

CS551 Second Increment Report

By
Rishabh Bhojak (SG2)
Bhargava Gellaboina (SG2)

Objectives

In the first increment, we were able to successfully create a working application that contained all of the desired features including functionality to observer, analyst and coordinator. However, the beta release of the program to some specific users revealed several bugs and performance issues with the current version, including crashes, lag, etc. The objectives of this second increment is to learn from user feedback and internal testing to specifically address these issues, in order to create a streamlined, bug-free version of the application that can be released to the public for widespread use.

For second increment, our main focus was major bug fixes and improving overall performance of the application, which makes this project an application.

Development was done on;

Android Development API 19
Google Play Services 5077000_r18

Import Existing Services/API

All Existing Services

Google Maps Android API v2: Coordinator opens the coordinator view in android

application which calls the MapView.getMap() function to load the Google maps for the Coordinator. At the same time, android application also pulls all the images using getAllImageData() function using Asynchronous Http Service it displays the images on google maps and shows it to the Coordinator.

To use the application, we followed the following steps:

1. We created SHA1 for our own signature key
2. We registered with Google APIs
3. We then generated key for our own application
4. Then created the project and use the key for development.

Google Maps Geolocation API: When the observer takes pictures using the application, we are saving Geolocation of the user(where the picture was taken)using geoLocService.getLatitude() and geoLocService.getLongitude() functions in to MongoDB and using the Latitude and Longitude values we are querying for the location using Geolocation API.

```

public void addGeoLocation() {
    geoLocService = new GeoLocationService(ObserverActivi- ty.this);
    // check if GPS enabled
    if (geoLocService.canGetLocation()) {
        double latitude = geoLocService.getLatitude();
        double longitude = geoLocService.getLongitude(); latTv.setText("Lat: " + latitude);
        lonTv.setText("Lon: " + longitude);
        double latlon[] = { latitude, longitude }; scene.setLocation(latlon);
    } else {
        geoLocService.showSettingsAlert();
    }
}

```

Example code to load the location of images with their description in google maps:

```

// Code to get the image data from the MongoDB

protected String doInBackground(String... params) {
    ByteArrayOutputStream out = null;
    HttpClient client = new DefaultHttpClient();
    String url = Utils.HOST + Utils.RES_LIST_ITEMS + "?start=" + sceneList.size() +
    "&limit=20";
    System.out.println(" Url: " + url);
    HttpGet get = new HttpGet(url);
    HttpResponse response;
    try {
        response = client.execute(get);
        out = new ByteArrayOutputStream();
        response.getEntity().writeTo(out);
        out.close();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        get.abort();
    }
    return out.toString();
}

```

```
// Post processing after retrieving the data

@Override
protected void onPostExecute(String result) {
dialogShow = false;
System.out.println("Data from server: " + result);
if(populateSceneData(result)){
//Choose the required data to display on map using below function call
populateMarkersToMap();
MapsInitializer.initialize(obj.getActivity());

// Updates the location and zoom of the MapView
System.out.println("Before CAMMAP");
CameraUpdate cameraUpdate = CameraUpdateFactory.newLatLngZoom(new
LatLng(sceneList.get(0).getLocation()[0], sceneList.get(0).getLocation()[1]), 15);
mMap.animateCamera(cameraUpdate);

//return obj.rootView;
}

// Populate the required data from the retrieved data

private boolean populateSceneData(String data) {
if (data == null) {
//listView.removeFooterView(loadMoreView);
return false;
}
JSONArray jsonList;
try {
jsonList = new JSONArray(data);
if (jsonList.length() == 0) {
loadmore = false;
//listView.removeFooterView(loadMoreView);
} else {
if (jsonList.length() < 10) {
//listView.removeFooterView(loadMoreView);
loadmore = false;
}
for (int i = 0; i < jsonList.length(); i++) {
JSONObject sceneJSON = jsonList.getJSONObject(i);

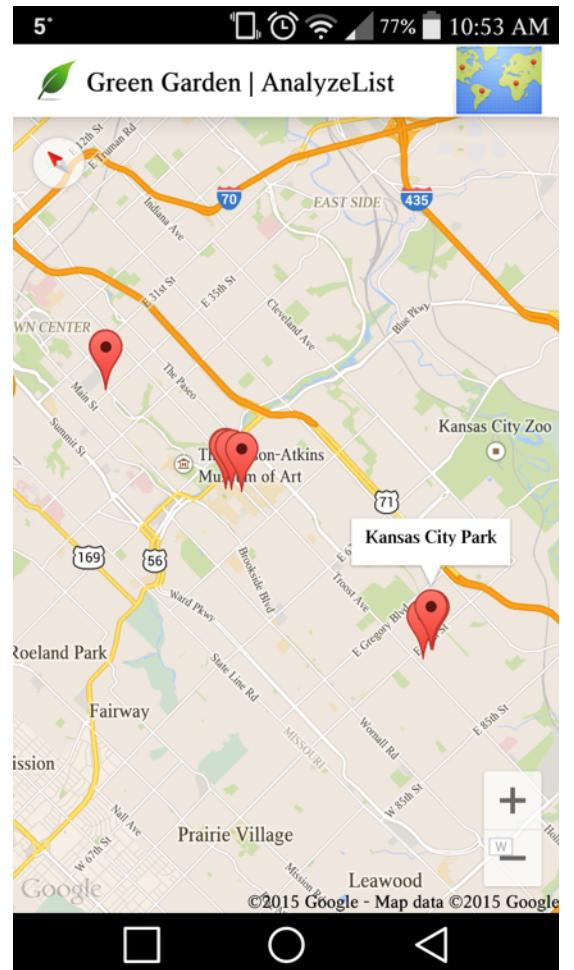
```

```

// Image Mapping
Image image = new Gson().fromJson(sceneJSON.getString("imageData"),
Image.class);
// Location Mapping
String sceneId = sceneJSON.getString("_id");
String locationString = sceneJSON.getString("location");
double location[] = extractLocation(locationString);
// Description Mapping
String description = sceneJSON.getString("description");
Scene scene = new Scene(sceneId, image, description, location);
sceneList.add(scene);
}
}
} catch (JSONException e) {
e.printStackTrace();
}
return true;
}

