****

**Project Report**

\

**For**

**MAP-IN APP**

**Version-1.0**

**Developed By**

**Adnan Jafri**

B. Tech, 4th year, Computer Science & Engineering

Aryan Institute of Technology, Ghaziabad

Email: [adnan.jafri7@gmail.com](mailto:adnan.jafri7@gmail.com)

**Supervisor**

**Khushboo Mirza**

Scientist/Engineer ‘SF’

Regional Remote Sensing Centre - North

National Remote Sensing Centre

Indian Space Research Organization (ISRO)

Department of Space, Government of India, New Delhi

**Table of Contents**

[1.0 Introduction 2](#_Toc120614432)

[1.1 Project 2](#_Toc120614433)

[1.2 Scope 2](#_Toc120614434)

[1.3 Technologies Used 2](#_Toc120614435)

[1.3.1 React Native (version 0.64.2) 2](#_Toc120614436)

[1.3.2 Node.js (version 14.17.3) 3](#_Toc120614437)

[1.3.3 JDK (version 8 or above) 3](#_Toc120614438)

[1.3.4 Android Studio (version 4.2.2) 3](#_Toc120614439)

[1.3.5 pgSQL (version 15) 3](#_Toc120614440)

[1.3.6 pgAdmin 4](#_Toc120614441)

[1.3.7 PHP (version 8.0.9) 4](#_Toc120614442)

[1.3.8 XAMPP Server (version 3.3.0) 4](#_Toc120614443)

[2.0 Functional Requirement 5](#_Toc120614444)

[2.1 Use Cases 5](#_Toc120614445)

[3.0 Data Description 12](#_Toc120614446)

[3.1 Data Flow Diagram 12](#_Toc120614447)

[3.2 Flow Charts 13](#_Toc120614448)

[3.3 Database Description 14](#_Toc120614449)

[3.3.1 Table ‘users’ 14](#_Toc120614450)

[3.3.2 Table ‘mapin’ 15](#_Toc120614451)

[4.0 User Manual 16](#_Toc120614452)

[4.1 Welcome Screen 16](#_Toc120614453)

[4.2 Register 17](#_Toc120614454)

[4.3 Login 19](#_Toc120614455)

[4.4 Map View 20](#_Toc120614456)

[4.5 Attribute Form 23](#_Toc120614457)

[4.6 History 24](#_Toc120614458)

[4.7 Show on Map 25](#_Toc120614459)

[4.8 User Profile 26](#_Toc120614460)

[4.9 About App/Support Screen 27](#_Toc120614461)

[5.0 Definitions and Acronyms 28](#_Toc120614462)

[6.0 References 29](#_Toc120614463)

# Introduction

## Project

**Map-In** is an interactive Mobile App for collecting Geo Spatial Data for field surveys. Map-In provides multi-purpose mapping solutions like location capture, drawing shapes (Point, Polygon, and Lines) using Geometry tools available on the App. Map-In is a real time location marking app used for field surveys.

## Scope

Geographical Information System (GIS) is an information technology system consisting of hardware, software, database, and spatial data. GIS can help to manage various data formats that contain graphics, layers, symbols, and attributes. Moreover, with the growth of the Global Positioning System (GPS), Location-based Service (LBS), and wireless communication, GIS has developed from desktop software to mobile applications. Map-In uses GIS and GPS for collecting the location data from the field. Map-In is built on the latest GPS technology which provides real time location data.

There are various network issues in remote areas, due to network issues people who do the Geo Location based surveys face problems in surveying. Map-In provides offline data collection which will be very useful for Field Surveys. If there’s a low network, then the user can mark the location and save the location data in the phone storage and later when the user comes in better network, he/she can upload that data to server. Map-In also provides an Export functionality, in which a user can export the saved data to an Excel File.

## Technologies Used

### [React Native](https://reactnative.dev/) (version 0.64.2)

React Native is a JavaScript framework used for developing a real, native mobile application for iOS and Android. It uses only JavaScript to build a mobile application. It is like React, which uses native components rather than using web components as building blocks. React Native apps are not web applications. They are running on a mobile device, and it does not load over the browser. React Native apps are the real native app, the JavaScript code stays as JavaScript, and they run in some extra thread by the compiled app. The user interface and everything is compiled to native code.

The entire MAP-IN application is written in react native.

### [Node.js](https://nodejs.org/en/) (version 14.17.3)

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. It is designed to build scalable network applications.

The app requires NPM (version 6.14.14) for installing the dependencies.

## 

### [JDK](https://openjdk.java.net/projects/jdk8/) (version 8 or above)

JDK is required for developing apps using React Native CLI.

## 

### [Android Studio](https://developer.android.com/studio/) (version 4.2.2)

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, designed specifically for Android development.

It also provides an emulator (a virtual device) on which the app can be run, thus helping in debugging.

## 

### [pg](https://www.postgresql.org/)SQL (version 15)

pgSQL is a free and open-source relational database management system (RDBMS) emphasizing extensibility and SQL compliance.

This app uses pgSQL database to store user information, marked locations, and location information.

### pgAdmin

pgAdmin is a GUI for interacting with the pgSQL database.

## 

### [PHP](https://www.php.net/) (version 8.0.9)

PHP is a general-purpose scripting language geared towards web development. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon, or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code forms the whole or part of an HTTP response.

PHP is used for connecting the react-native frontend with the MySQL database of the app.

## 

### [XAMPP Server](https://www.apachefriends.org/download.html) (version 3.3.0)

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

# Functional Requirement

## Use Cases

|  |  |  |
| --- | --- | --- |
| **Use Case 1: REGISTER** | | |
| * Actors | : | User |
| * Type | : | Primary and Essential |
| * Pre-condition | : | None |
| * Main Case Scenario | : | * User enters personal information (name, mobile number, email, street address, city, state) to create a new account. * On submitting, the system creates a user account and stores the required information into the database. |
| * Exception Scenario | : | * User inputs a mobile number containing less than 10 digits. The user is requested to provide the correct information to create an account. * User leaves either of the necessary fields - name, mobile number, email, street address, city, state - empty. The user is requested to provide all the details to create an account. * The system is unable to connect to the database. ‘Server Error’ message is displayed, and the user is requested to try again. |

|  |  |  |
| --- | --- | --- |
| **Use Case 2: LOG IN** | | |
| * Actors | : | User |
| * Type | : | Primary and Essential |
| * Pre-condition | : | User already has an account registered by their mobile number |
| * Main Case Scenario | : | * User enters the registered mobile number to log into the account. * The system checks for an account with the provided mobile number in the database and fetches the account details. * User successfully logs into their account. * User’s fetched details has been saved into the local database. |
| * Exception Scenario | : | * The mobile number entered contains less than 10 digits. The user is requested to enter the correct mobile number. * No account exists with the entered mobile number. The user is requested to create a new account. * The system is unable to connect to the database. ‘Server Error’ message is displayed, and the user is requested to try again. |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **Use Case 3: MAP** | | |
| * Actors | : | User and System |
| * Type | : | Primary and Essential |
| * Pre-condition | : | User has already logged into their account. |
| * Main Case Scenario | : | * The system fetches the Map and shows the Map to the user. * User will draw the Point, Polygon, or Line using the draw tool available on the Map screen. * User can change the Map Layer type to Bhuvan, Satellite, or OSM. * After drawing, User will be taken to the Form Screen where User must fill the details of the marked location and save it. |
| * Exception Scenario | : | * Due to low network connectivity Map has not been loaded and a white screen has been displayed. * User denied the location permission due to which user location has not been displayed on the Map. |

|  |  |  |
| --- | --- | --- |
| **Use Case 4: FORM SCREEN** | | |
| * Actors | : | User |
| * Type | : | Primary and Essential |
| * Pre-condition | : | User has already logged into their account and has drawn a shape on to the map. |
| * Main Case Scenario | : | * The user enters the marked location details (place name, landmark, landuse class, and remarks), capture minimum one photo or maximum two photos of the location. * The user can save the details to the local database or user can send the details directly to the server (Data which are sending to the server will be stored first at local DB). |
| * Exception Scenario | : | * User has not filled all the fields, an alert has been popped up ‘all fields are required’. * User denied the camera permission or local storage permission an error has been popped up. * On sending the data to the server if there is no internet connection then an alert will be popped up ‘No internet connection’. * On sending data to the server, if there is an issue in the server or in data then an alert has been popped up which shows ‘Some Error Occurred’. * On saving the data to the local DB, if there’s an error then an alert will be popped up which shows ‘Some Error Occurred’. |

