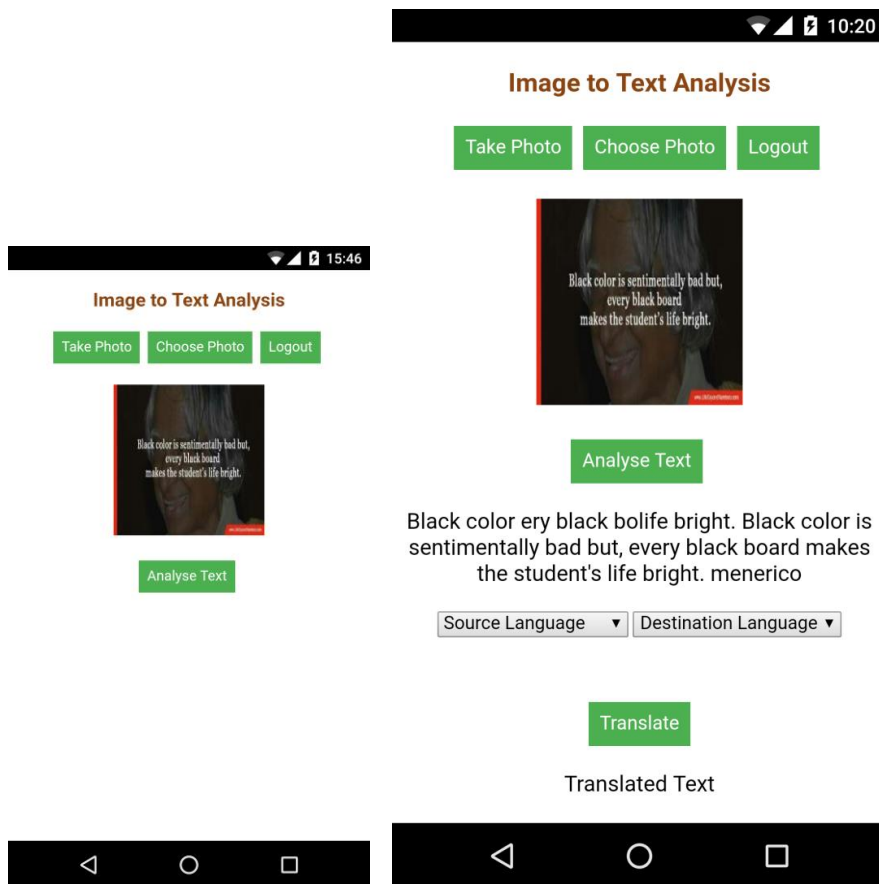




**4. OCR Plugin** OCR plugin is used to extract text from an image so that tourist can take a picture of an image and convert the text in the image to editable text. User can take a picture using camera of the device and select an existing image so that he can extract meaningful text from the image.