```

We were able to load the image data from MongoDB in google maps. Here is the recent screenshot of our application.



Camera Object Android: We are using Android Camera Object to provide the observer functionality to take pictures. Pictures are taken using the camera object and stored into mongoDB.

To use camera object we declared the permission in manifest, open(init) to open the instance of the camera, set index value and give this to onClick functions.

Below is the example code for using camera object to take camera pictures in our application:

```
private Camera getCameralnstance() {  
    Camera c = null;  
    if (ObserverActivity.this.getPackageManager().hasSystemFeature(  
        PackageManager.FEATURE_CAMERA_ANY)) {  
        Log.i("Camera", "There is no Camera on this device");  
    } else {  
        Log.i("Camera", "There is no Camera on this ");  
    }  
    try {  
        Camera.Cameralnfo cameralnfo = new Camera.Cameralnfo();  
        int cameraCont = Camera.getNumberOfCameras();  
        for (int camIndex = 0; camIndex < cameraCont; camIndex++) {  
            Camera.getCameralnfo(camIndex, cameralnfo);  
            if (cameralnfo.facing == Camera.Cameralnfo.CAMERA_FACING_BACK) {  
                try {  
                    c = Camera.open(camIndex);  
                } catch (RuntimeException e) {  
                    Log.e("Camera",  
                        "failed to open camera: "  
                        + e.getLocalizedMessage());  
                }  
            }  
        }  
    } catch (Exception e) {  
        Log.i("camera", "Unable to open camera");  
    }  
}
```

```
}

return c;
}

public void onClickCapture(View v) {

capture_status.setText("scene captured");
capture_status.setVisibility(View.VISIBLE);
capture_status.setAlpha(0.6f);
image_capture.setText("Captured");
mCamera.takePicture(null, null, new MyPictureCallback(
ObserverActivity.this));
image_capture.setEnabled(false);

}
```

Detail Design of Services

**Write User Stories using ScrumDo
(Screenshot)**

Stories

#20 As a developer I want to write test cases for the application to do unit & integration testing.
Unit testing by writing jUnit Test Cases for various test cases
Reviewing Tasks | 0 Comments junittesting testing testcases Testing rbx44, bgz82 20

#19 As a developer I want to test the performance of the application using 15 phones at same time.
System testing was done to do the thorough evaluation of the performance by installing the application on 15 phones and running at some time. No major flaws found except analyst list loading getting slower due to the server limitation
Done Tasks | 0 Comments systemtesting performanceevaluation rbx44, bgz82 8

#18 As a developer I want to add health_status of an image to the analyst, and purpose a new machine learning solution to it.
Analyze the health of the green garden by checking the temporal & intensity of color of green garden in images automatically using supervised machine learning technique.
Todo Tasks | 0 Comments machinelearning gardenhealth greenish temporal intensity histogram analyst Functionality rbx44, bgz82 40

#17 As a developer I want to change the image holder view to select the regions in an image to analyze.
Selecting the regions and playing with holder to stretch or shrink to an particular region was difficult, We changed the holder view, which makes it easy to select regions easily for analyst.
Done Tasks | 0 Comments Imageholder selectregions analyst Bugs rbx44, bgz82 5

#16 As a developer, I want to fix Coordinator loading Google maps problem and add functionality to show where the images were taken in the map and load them.
Previously Coordinator was broken, Coordinator can now view the images and their location in google map and load the images from there.
Reviewing Tasks | 0 Comments googlemaps loadimages coordinator Bugs rbx44, bgz82 13