|  |  |  |
| --- | --- | --- |
| **Use Case 5: HISTORY** | | |
| * Actors | : | System and User |
| * Type | : | Secondary and Essential |
| * Pre-condition | : | User has already logged into their account. |
| * Main Case Scenario | : | * System displays the locally stored details of the locations which a user has saved in the device and sent to the server. (Note: Only locally stored data will be displayed.) * User can select multiple items which have not been sent to the server and can send these items to the server or delete these items. * User can export all the items displayed on the History Screen to local storage as an Excel file. The file will be saved in “Downloads/Mapin/exported” folder and the images will be saved in “Downloads/Mapin/images” folder. |
| * Exception Scenario | : | * If there is no data saved locally or sent to the server by the logged in user then ‘No Data Found’ text has been displayed. * The system is unable to connect to the database. ‘Some Error Occurred’ message will be displayed. |

|  |  |  |
| --- | --- | --- |
| **Use Case 6: SHOW LOCATIONS** | | |
| * Actors | : | System |
| * Type | : | Optional |
| * Pre-condition | : | User has already logged into their account. |
| * Main Case Scenario | : | * On the History Screen, when a user presses the ‘Show on Map’ button of a particular location item, then this screen has been displayed. * System will display a Map zoomed into the marked location and displays the marking type (Point, Polygon, or Line) automatically drawn on the Map. |
| * Exception Scenario | : | * Due to low network connectivity Map has not been loaded and a white screen has been displayed. |

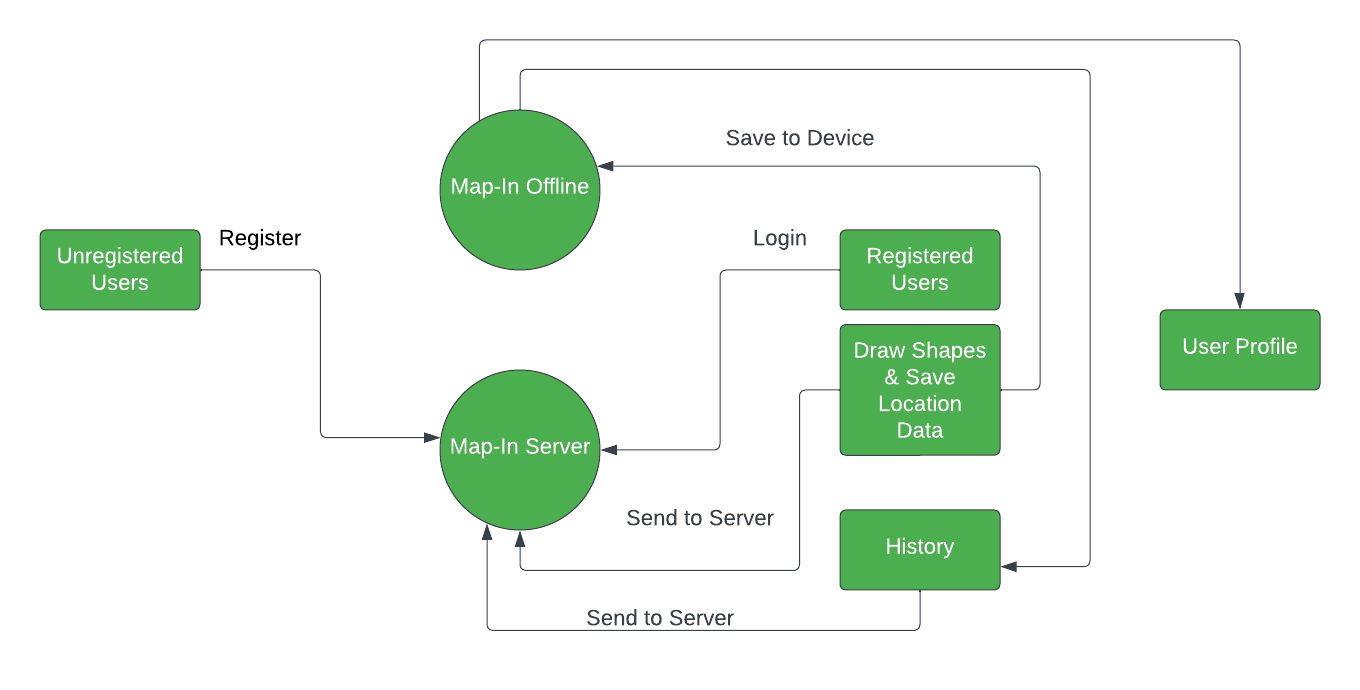
|  |  |  |
| --- | --- | --- |
| **Use Case 7: PROFILE** | | |
| * Actors | : | System & User |
| * Type | : | Optional |
| * Pre-condition | : | User has already logged into their account. |
| * Main Case Scenario | : | * User profile details like Name, Email, Address, Mobile No. has been fetched from local storage and displayed. * When a user presses the ‘Logout’ button, only the locally stored user’s profile data has been cleared (not the user’s marked location data) and the user will be returned to the Login screen. |
| * Exception Scenario | : | * User manually disabled the storage permission then an error has been occurred. |

|  |  |  |
| --- | --- | --- |
| **Use Case 8: ABOUT** | | |
| * Actors | : | System |
| * Type | : | Optional |
| * Pre-condition | : | User has already logged into their account. |
| * Main Case Scenario | : | * This screen displays all the information about the app as well as useful links and support information like developer email and mobile. |
| * Exception Scenario | : | * None. |

## 

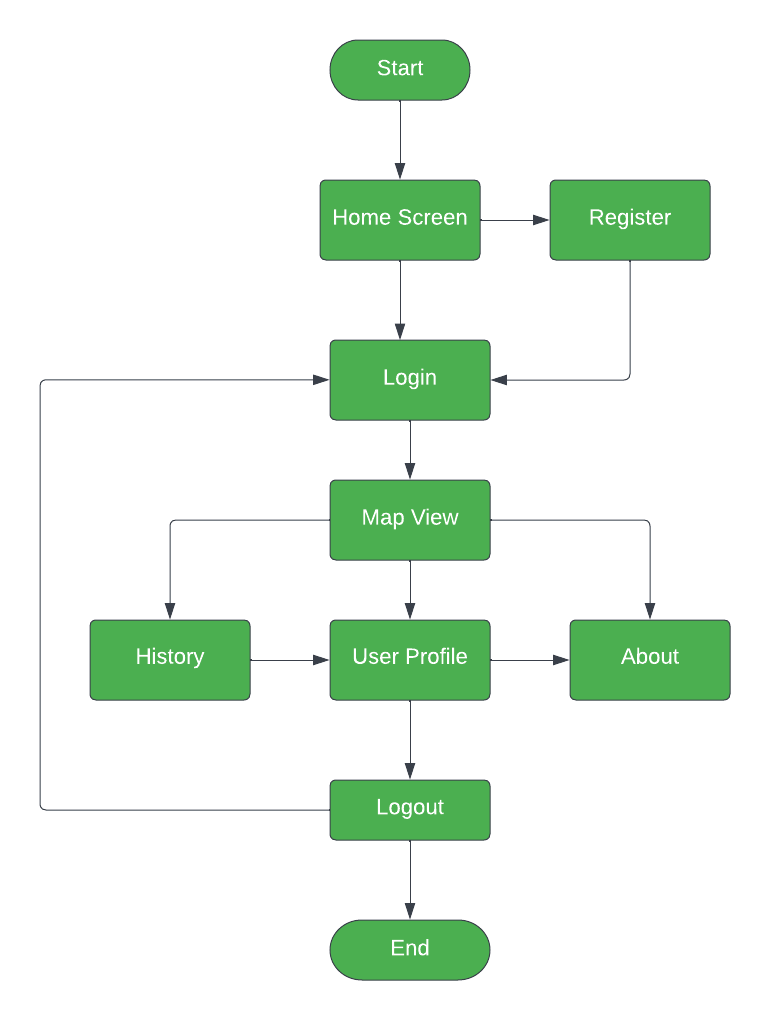
# Data Description

## Data Flow Diagram



* + 1. **Level - 0 Data Flow Diagram**

## Flow Charts



## Database Description

### Table ‘users’

The table contains the information of the users registered in the app.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| id | serial | Gives a serial number and is the primary key of the table. |
| name | VARCHAR(50) | Stores the name of the user |
| mobile | VARCHAR(10) | Stores the mobile number of the user. |
| email | VARCHAR(100) | Stores the email of the user. |
| address | VARCHAR(200) | Stores the street address of the user. |
| city | VARCHAR(50) | Stores the city name of the user. |
| state | VARCHAR(20) | Stores the state name of the user. |
| pincode | VARCHAR(6) | Stores the pincode of the user’s city. |
| registered\_on | timestamp | Stores the registration time of the user. |
| model\_name | VARCHAR(50) | Stores the Brand Name and Model Name of the user’s mobile |
| os\_version | VARCHAR(50) | Stores the OS name and OS version. |