## Location Tab:



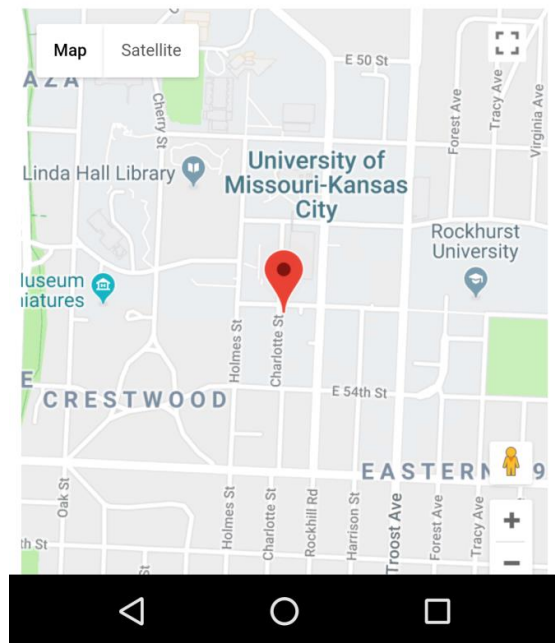
### Location Details

Logout

**Latitude:** 39.0307332

**Longitude:** -94.5769098

**Address:** 803 E 53rd St, Kansas City, MO 64110, USA



**8. Speech:** The Speech tab is used to translate the text that is translated so that the local people can understand.

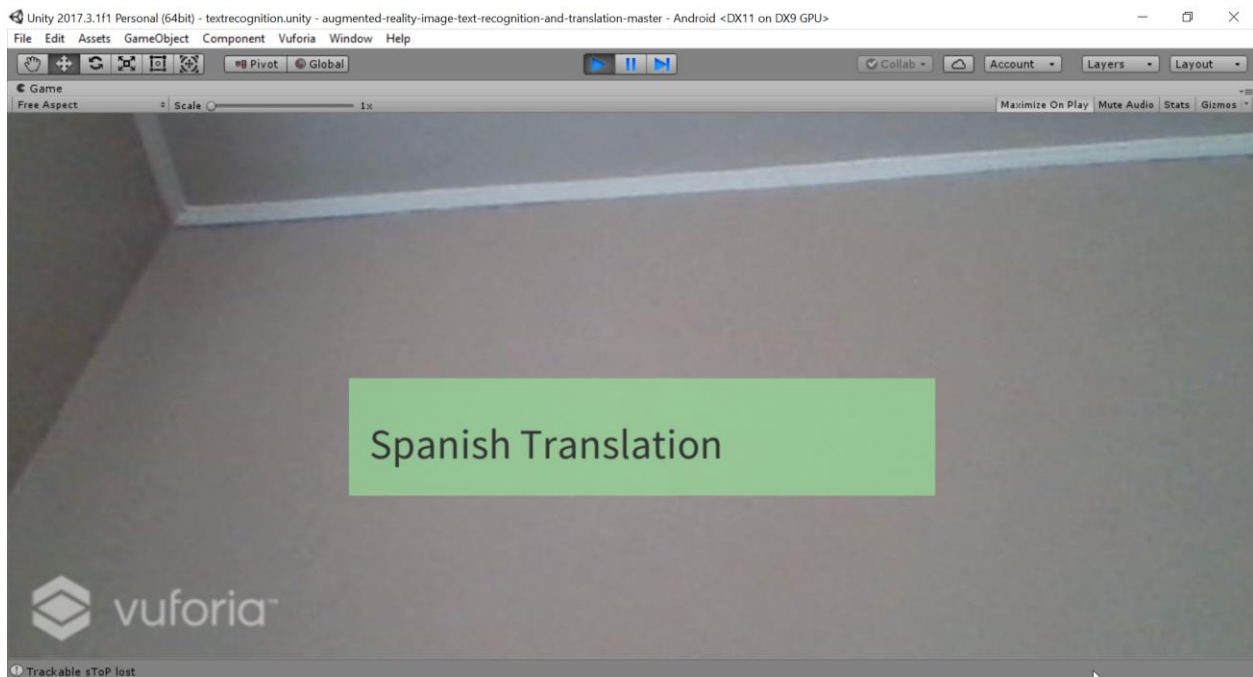
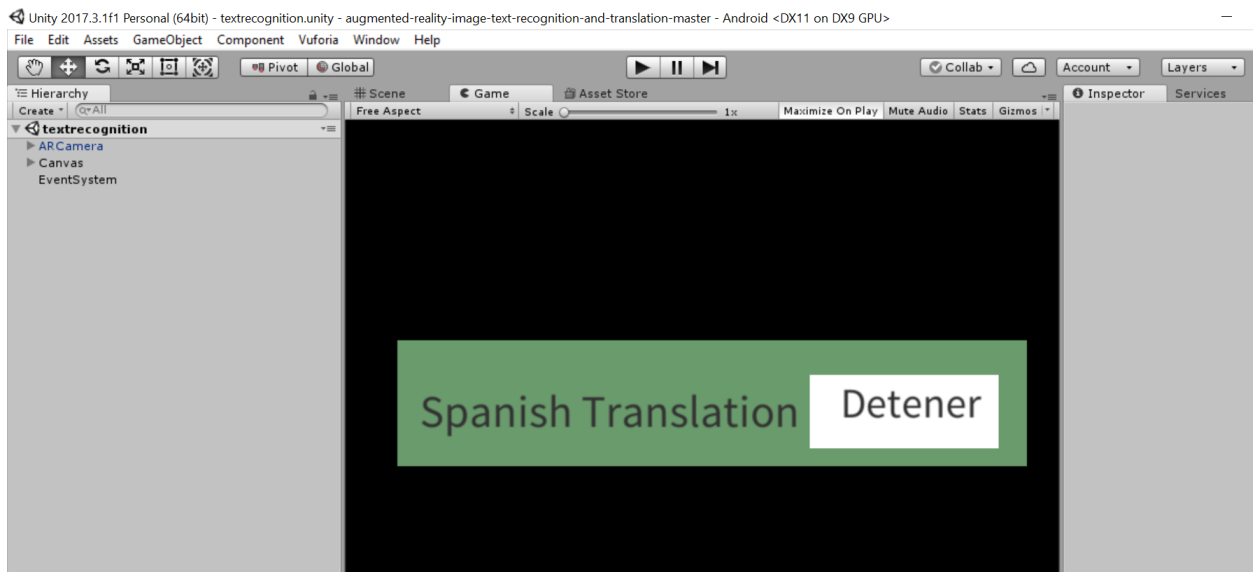