#15 As a developer I want to add functionality to Analyst to analyze multiple regions in an image and provide a layout which makes it easier to represent.
Earlier, Analyst were able to analyze only one part of the image. Analyst can analyze multiple regions in an image and can view the already analyzed regions in an image in the side, no need to remember what you did.
Done Tasks | 0 Comments multipleregions addlayout analyzeregions analyst Functionality rbx44, bgz82 13

#14 As a developer, I want to fix the Observer image capture and also add Status(ack back from Mongo) to make sure that the image was loaded to MongoDB.
Previously, only the images loaded into mongoDB were analyzed, analyzing the image captured by the application crashed the application.
Done Tasks | 0 Comments bugs observerimage codeerror None rbx44, bgz82 8

#13 As a developer we want to do introduce the caching for the first time.
Caching helps from not loading the thumbnails all the time, for one session it loads only one and then it is cached, which also increases the performance of the application by n times. New images are loaded by load more data only. Good for multiple users running at same time.
Doing Tasks | 0 Comments cache cacheenabled AnalystList analystlistload performance rbx44, bgz82 8

#12 As a developer I want to load the thumbnails of the images instead of loading the whole image from MongoDB in AnalystList
Another fix to increase the overall experience of the application and fixing the crashing of application, instead of loading all the images from MongoDB, now it loads only the thumbnails of the application which significantly increases the load on the server and thus improves the performance of application more than 10 times.
Done Tasks | 0 Comments thumbnailsload AnalystList appcrash bugs rbx44, bgz82 5

#11 As a developer I want to resize the images taken by the observer to improve the speed of loading and retrieving in AnalystList
The AnalystList loading was taking too much time, for multiple users loading at same time, application was crashing. Resizing the image taken by observer before sending it to mongoDB to 600 by 600.
Done Tasks | 0 Comments imageresize analystlistload speed performance bugs None rbx44, bgz82 3

#10 As a developer, we want to address the bugs and performance issues to improve the overall experience of the application.
Due to the slow server and programming issues found in the application, application was not usable, bugs found in image size, loading of analyst list, observer taking pictures(status), multiple regions analyzing, coordinator, image holder.
Done Tasks | 0 Comments fixes bugs bgz82 3

#9 As a developer I want to thorough test the application manually and look for bugs and improvements
System testing, look for major bugs and improving the performance of the application as a whole, fixes and do the evaluation.
Done Tasks | 0 Comments bugs flaws fixes improvements finds None rbx44 13

Service Description

Existing Services

Data Layer

MongoDB, for backend storage.

We created the database ‘umkc’, as part of this project we created collection ‘green’.

To make the connection with the application and the server.

It makes the Http request to get the ID of the images.

Once the connection has been made and the ID is retrieved it makes another Http request with the MongoDB to get the actual image.

To get the text data associated with image it makes the Http get request to get the attributes(type, degree of damage associated with the image).

Using the Rest service we get the geo location of the images and we are visualizing it in google maps.

Schema – JSON Format fields explained below with an example:

| Field | Type | Description |
|--------------------------|-------------|---|
| _id and \$oid | JSON Object | The id that uniquely refers the object |
| sceneld | String | default filename is used here |
| imageData | Json Object | Image information like type and data is stored in JSON format |
| type(child of imageData) | String | image type - “JPEG, PNG” |
| data(child of imageData) | String | Image encoded as base64 string |
| description | String | Scene description |

| | | |
|------------------------------|------------------------|---|
| location | Array of double values | Latitude and longitude values stored in JSON array |
| results | JSON Array | Stores the region information and from the device from which it is marked |
| deviceId(child of imageData) | String | Network id used to identify the device uniquely |
| region(child of imageData) | JSON Array | Selected region information is stored |
| category | Integer | Used to find type of Object |
| damageLevel | Integer | Damage level of the selected region |
| boundary | JSON Object | Selected region in the scene |

Example

```
{
  "_id": {
    "$oid": "53d26a76e4b0ad6bce969780" },
    "scenId": "http://esridev.caps.ua.edu/MooreTornado/Images/Day3/Christine/
  IMG_4311.JPG", "imageData": {
    "type": "JPG",
    "data": "/9j/4AAQSkZJRgABAgAAAQABAAD/2wBDAcgcHiMeGSgjISMtKygwP-
  GRBD- c3PHtYXUlkkYCZlo+AjlqgtObDoKrarYqMyP/L2u71///m8H///6/+b9//j/.....
  },
    "description": "One completely damaged building", "location": [35.31747,
  -97.563394],
    "results": [
      "deviceId": "e8:99:c4:8f:ef:5d",
      "region": "[{"categroy":3,"damageLevel":1,"boundry":{"bottom":477.96045,"right":399.5773,"left":5.310051,"top":145.84058}},{"categroy":1,"damageLevel":0,"boundry":{"bottom":481.42346,"right":390.42566,"left":-23.818665,"top":154.68658}]}]"}}}
```

Services Implemented as part of Second Increment

Bugs:

Service #1

Analyst List : Loading the images in analyst list for analyzing them takes forever to load. Also, multiple user loading the list in their respective devices crashes the application.