### Table ‘mapin’

The table contains the information of marked locations by the user.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| id | serial | Gives a serial number and is the primary key of the table. |
| userName | VARCHAR(20) | Stores the user name of the user. |
| placeName | VARCHAR(50) | Stores name of the place marked. |
| remarks | VARCHAR(200) | Stores the remarks of the marked location by the user. |
| landmark | VARCHAR(100) | Stores the landmark of the marked location by the user. |
| landuseclass | VARCHAR(20) | Stores the landuse class of the marked location by the user. |
| dateTime | VARCHAR(50) | Stores the date and time of the marked location by the user. |
| locations | VARCHAR(1000) | Stores the latitudes and longitudes of the marked location by the user. |
| shapeType | VARCHAR(10) | Stores the shape type of the marked location by the user. |
| image1 | Text | Stores the path of the first image of the location captured by the user. |
| image1 | Text | Stores the path of the second image of the location captured by the user. |

# User Manual

## Welcome Screen

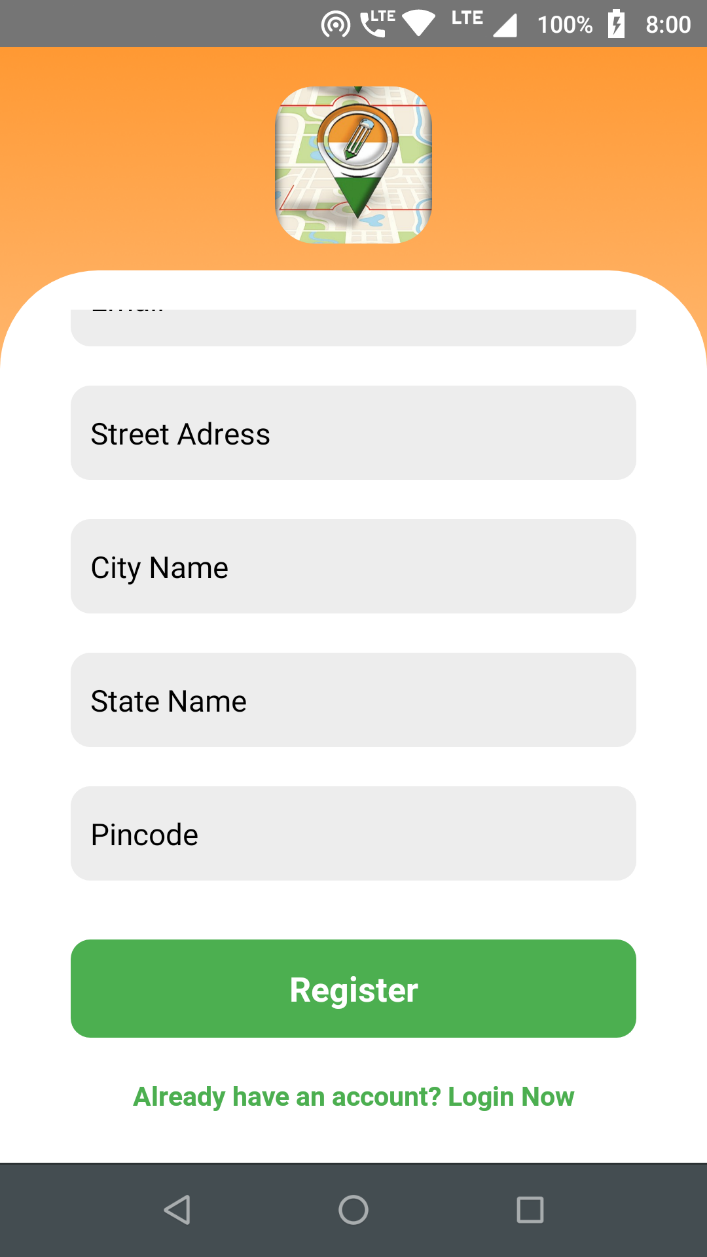
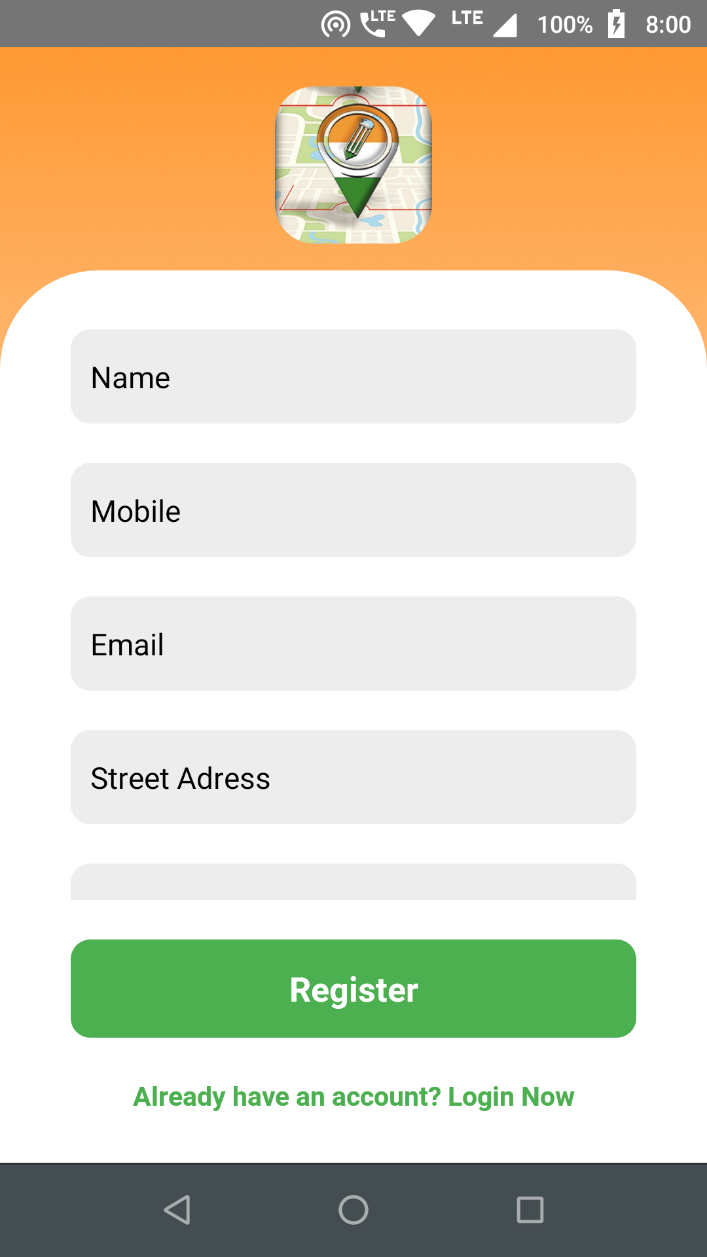
On Launch welcome screen will be displayed to the user as shown below

The screen contains two buttons:

1. **Register (For new users):** New users may create their account
2. **Log In (For registered users):** Existing users can log into their account