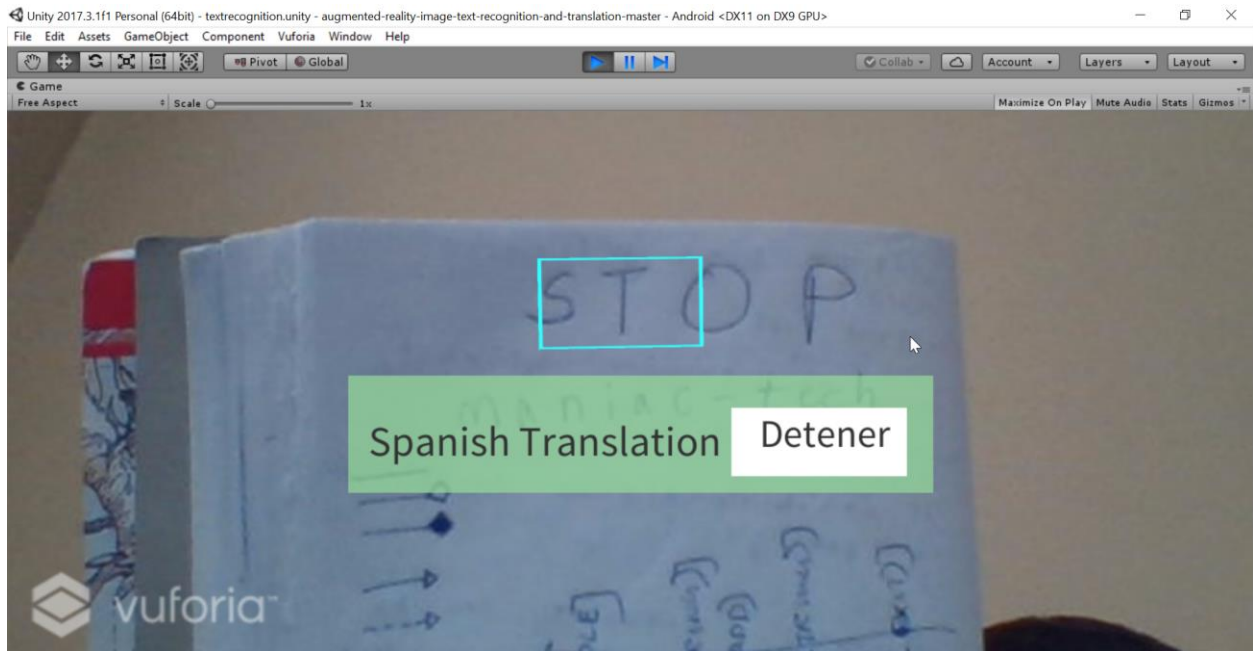
**Heroku Cloud Deployment** The project has been deployed into the Heroku cloud server

**Heroku Cloud URL** [Heroku deployment](#)

## Augmented Reality:

Designed, Developed and implemented the Augmented Reality application with Vuforia and unity. This application will detect the sign posts and convert into the desired language. We have trained the system to detect the word “STOP” and convert into Spanish language





