Project Title: CROPDEX v2.0

Project Description : A Crop Identification & Crop Disease Diagnostic Mobile application

Team Name : Blitz

Team Members : Anjali Rahangdale
Siddhesh Sovitkar
Pradyumna Mahajan
Nitin Giri
Ashutosh Pandey
Ritik Kulkarni

About the App

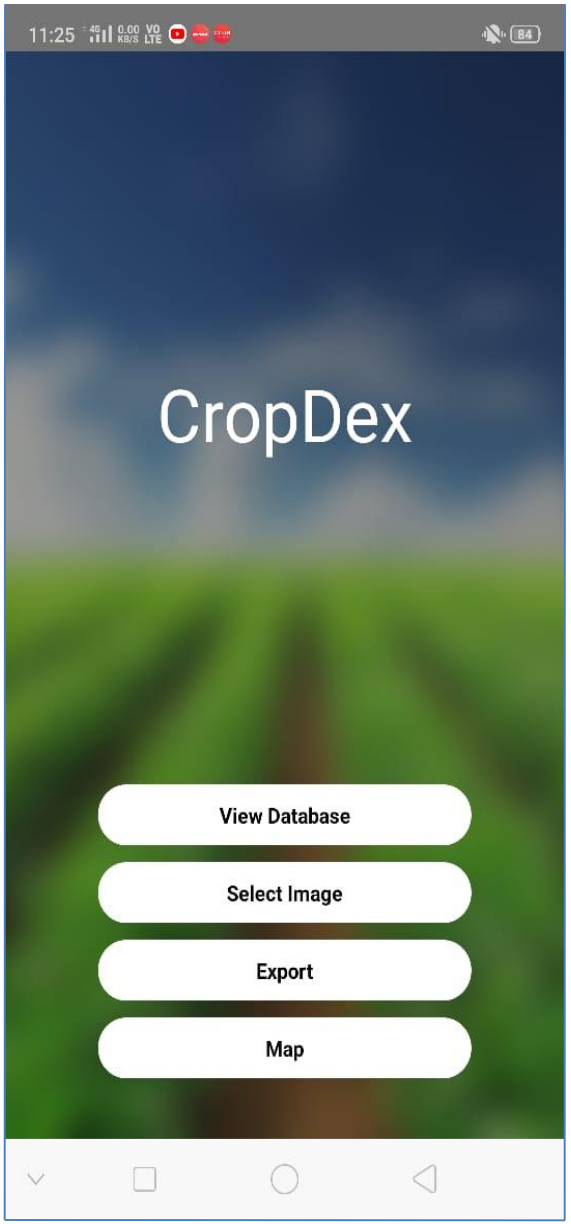
- **CROPDEX v1.0** is a **Cross Platform** mobile application developed by our team, using **React Native**, which acts as a encyclopedia of crops, helping the user to identify a crop from a picture of it which is captured/ uploaded by the user, using Deep Learning.

- The **v1.0** just focuses on identifying and giving information about the crop based on the image uploaded by the user.

- **CROPDEX v2.0** , which will be built on top of v1.0, would also focus on **detecting plant diseases on the most important crops and recommends customized treatment measures.**

- The App will serve as **Crop Diagnostic System** that would assist the farmers to identify the crop diseases in case if any & provide guidance on best and latest available treatment for it.

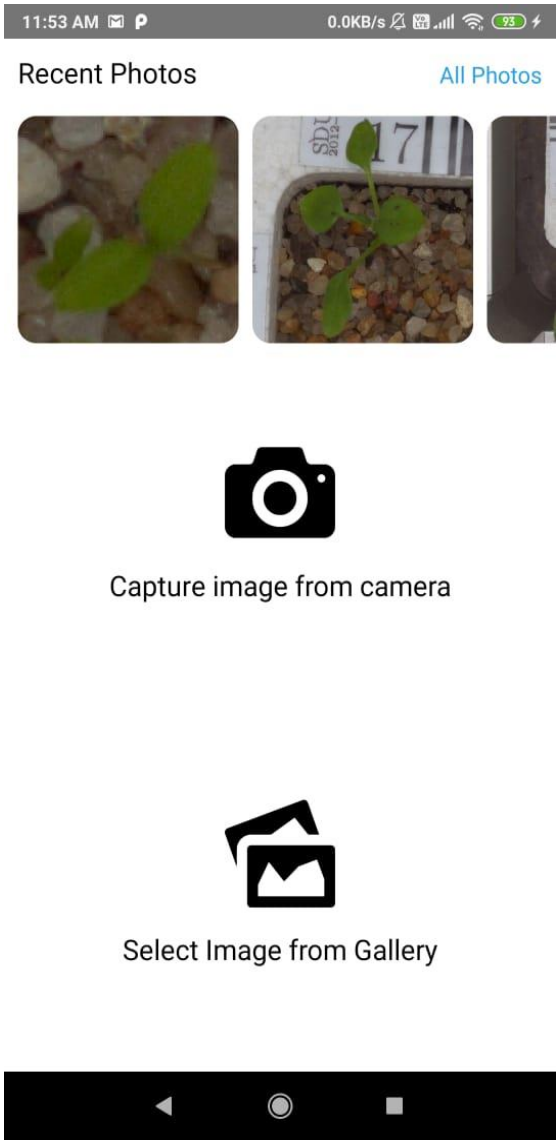
- It can also be used as a **smart informative guide for those interested in exploring farms**.