## Register

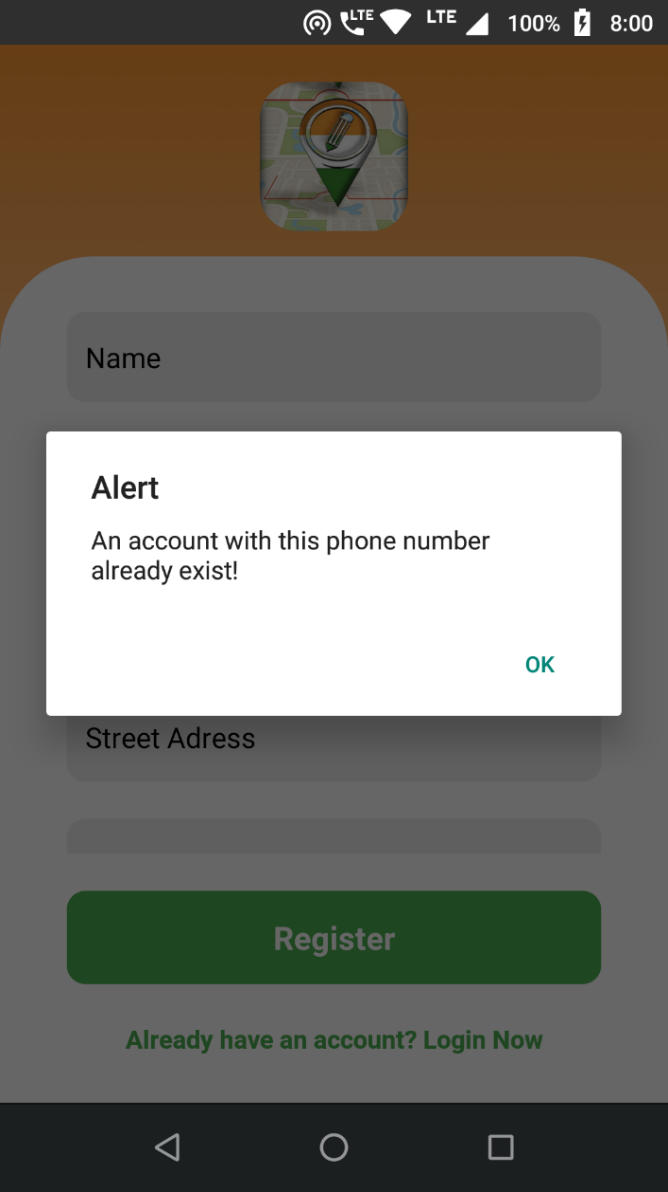
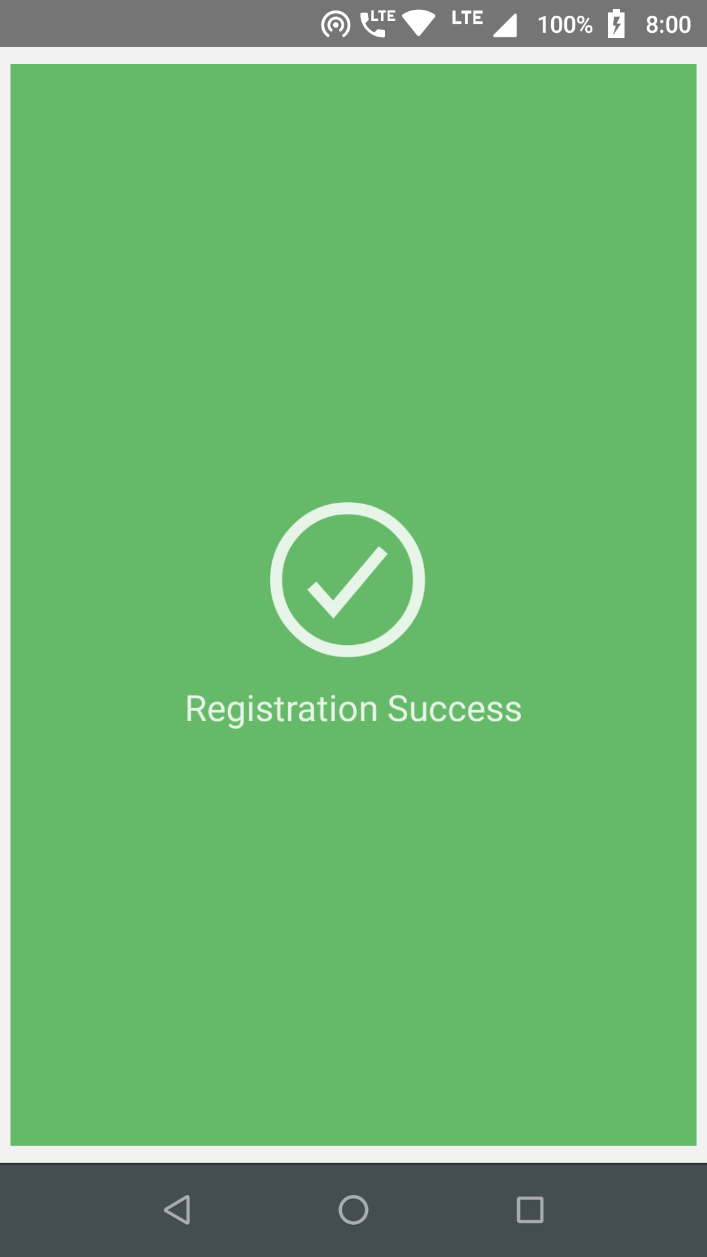
This contains the user registration form. New users may fill in their details which includes Name, Mobile, Email, Street Address, City Name, State Name, and Pincode to create their account.

If a user enters incorrect data (like mobile number having less than 10 digits) or any of the fields is left blank, the same will be notified and registration will be aborted.

If there is no internet connection then an alert will be popped up showing “No internet connection.”

A user can only register using a unique mobile number.

On clicking “Already have an account? Login Now” user will be navigated to the “Login Screen”.

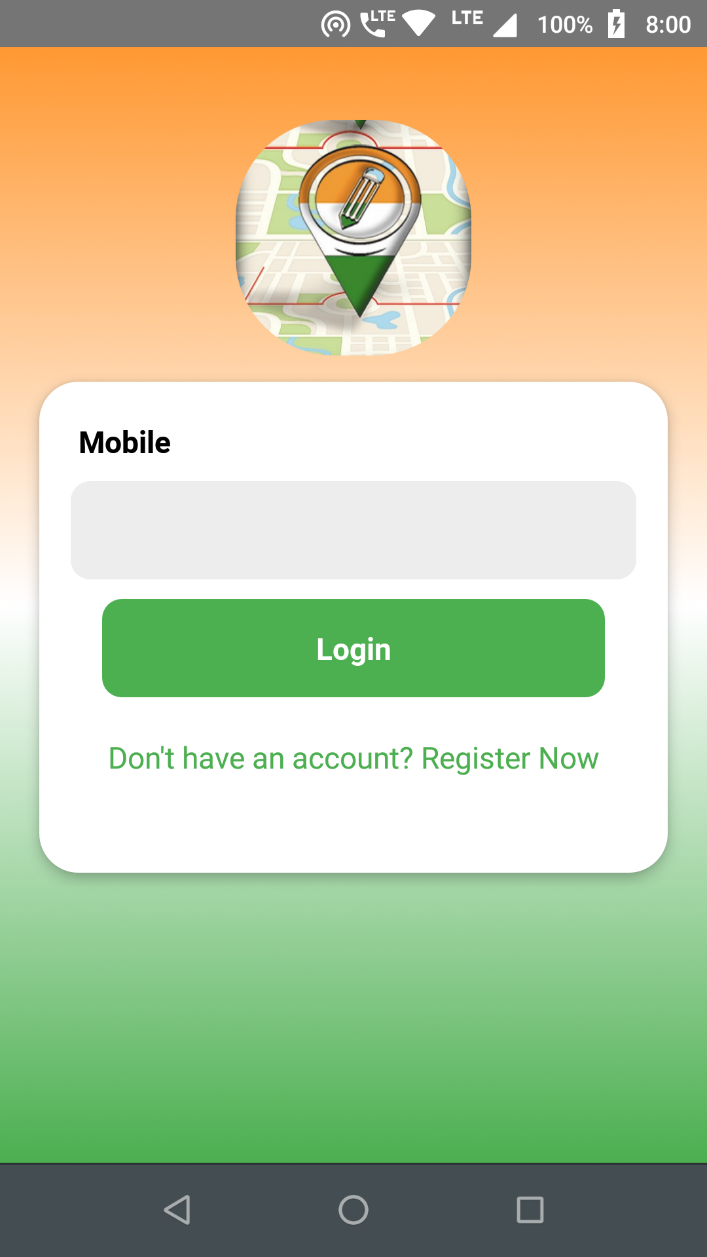


If a user is already registered then an alert will pop up alerting that “An account with this phone number already exists!”