## Testing

### Unit Testing

Each and every feature has been validated and is working as expected

**Performance Testing** The performance of the entire project has been evaluated using Yslow tool which has described below

chrome-extension://ninejcohidippngpapiilmkgllmkh/yslow.html#1

Home **Grade** Components Statistics Rulesets **YSlow(V2)** Edit Help

**Grade C** Overall performance score 77 Ruleset applied: YSlow(V2) URL: https://hidden-garden-97186.herokuapp.com/index.html

ALL (23) FILTER BY: CONTENT (6) | COOKIE (2) | CSS (6) | IMAGES (2) | JAVASCRIPT (4) | SERVER (6)

**C Make fewer HTTP requests**

- F Use a Content Delivery Network (CDN)
- A Avoid empty src or href
- F Add Expires headers
- F Compress components with gzip
- A Put CSS at top
- E Put JavaScript at bottom
- A Avoid CSS expressions
- n/a Make JavaScript and CSS external
- A Reduce DNS lookups
- A Minify JavaScript and CSS
- A Avoid URL redirects
- A Remove duplicate JavaScript and CSS
- A Configure entity tags (ETags)
- A Make AJAX cacheable
- A Use GET for AJAX requests
- A Reduce the number of DOM elements
- A Avoid HTTP 404 (Not Found) error
- A Reduce cookie size
- A Use cookie-free domains

**Grade C on Make fewer HTTP requests**

This page has 9 external Javascript scripts. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[Read More](#)

Copyright © 2018 Yahoo! Inc. All rights reserved.

## **Project Management**

### **Implementation Status Report**

#### **Work Completed:**

**1. Google OCR** In the previous iteration we have used Ionic OCR to convert image to text but due to low accuracy, in this iteration we have used Google OCR to convert images to text.

#### **Responsibility and Time Taken**

Aravind – Designed, Implemented and tested the OCR image conversion Page (24 days)

#### **2. Augmented Reality Application**

Developed Augmented Reality application with Vuforia and Unity. This story should be able to detect the signs and convert them into desired language.

**Responsibility and Time Taken** Sharath – Design, Implementation and Testing of the translation page (24 days)

**3. Integrating Augmented Reality with Ionic** Describes the integration of Augmented Reality Application with Ionic Framework. Aditya - Design, Implementation and Testing the integration of Augmented Reality (24 days)

**4. Unit & Sanity Testing** Bhanu – Validated all sanity and unit test cases for the entire application (16 days)

#### **Concerns**

We are facing issues while integrating the Vuforia-Unity application with Ionic framework

#### **Github References**

[Vuforia unity Integration](#)

#### **Future Work**

Analysing Twitter Social Media content is the pending story and will be completed in future.



## Sprint Burndown Chart

### Sprint 3

In this we are going to handle the voice to voice API conversions from different languages , unit test each case, make a kind of regression,blackbox and whitebox testing

Start: **Mar 20, 2018** [Change](#) Due: **Due today** [Change](#)