Service #2

Slowness of application: We gave access to our application to 20 people and asked them for their feedback, most heard comments were about the slowness of application and overall experience of the application.

Service #3

Observer Image Capture: The main job of observer is to take pictures using the built-in camera application. However, we found two issues in observer.

- a. Analyzing the image taken by observer: Image uploaded to mongoDB manually were fine and functioning for analyst but images captured by observer were not, however loading them in mongoDB and retrieving them in analyst list was working fine.
- b. Exception handling with observer images: If observer takes images, they get stored in mongoDB but there was no way that shows the image was stored successfully and

also, if users doesn't take any picture and still tries to submit it, a null image object gets stored in mongoDB.

Service #4

Analyzing Multiple Regions: There was no way for analyst to analyze multiple regions in an image. Only one region of image can be analyzed. As we know, there can be multiple regions that analyst would want to analyze.

Service #5

Once we introduced Multiple Regions, there was no way to see the already analyzed regions in an image by the same analyst. Analyst should remember what regions he has analyzed already.

To fix this problem, we add multiple region view, which can be seen in the implementation section.

Service #6

Coordinator Map View: Getting the image data from mongoDB and displaying on Google Maps was not working. Coordinator can only see his current location but the image data is not loaded.

Service #7

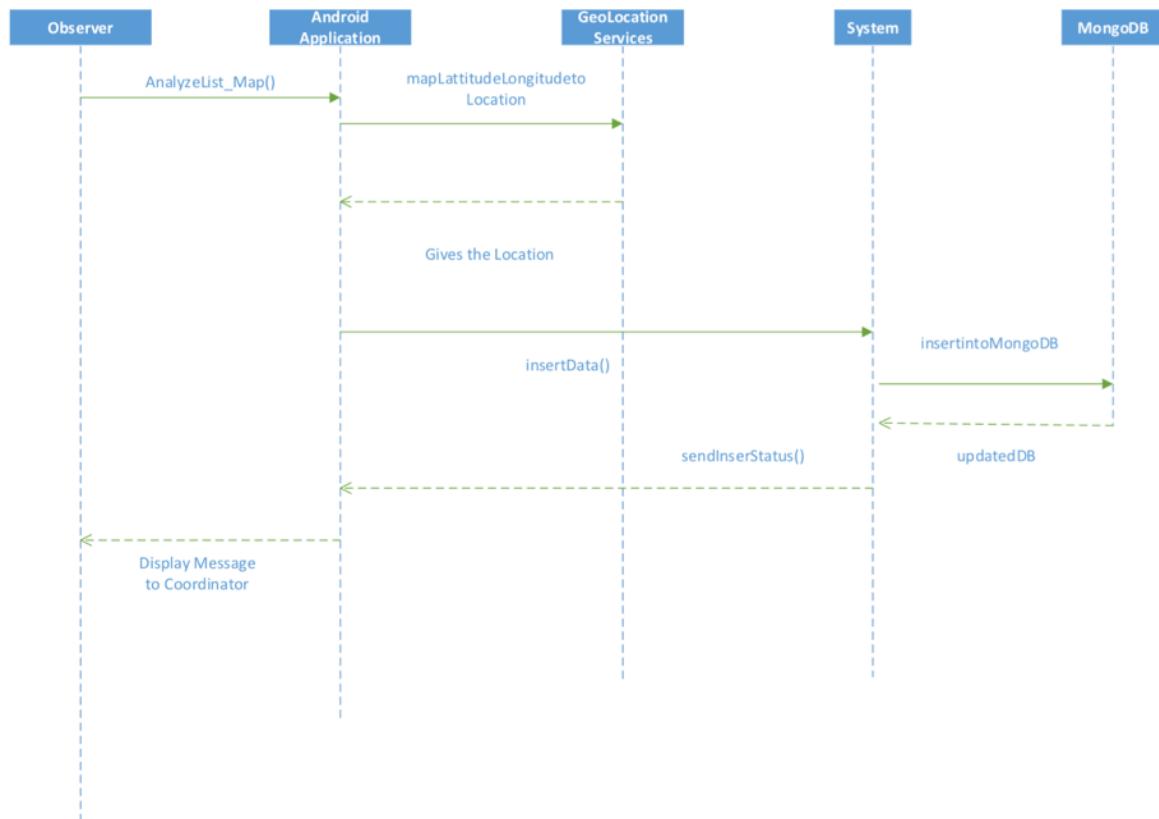
Image Holder View: Selecting the regions and playing with the square/ellipse to do the same and stretching/shrinking it was a pain. Never worked for first time.