Successful registration will display the message “Registration Successful” and user will be navigated to “Login” screen.

## Login

The Login Screen will be displayed to the user as shown below.

The user should enter their registered mobile number (containing 10 digits) to successfully log into their account.

If the mobile number is not registered, a warning message will be prompted and the user may register from the link given below the log in button.

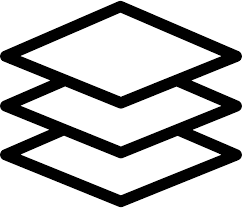
If there is no internet connection then an alert will be popped up showing “No internet connection.”

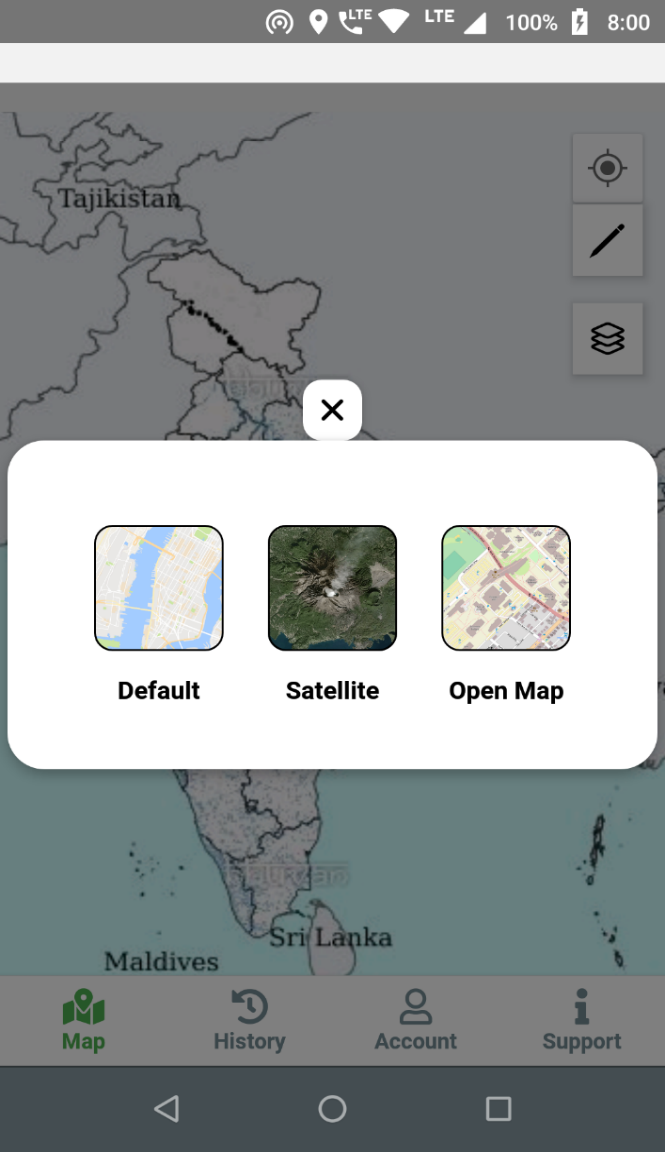
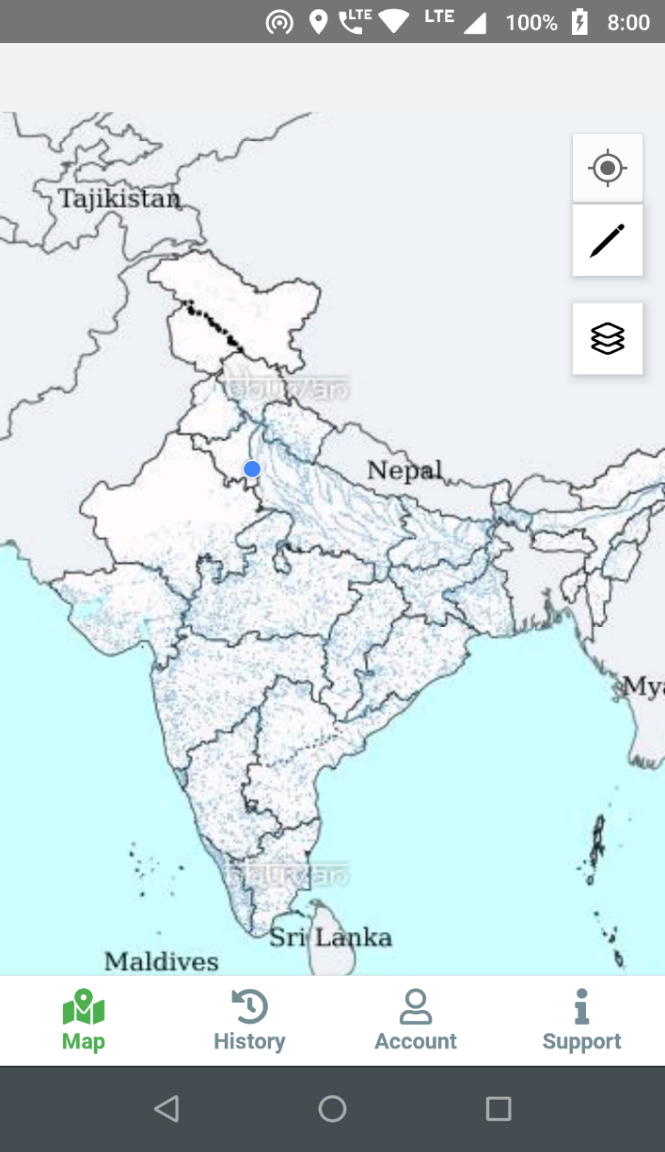
After successful authentication the user will be navigated to the Map Screen.

## Map View

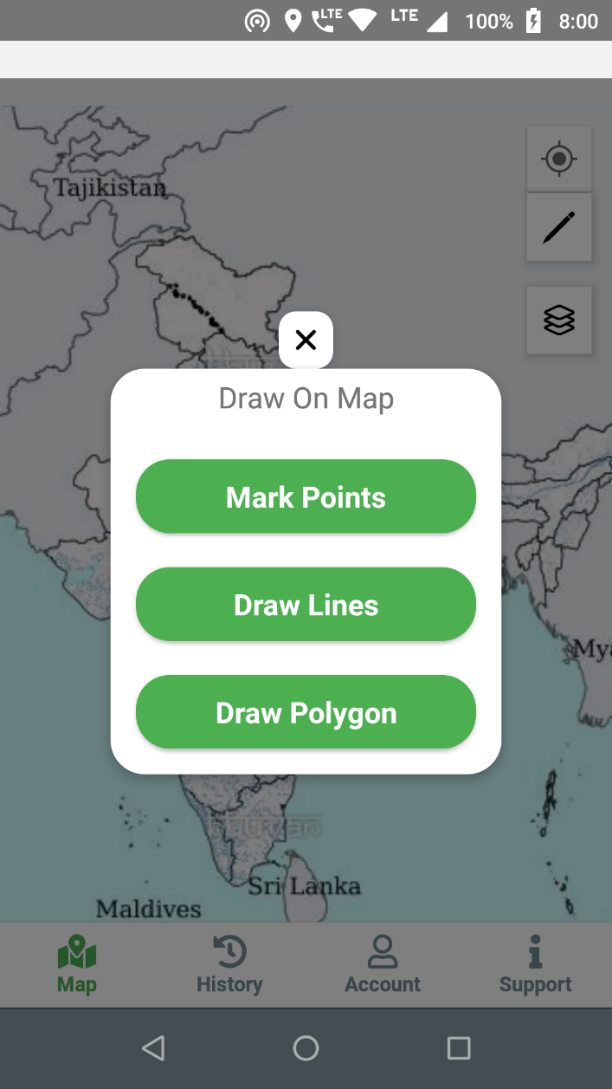
After successful login, user will be navigated to the Map screen. Map screen fetches and displays the Map.

After the Map has been loaded then a prompt to ask the location permissions will be displayed, if a user allows the location permission then the user location will be displayed on the Map. Without the location permissions the app will not work.