 Labels ▾


 Hide Pull Requests

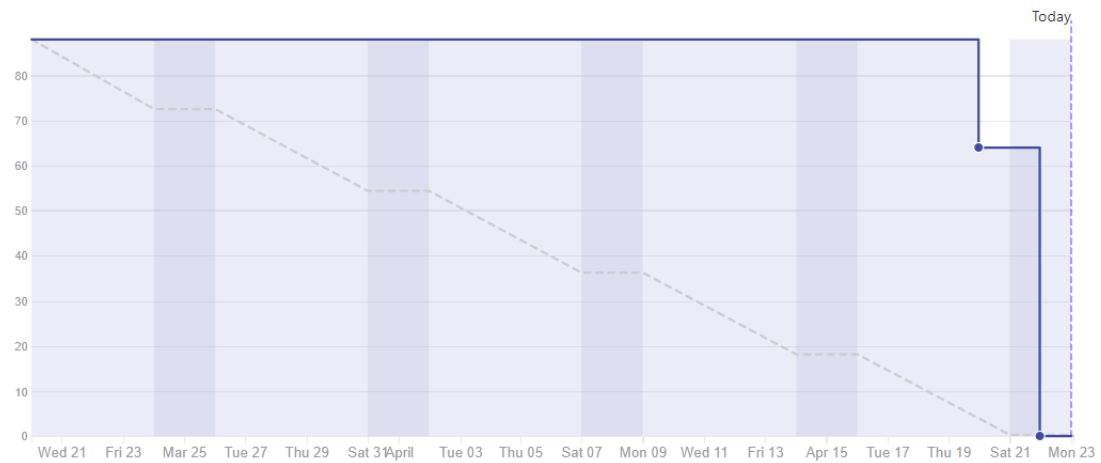
 Burn Pipelines ▾

#### Burndown report

 Weekends

 Ideal

 Completed



**88 Total Story Points**

88 Completed / 0 Remaining

**11 Total Issues and Pull Requests**

11 Completed / 0 Remaining

# Explore the World

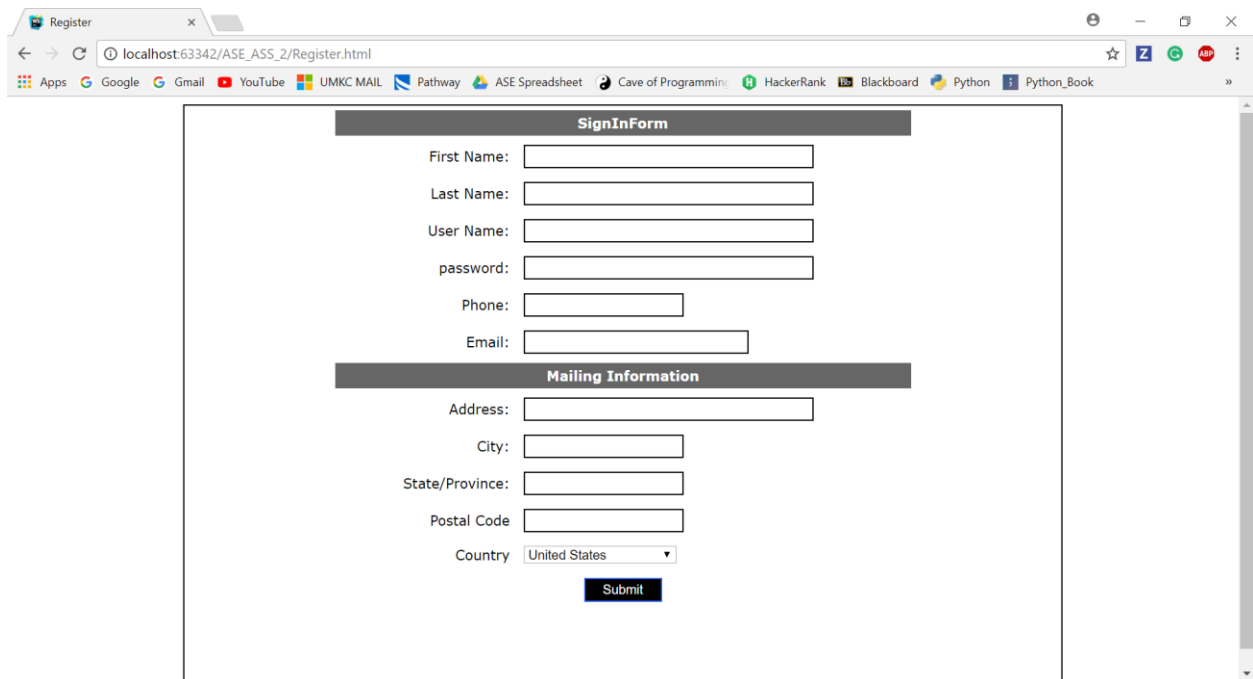
*User Manual*

# 1 Introduction

Android/iOS application that helps in travel and tourism.