Service #8

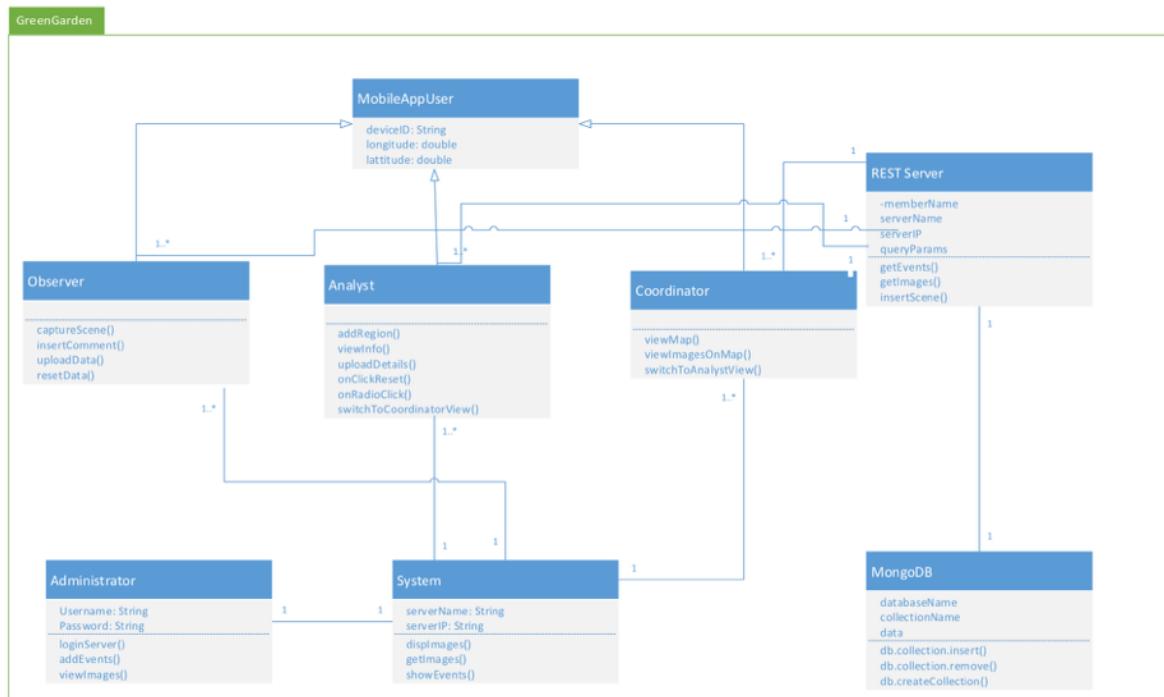
Add health status of an image to the analyst, and purpose a new machine learning solution to it. Analyze the health of

the green garden by checking the intensity of color of green garden in images based on temporal and spatial, automatically using supervised machine learning technique.

Sequence Diagram



Class Diagram



Design of Mobile Client Interface

Analyst: Responsible for analyzing a provided scene by identifying the type of objects, degree of damage, and marking boundaries.

Coordinator: Responsible for crowdsourcing management and decision-making based on the available scenes and crowdsourcing results.

Observer: Responsible for capturing and collecting the images of disaster scenes.



- Main Activity: This is the screen which will be displayed when the application is loaded. The options can be (a) Analyst, (b) Coordinator and (c) Observer. On Selecting Analyst will takes to Analyst List page.
- Analyze Item List: The user will be provided with the list of scenes. Users can switch to Co- coordinator page, load more scenes from the cloud and select a scene to analyze.
- Analyze A Scene: A scene can be analyzed with the detection techniques. We are still thinking what all methods will be and set the damage degree for each categorized object.
- Co-coordinator View: All the scenes are visualized on a Map with markers representing each scene at a particular location. Selecting a marker provides more details of the crowdsourced information for the selected scene. Once more details are listed, if the user wish to analyze the scene identifying different category objects, it can be done by selecting the Analyze button.
- Observer View: Provides the interface to capture the images. User can take picture. These pictures will be uploaded to the cloud automatically. We are thinking to use MongoDB at the backend.

Design of Unit Test Cases

NUnit

1. To retrieve the images from mongoDB
2. To retrieve the image IDs from mongoDB
3. Submit Observer Data to mongoDB

JUnit

For radio button for all events.

Implementation

Implementation of REST services

Rest service is implemented to connect to MongoDB.

Http Request/Response methods are used to pull and push the data.

Below is the example for connecting to MongoDB, and pulling the events data for the Spinner.

```

@Path("/cisa")
public class MyResource {

    @GET
    @Produces("text/plain")
    public String getIt() {
        return "Hi there!";
    }
    @GET
    @Path("events")
    @Produces("text/plain")
    public String getEventsInfo() throws JSONException
    {
        MongoClient mongo;
        JSONObject spinnerArray;
        //JSONArray sArray;
        String result[] = new String[100];
        String concat="";
        int i=0,j=0;
        try {
            mongo = new MongoClient("lasir.umkc.edu", 27017);
            DB db = mongo.getDB("umkc");
            DBCollection table = db.getCollection("events");
            DBCursor cursor = table.find();
            while (cursor.hasNext()) {
                spinnerArray=new JSONObject((cursor.next().toString()));
                result[i]=spinnerArray.get("name").toString();
                i++;
            }
        } catch (UnknownHostException e) {
            e.printStackTrace();
        }
        for(j=0;j<i;j++)
        {
            concat=concat + result[j] + ";";
        }
        return concat;
    }
}