Map Screen displays a Map in satellite view by default. A user can change the layer by pressing the layer icon to change the source to OSM, Bhuvan Map, or Satellite image.



Draw - Free interface iconsUser will draw the shapes on the map by pressing the pencil icon , After pressing the pencil icon a popup will be displayed to select the type of the geometry

for drawing on the map , available options are Point, Polygon, or Line as shown in the figure below.

After selecting the shape type, a drawing window will be displayed to draw the shape on the map.

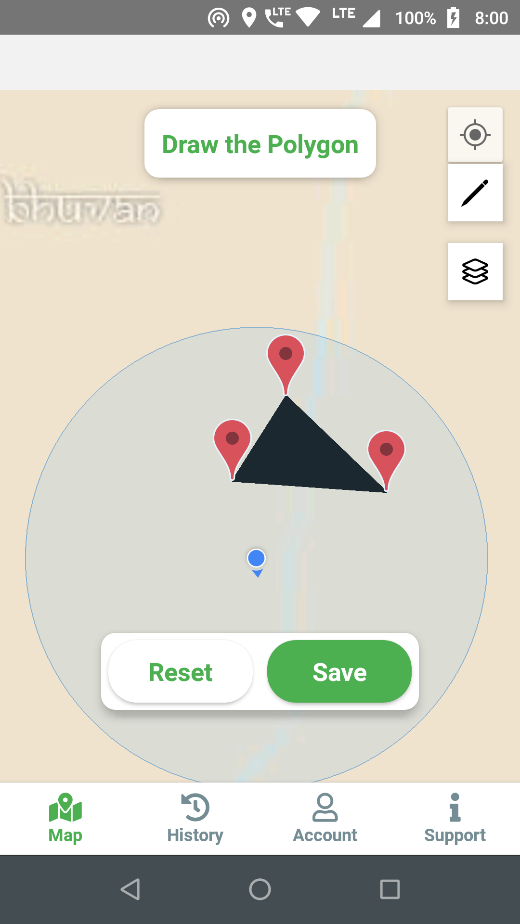
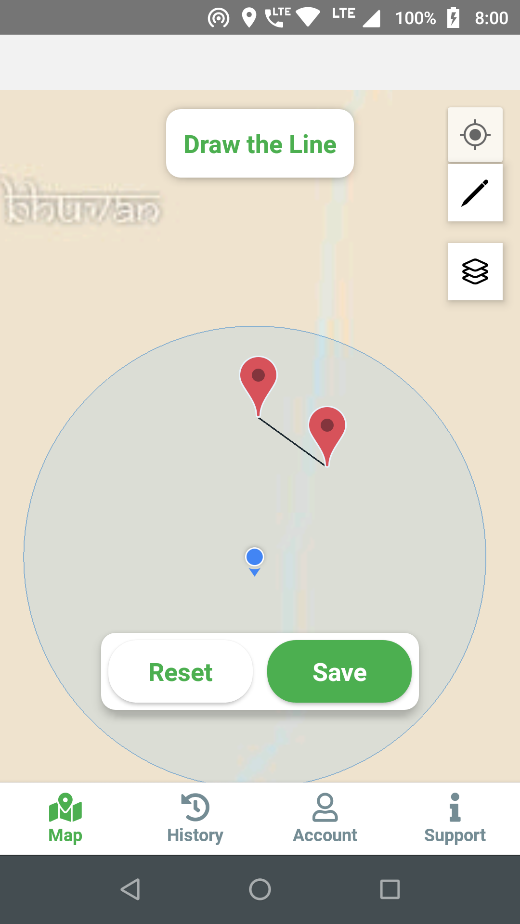
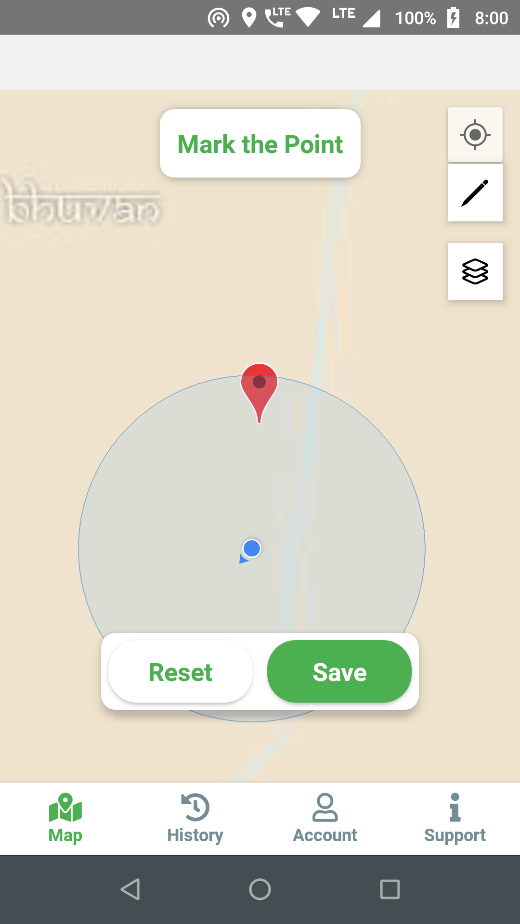
To mark a “Point” user must mark at-least one location,

To draw the “Polygon” user must mark at-least three locations,

and to draw “Line” user must mark at-least two locations.

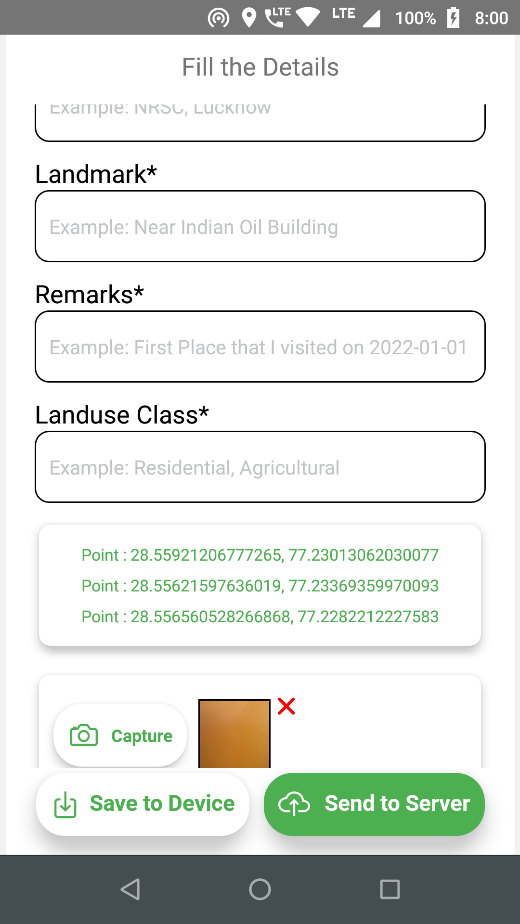
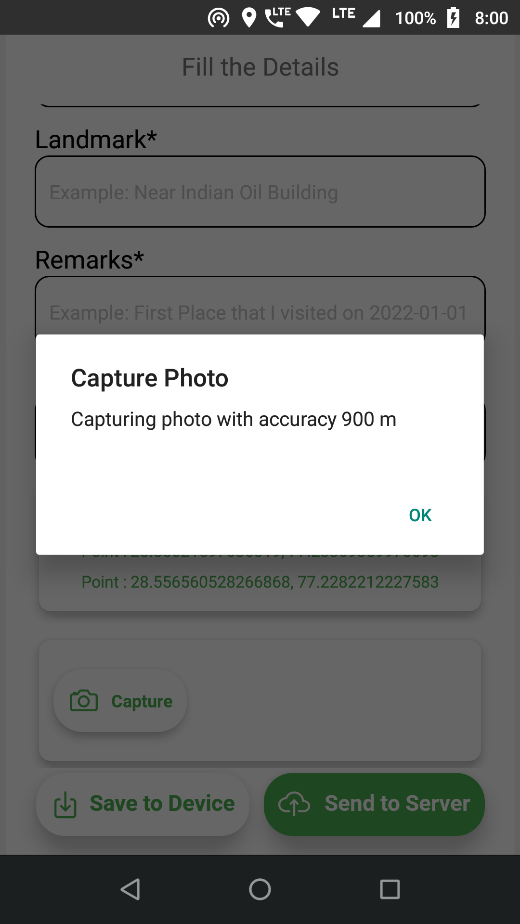
Two buttons “Save” & “Reset” in the drawing area will be displayed. By pressing the “Reset” button the marker will reset and clear the drawings