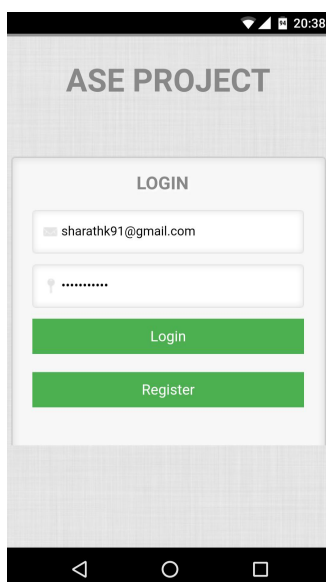
## 1.1 User Login: Using Registered Email Address or Google Authentication

### 1.1.1 New users register their email address.



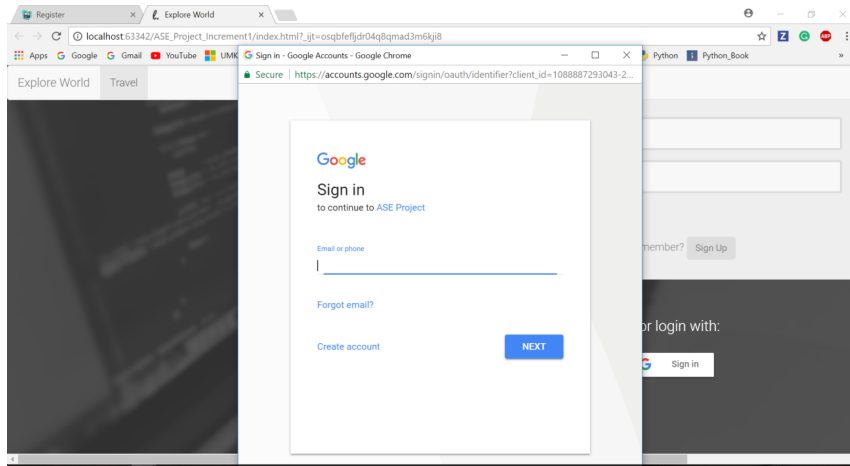
The screenshot shows a web browser window with the address bar displaying 'localhost:63342/ASE\_ASS\_2/Register.html'. The browser's taskbar at the top includes icons for Apps, Google, Gmail, YouTube, UMKC MAIL, Pathway, ASE Spreadsheet, Cave of Programming, HackerRank, Blackboard, Python, and Python\_Book. The main content area contains a registration form with two sections: 'SignInForm' and 'Mailing Information'. The 'SignInForm' section includes input fields for First Name, Last Name, User Name, password, Phone, and Email. The 'Mailing Information' section includes input fields for Address, City, State/Province, Postal Code, and a Country dropdown menu set to 'United States'. A 'Submit' button is located at the bottom of the form.

### 1.1.2 Registered Users login using their registered email id.



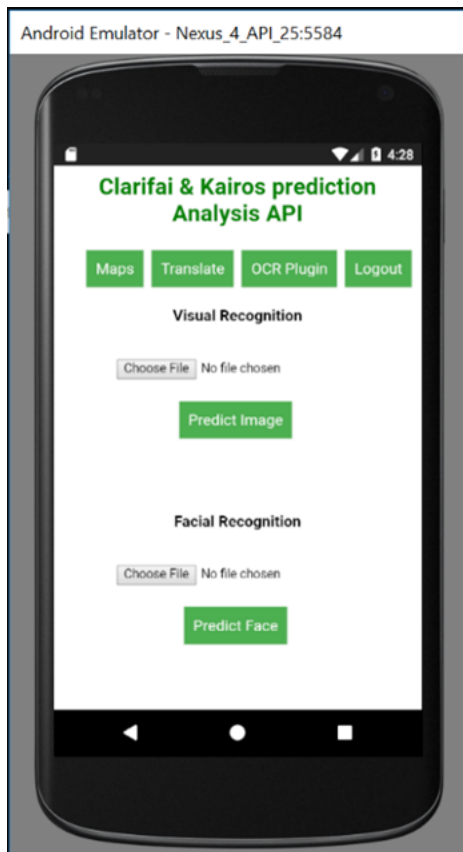
The screenshot shows a mobile application interface for 'ASE PROJECT'. The screen displays a 'LOGIN' section with two input fields: one for the email address (containing 'sharathk91@gmail.com') and one for the password (masked with dots). Below the input fields are two green buttons: 'Login' and 'Register'. The top status bar shows the time as 20:38 and various icons. The bottom navigation bar shows standard Android navigation icons.