```

To make the application efficient, we thought of loading only 6 images at a time. To implement the same, we used REST. Here we are making the connection with the Server using REST to pull 6 images at one time. Below is the implementation.

```

int start=0;
String[] ids = new String[6];
try
{
    URL dest = new URL("http://lasir.umkc.edu:8080/greengarden/webresources/ggarden/sceneids?start="+start+"&limit=6");
    URLConnection yc = dest.openConnection();
    BufferedReader in = new BufferedReader(new InputStreamReader(yc.getInputStream()));
    String inputLine="";
    String getData="";
    int i=0;
    while ((inputLine = in.readLine()) != null)
    {
        getData+=inputLine;
        i++;
    }
    in.close();
    i=0;
    StringTokenizer st=new StringTokenizer(getData, " ");
    while(st.hasMoreTokens())
    {
        ids[i]="http://lasir.umkc.edu:8080/greengarden/webresources/ggarden/image/"+st.nextToken();
        i++;
    }
}
catch(Exception e)
{
    out.println(e);
}
start=start+6;
%>

```

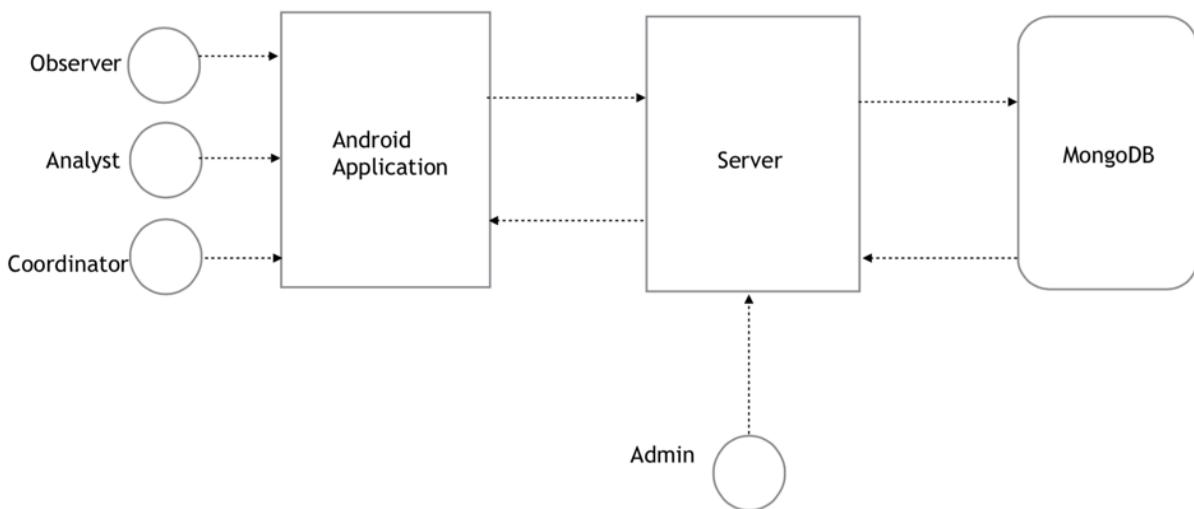
We are using AJAX to implement the next & previous button to load the next six images or the previous six images. Connection is made through REST service and we also implemented the design using Jssor_slider.