WORKFLOW OF THE APP

The following basic steps are involved in identifying the Crop from the Image :

1. The User captures / selects crop image using the app's "Select Image" interface.
2. The App uploads the image to the Server.
3. The Server side app classifies the crop image using a trained Resnet50 CNN classifier and sends the response to the app.
4. The app stores the crop image along with its geolocation & received response from the server into an Internal Database.



Salient features :

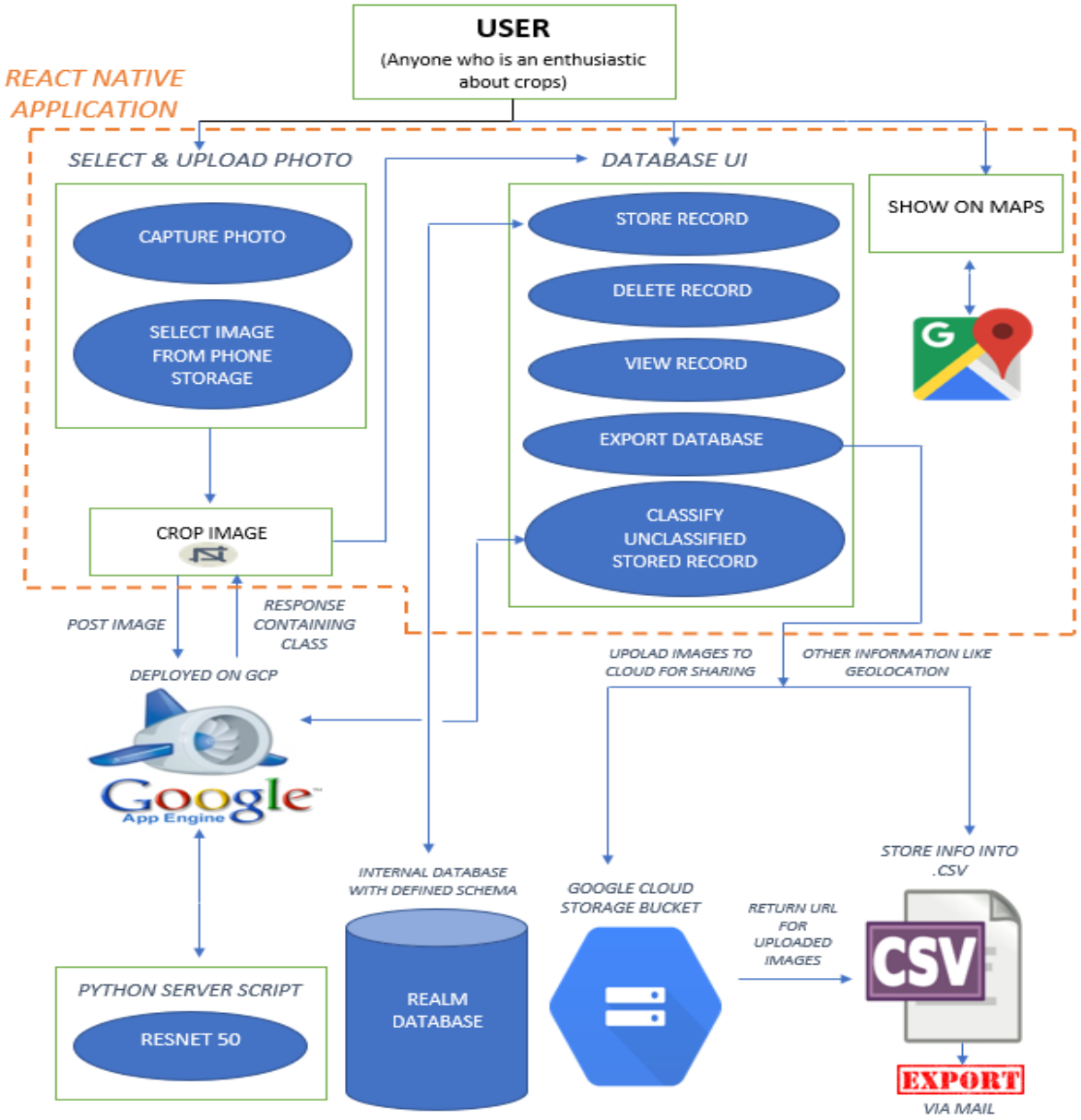
▪In case of **Low / No Internet connectivity**, the app stores the captured image along with geolocation into the Internal DB ,which can be then classified later when an active Internet connection is available.