## 1.1.2 Google Login



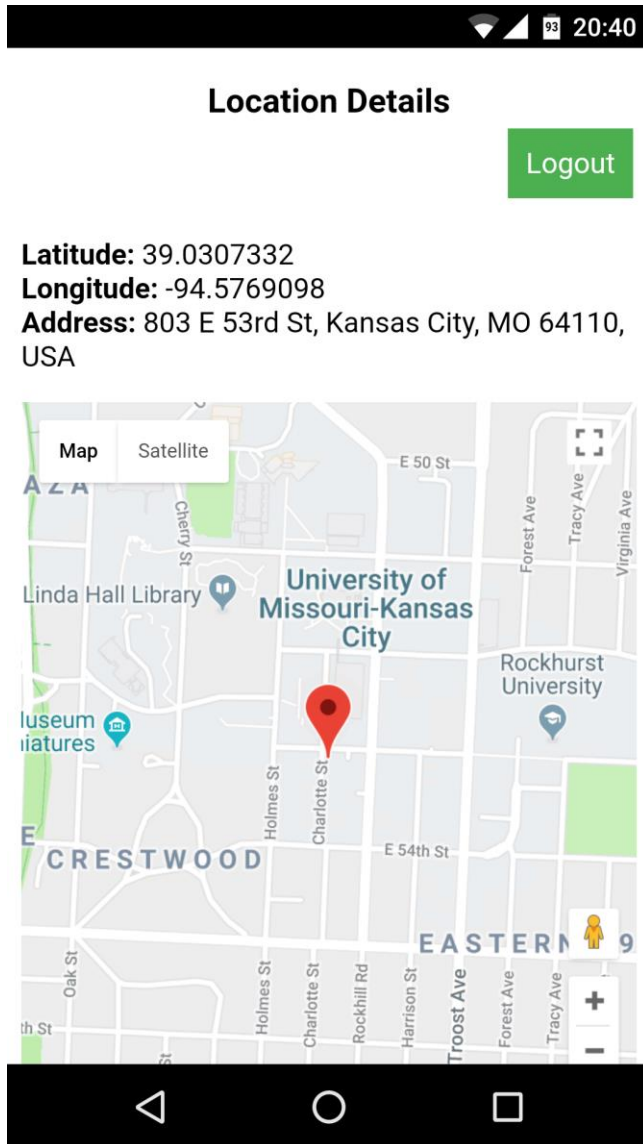
## 2 Home Page

**Home Page has various links to navigate the application**



### 3 Location Services and Maps

**Gets the current location and address of the user.**




## 4 Image to Text Analysis

### Image to Text Analysis Page



## 5 Translation/Analysis

### Image Analysis




### Image to Text Analysis

Take Photo

Choose Photo

Logout



Analyse Text


Black color ery black bolife bright. Black color is sentimentally bad but, every black board makes the student's life bright. menerico

Source Language ▼

Destination Language ▼


Translate

Translated Text



## 6 Language Translation

### Language translation

14:51

### Translate Text

Logout

English ▼German ▼


Enter Here to Translate

How are you doing

Translate

Translated Text

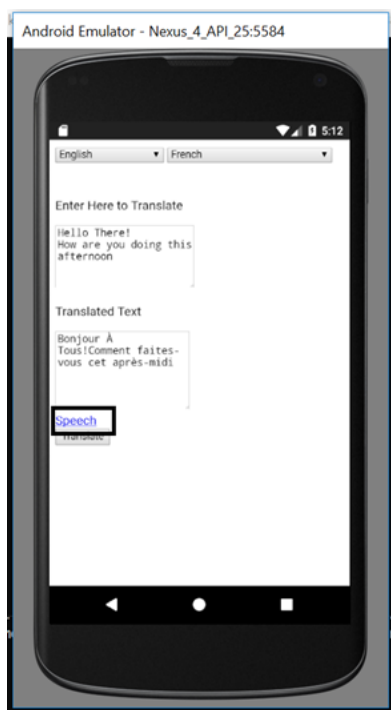
Wie machst du





## 7 Speech To Text/ Text to Speech

### Convert text to speech



# Acknowledgement

The work has been completed under the guidance of Dr. Yugi Lee and TAs (Rohith Nagulapati, Sidrah Junaid, Nageswara Nandigam) in CS5551 Advanced Software Engineering, University of Missouri -Kansas City), Spring 2018.