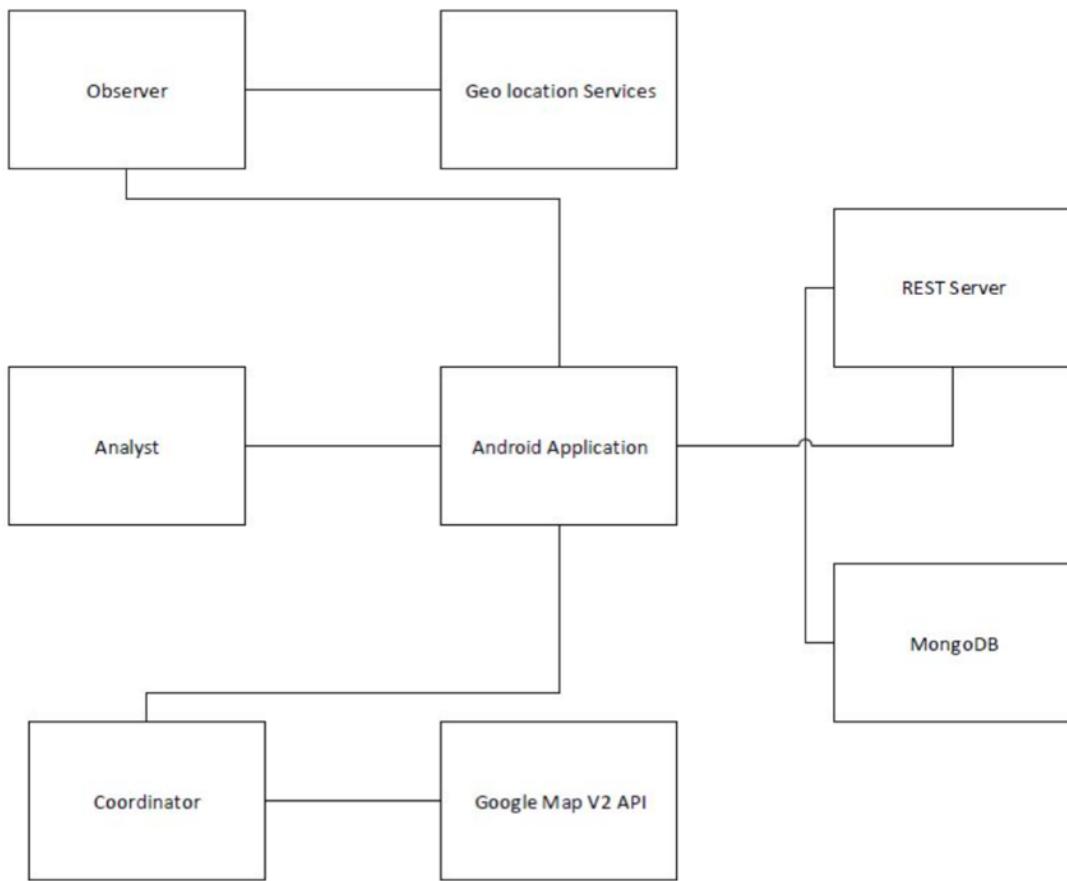
```

<script>
    jssor_slider1_starter('slider1_container');
</script>
</div>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<script type="text/javascript">
function getNextImages()
{
    var ids;
    var check;
    var starts = document.getElementById("start").value;
    console.log(starts);
    $.ajax({
        url: "http://lasir.umkc.edu:8080/greengarden/webresources/ggarden/sceneids?start=" + starts + "&limit=6",
        type: 'get',
        async: false
    })
    .done( function (data, status) {
        ids = data.split(', ');
        var i=0;
        if(data == "")
        {
            alert("End of Images");
            check="0";
            return;
        }
        while(i<ids.length)
        {
            ids[i] = "http://lasir.umkc.edu:8080/greengarden/webresources/ggarden/image/" + ids[i];
            i++;
        }
        console.log(ids.length);
    })
    .fail( function (data, status) {
    });
    if(check == "0")
    {
        return;
    }
    document.getElementById("start").value = parseInt(starts) + 6;
    document.getElementById("image1").src = ids[0];
    document.getElementById("image2").src = ids[1];
}

```

Implementation of User Interface





Implementation of Existing Services

1. The goal of this project is to build an android application which has the following the system features.

Analyst: Responsible for analyzing a provided scene by identifying the type of objects, degree of damage, and marking boundaries. **Coordinator:** Responsible for crowdsourcing management and decision-making based on the available scenes and crowdsourcing results. **Observer:** Responsible for capturing and collecting the images of disaster scenes.

2. Add roles and functionalities for the different types of users in this application, including Observer, Analyst and Coordinator.
3. Install Apache Tomcat, update Java, setup Mongo, upload data to Mongo, connect to Application
4. To implement REST service to connect to the server and MongoDB.
5. Use the server memory efficiently by loading only six images at a time. User can move to the next six images using the slider option.
6. Using Web Page Interface , provide an option to administrator to add the events.
7. Login page for administrator and validate the credentials and move to appropriate page.
8. Add events as a spinner item to the Android Application and get the events from MongoDB and add the items to spinner.

Services Implemented as part of Second Increment

As explained in Service Description, there are 8 services we thought of implementing as a part of our second increment.

Fixing Bugs

Service #1

Problem Statement

Analyst List : Loading the images in analyst list for analyzing them takes forever to load. Also, multiple user loading the list in their respective devices crashes the application.

Solution Proposed:

1. Image Resizing: We figured out that there is a limit of network transaction from any android application, which is 1MB and we were running the application on devices with 16 megapixels camera, so the size of these images taken were too large and thus it was exceeding the limitation of network transaction. To fix this issue, once the image is taken and before it is been sent to the server, we are resizing the image into 600 by 600. This fixed the issue of network transaction in android, however, the application was still buggy and slow.
2. Thumbnails Load: Our second approach to make the application faster was to do the thumbnails loading. Earlier in analyst list, it was retrieving entire images with their metadata into analystList. We figured out a way to make

this application faster, instead of loading images, it now only loads thumbnails of images in analyze list. By doing this, it fixed this major bug and made it pretty faster. Performance evaluation will be done later.