▪The **"Select Image " UI** allows user to capture photos using camera as well as select photos from phone storage (Gallery).

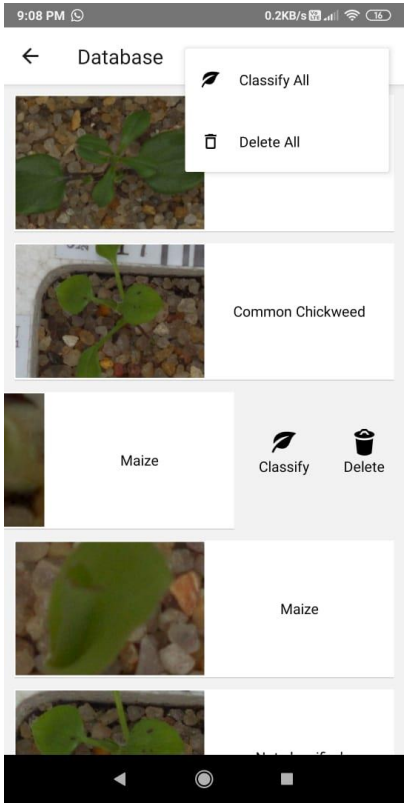
▪The **Internal DB** is a local on-device database created using **Realm.js** which stores the information of identified crops, and can be accessed using the app's "View Database" interface.

▪The Internal DB can be **exported as CSV file via Mail** & also **uploaded to Cloud Storage** for **backup purpose**, which is managed through **user login credentials for the app**.

Use Case and Flow Diagram



More UI / UX features of the App



“View Database” UI :

•It is an interface to access the internal DB.

•It provides options like:

•**Delete** : to delete a record from the DB.

•**Delete All** : to delete all records from the DB.

•**Auto Classify** : If this option is selected, all unclassified records will be classified automatically whenever active internet connection is detected by the app. (No human intervention needed)

•**Classify** : To manually classify an individual unclassified record(crop Image) on active internet connection.

•**Classify All** : it will classify all unclassified records. (User needs to check status of internet connection)

“View on Maps” UI :

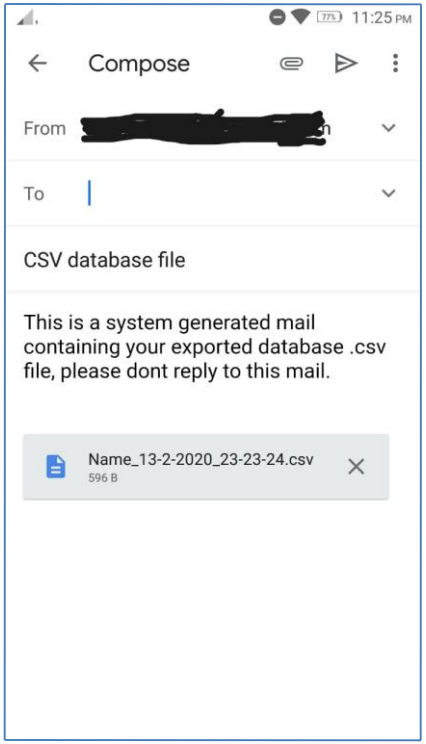
▪It Pins the Geolocation of all the Crop Images captured by the user onto a Map.

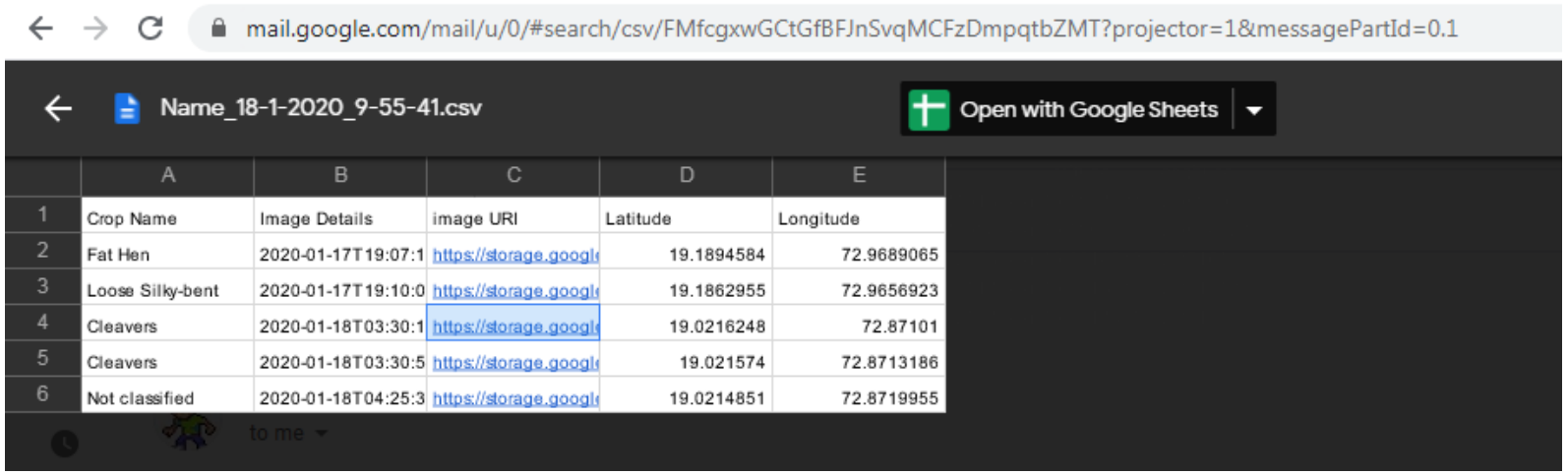
▪The Map Markers on being clicked, display the respective crop name and details of the date of capture.