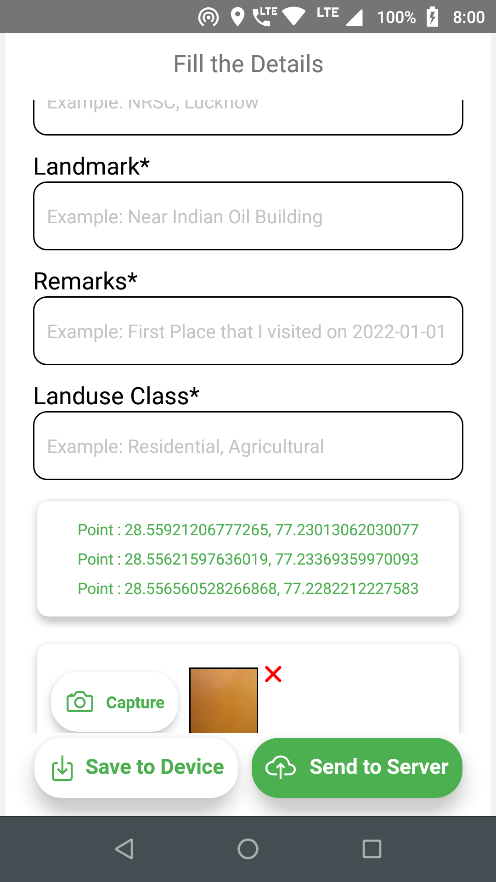
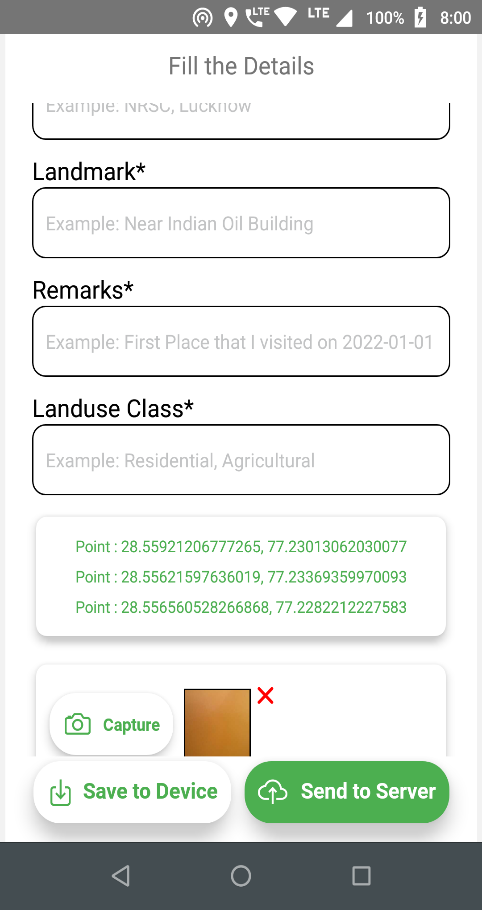
On pressing save the user will be navigated to the Attribute Screen.



## Attribute Form

On this screen, user will fill all the attributes related to the captured coordinates. The Attributes are Place name, Landmark, Remarks, Land Use class (all fields are mandatory) , User has to capture at-least one picture and maximum two pictures of the location with accuracy of 30 meters of the drawn shape. A user can delete and retake pictures if needed. Latitude and Longitude of the marked points by the user are displayed below the text input area.



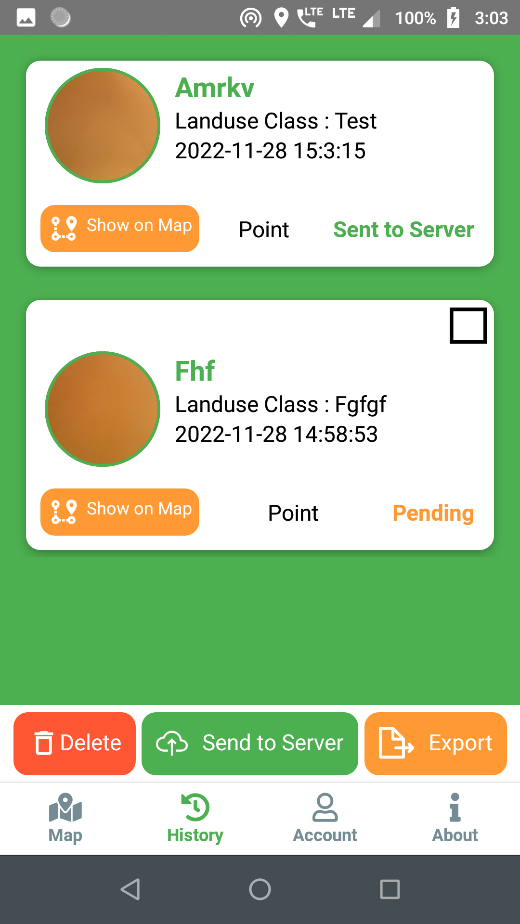


There are two buttons & , on pressing “Save to Device the data will be saved in the local database, while on pressing “Send to Server” the data will be sent to the server as well as saved in the local database.

After successfully saving or sending of the data the user will be navigated to the Map Screen.

## History

History Screen displays the data available in the local database as a list , where each item contains an image captured by the user, Place name, Landuse Class, time stamp, “Show on Map” button, shape type & the status of items, where “Pending” is displayed when the item has not been sent to the server and “Sent to Server” when item has been sent to the server.

If there’s no item available in the local DB then “No Data Found” will be displayed.

When a user touches the image icon then a popup containing captured image will be displayed on the screen.

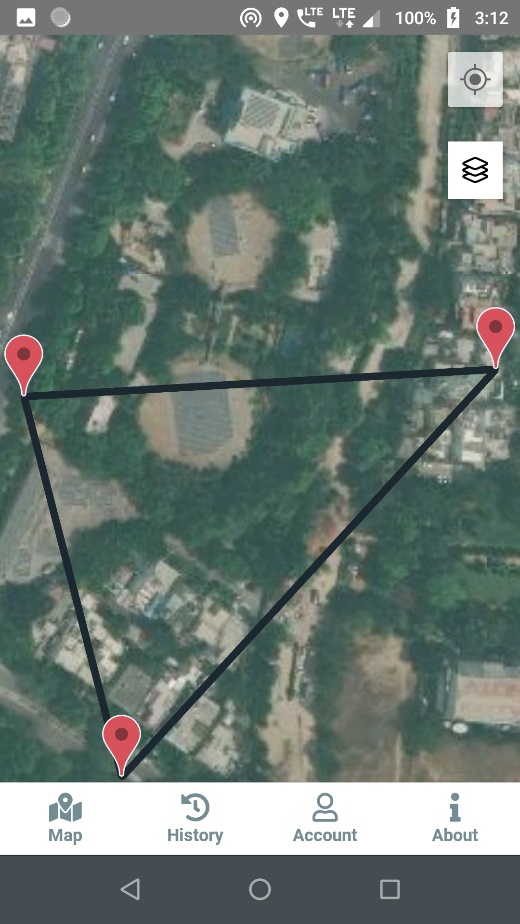
At the bottom there are 3 buttons: “Delete”, “Send to Server”, & “Export”.

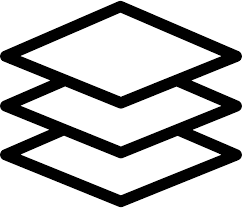
* **Delete**: User can select multiple items to delete from the local DB. Items which are only saved in local DB can be deleted. Items which are “Sent to Server” cannot be selected for deletion.
* **Send to Server**: User can send multiple items to the server by selecting the items.
* **Export**: User can export all the items displayed on the History Screen to local mobile storage as an Excel file. The file will be saved in “Downloads/Mapin/exported” folder and the images will be saved in “Downloads/Mapin/images” folder.

When a user clicks on the “Show on Map” button then a new Screen will be opened to display its location on the map.