▪It helps user to locate the captured crop images easily on Maps ,as latitude and longitude values are not understood by everyone.

Exporting Internal DB as CSV file

- The DB is exported as a CSV file.
- A copy of the all the locally stored Crop images is uploaded on **Google Cloud Storage** for **backup** as well as **ease of access** for others to whom the information is shared.
- Before exporting the DB file, the **URL** of the copy of Crop Images stored on Google Cloud Storage is resolved and entered in the CSV files, so that images can be viewed from any machine.





	A	B	C	D	E
1	Crop Name	Image Details	image URI	Latitude	Longitude
2	Fat Hen	2020-01-17T19:07:1	https://storage.google	19.1894584	72.9689065
3	Loose Silky-bent	2020-01-17T19:10:0	https://storage.google	19.1862955	72.9656923
4	Cleavers	2020-01-18T03:30:1	https://storage.google	19.0216248	72.87101
5	Cleavers	2020-01-18T03:30:5	https://storage.google	19.021574	72.8713186
6	Not classified	2020-01-18T04:25:3	https://storage.google	19.0214851	72.8719955

UPDATED FEATURES IN CROPDEX v2.0

- Currently , the Resnet50 Convolution Neural Network (CNN) Model is trained on Crop seedling dataset available on Kaggle.com (https://www.kaggle.com/vbookshelf/v2-plant-seedlings-dataset) as well as a **dataset of 10 crops** created by our team through **Web scraping**, making it capable to identify and classify crops at most of the growth stages.

- To make the model also identify Crop diseases, we would train it on an updated dataset consisting of Diseased Crop classes in addition to the existing classes of healthy crops.

- The Server App can also use the **images uploaded by users on cloud storage to train the model**, making the model training process more dynamic which in turn would make the model robust

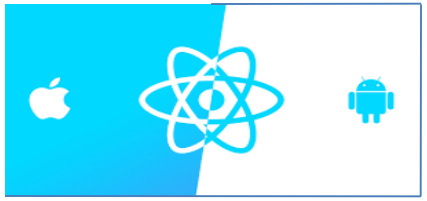
- The Server side application will contain a **database** stored on **Google App Engine's Datastore**, consisting of **up-to-date information on crop diseases and its remedies,** which would be recommended to the user as a response along with the identified crop disease.

- The App has **login** for each user to manage operations such as storing images and other information on Cloud Storage as well as to export data via Mail, providing **security** and **backup & recovery.**

- The existing internal DB of the Mobile app can be modified to store extra information such as :
    - Potential Diseases for the crop.
    - Disease Type (If detected, otherwise NULL)
    - Other Common Symptoms
    - Remedy/Treatment

TECHNOLOGY STACK

React Native is used for developing **the Cross Platform** mobile application, that runs both on Android & iOS.



The **Fastai Library** is used in training the CNN model on the dataset. It is **a Deep Learning Library** built on top of PyTorch Library, and makes task of training models easy.

Google App Engine is used for hosting our **Server Application** wh identifies the crops & crop diseases. It is a Platform for developing a hosting web applications in **Google-managed Data centers**.

Google Cloud Platform (GCP) is used for providing services such as **Cloud Storage**. Even Google App Engine is one of the services provided by GCP.

Starlette is light-weight Python based framework used for Server side scripting.

Google Maps Platform is used to integrate Map based features into the application

Dependencies / Show stopper

Dependencies :

- Proper servers are needed to store all the crop pictures and their respective informative entries.
- Good internet connection is a must for sending requests.

Show stopper :

- Google Servers (GCP) may sometimes shutdown for updation.
- Low network connectivity.