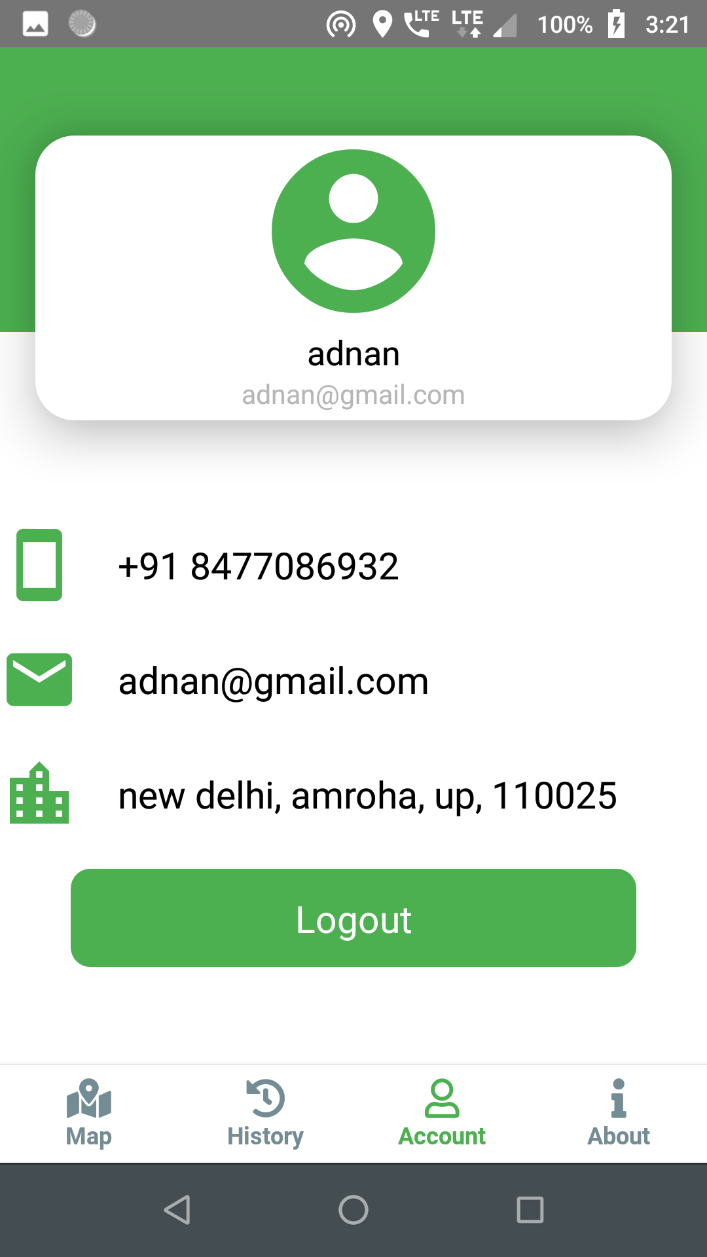
## Show on Map

When a user clicks on the ‘show on map’ button on history screen then this screen will be opened. In this screen a map containing the drawn shape on the map zoomed to the shape will be displayed.



When a user clicks on the marker then Place Name & Landmark will be displayed on the marker. A user can change the layer by pressing the layer icon to other available sources like OSM, Bhuvan, or satellite image for visualization.

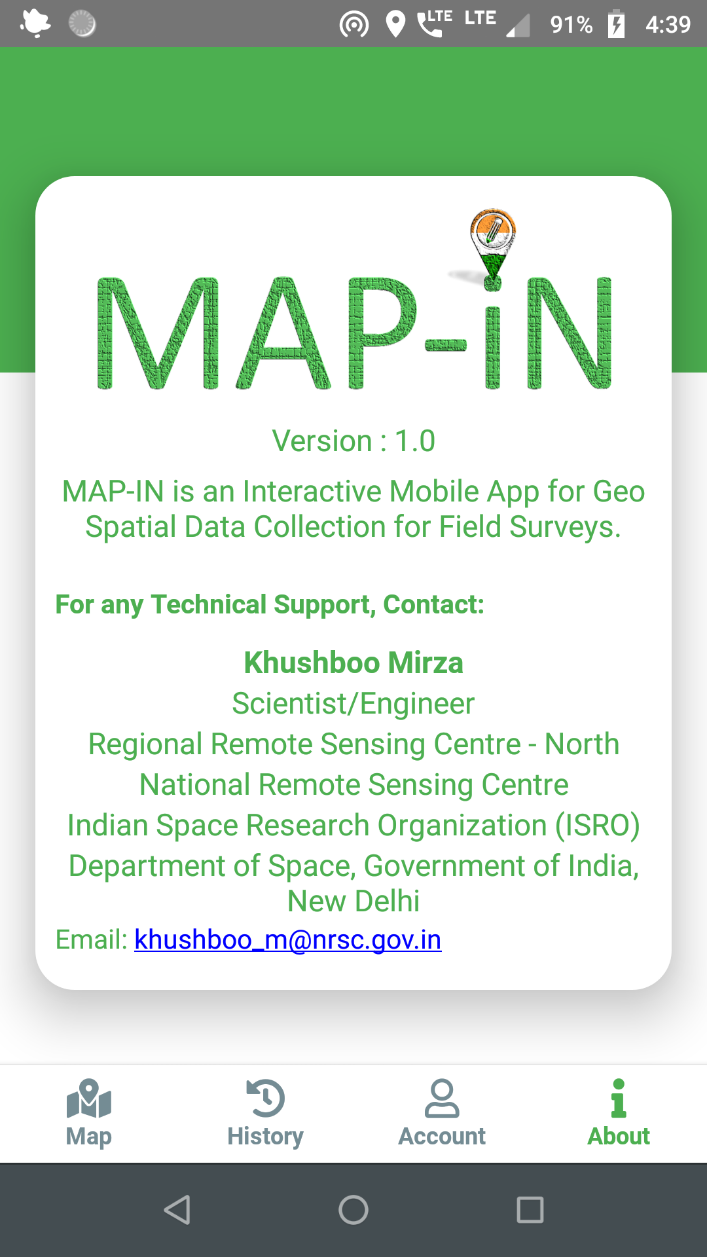
## User Profile

The User Profile Screen contains all the information of the user, entered at the time of registration. There is a “Logout” button, when pressed the user will be logged out of the app and navigated to the Login Screen of the app.

*The profile can only be viewed and it cannot be changed.*

## About App/Support Screen

This screen contains the info about the app, version of the app, and contact details of the support team.



\*\*\*\*\*\*\*\*\*\*\*\*\*

# Definitions and Acronyms

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| SDD | Software Design Document |
| DFD | Data Flow Diagram |
| NPM | Node Package Manager |
| JDK | Java Development Kit |
| CLI | Command Line Interface |
| SQL | Structured Query Language |
| PHP | PHP Hypertext Preprocessor |
| HTTP | Hypertext Transfer Protocol |
| XAMPP | XAMPP Apache + MariaDB + PHP + Perl |
| IDE | Integrated Development Environment |
| CGI | Common Gateway Interface |
| RDBMS | Relational Database Management System |
| GUI | Graphical User Interface |
| NA | Not Applicable |
| ACC | Accuracy |
| OSM | Open Street Map |
| pgSQL | PosgreSQL |

# References

[1] [React Native official website](https://reactnative.dev/)

[2] [Node.js official webpage](https://nodejs.org/en/)

[3] [JDK official webpage](https://openjdk.java.net/projects/jdk8/)

[4] [Android Studio official webpage](https://developer.android.com/studio/)

[5] [PostgreSQL official webpage](https://www.postgresql.org/)

[6] [PHP official webpage](https://www.php.net/)

[7] [XAMPP official webpage](https://www.apachefriends.org/download.html)

[8] [phppgAdmin github repo](https://github.com/phppgadmin/phppgadmin)

[9] [Stack Overflow](https://stackoverflow.com/)

[10] [Bhuvan Map](https://bhuvan-app1.nrsc.gov.in/bhuvan2d/bhuvan/bhuvan2d.php)

[11] [Open Street Map](https://en.wikipedia.org/wiki/OpenStreetMap)

[12] [Map Box](https://www.mapbox.com/)

[13] [Git Hub](https://github.com/adnan-jafri7)