Code:

Image Resize

```
// Code to reduce the image size and encode it to Base64 format

public static void setImage(String url) {

    System.out.println("IN NOTOB");
    File file;
    String encoded;
    byte [] encodeByte,b;
    if (image == null) {
        file = new File(url);
        //byte[] bFile = new byte[(int) file.length()];

        try {
            encoded = com.cisa.androidapp.utils.Base64.encodeFromFile(url);
            // System.out.println(encoded);
            encodeByte=com.cisa.androidapp.utils.Base64.decode(encoded);
            Bitmap bitmap=BitmapFactory.decodeByteArray(encodeByte, 0, encodeByte.length);
            Bitmap insertImage = ThumbnailUtils.extractThumbnail(bitmap, 600, 600);
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            insertImage.compress(Bitmap.CompressFormat.JPEG, 100, baos);
            b = baos.toByteArray();
            encoded = com.cisa.androidapp.utils.Base64.encodeBytes(b);
            image = new Image(encoded, getFileExtension(url));
            b = null;
            bitmap = null;
            insertImage = null;
        }
    }
}
```

```

        catch (IOException e) {
            e.printStackTrace();
        }
    }
    else
    {
        }
    }
}

```

Thumbnails

//Convert the images to thumbnails in the analyst view holder

```

    public View getView(int position, View convertView, ViewGroup parent) {
        ViewHolder holder = null;
        if (convertView == null) {
            convertView = getLayoutInflater().inflate(
                R.layout.item_info, parent, false);
        }

```

```

        holder = new ViewHolder();
        holder.image = (ImageView) convertView
            .findViewById(R.id.imageicon);
        holder.locality = (TextView) convertView
            .findViewById(R.id.locality);
        Scene currentScene = sceneList.get(position);
        byte[] encodeByte;
        try {
            encodeByte = Base64.decode(currentScene.getImageData()
                .getData());
            bitmap = BitmapFactory.decodeByteArray(encodeByte, 0,
                encodeByte.length);
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        //ThumbnailUtils thumb;

```



```
Bitmap thumbNail = ThumbnailUtils.extractThumbnail(bitmap, 64, 64);
holder.image.setImageBitmap(thumbNail);
double lat = currentScene.getLocation()[0];
double lon = currentScene.getLocation()[1];

Geocoder mGeocoder = new Geocoder(this.getContext());
List<Address> addresses = null;
try {
    addresses = mGeocoder.getFromLocation(lat, lon, 1);
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

String addressText = null;

if (addresses != null && addresses.size() > 0) {
    // Get the first address
    Address address = addresses.get(0);
    /*
     * Format the first line of address (if available), city,
     * and country name.
     */
    addressText = String.format(
        "%s, %s, %s",
        // If there's a street address, add it
        address.getMaxAddressLineIndex() > 0 ? address
            .getAddressLine(0) : " ",
        // Locality is usually a city
        address.getLocality(),
        // The country of the address
        address.getCountryName());
    // Return the text
    // return addressText;
} else {
    // return "No address found";
}
holder.locality.setText(addressText);
return convertView;
}}
```

Service #2

Problem Statement:

Slowness of application: We gave access to our application to 20 people and asked them for their feedback, most heard comments were about the slowness of application and overall experience of the application.

Proposed Solution:

Caching: For the first time, we introduced caching in android application, which is a integral part of android development but we had issues using it. After exploring and reading papers about the same, we figured out these issues and fixed them. Using Caching increased the overall experience, instead of loading the thumbnails again and again from mongoDB, it now loads it only once and for some time, these thumbnails are cached and shown automatically. However, no changes were made on how new data is loaded. Caching is still in “doing” phase.

Code: Integral part of android development.

Service #3

Problem Statement

Observer Image Capture: The main job of observer is to take pictures using the built-in camera application. However, we found two issues in observer.

- a. Analyzing the image taken by observer: Image uploaded to mongoDB manually were fine and functioning for analyst but images captured by observer were not, however loading them in mongoDB and retrieving them in analyst list was working fine.
- b. Exception handling with observer images: If observer takes images, they get stored in mongoDB but there was no way that shows the image was stored successfully and also, if users doesn't take any picture and still tries to submit it, a null image object gets stored in mongoDB.

Proposed Solution:

To fix these issues, image resizing(explained earlier) and code review was done, some changes were done in code how the images are retrieved from the list and analyzed.

Unless the image is captured, we don't store it anywhere and also, once the images is stored it sends back the (ack) status message back from MongoDB to Observer.

Code:

```
// Code to handle multiple clicks for the image capture (avoids the application crash)
public void onClickCapture(View v {

    capture_status.setText("scene captured");
    capture_status.setVisibility(View.VISIBLE);
    capture_status.setAlpha(0.6f);
    image_capture.setText("Captured");
    mCamera.takePicture(null, null, new MyPictureCallback(
        ObserverActivity.this));
    image_capture.setEnabled(false);
```

```
}
```

```
// Feedback to the observer after successful insertion of data into MongoDB

public void uploadData(View v) {
if(Utils.image != null){
if (scene != null) {
scene.setSceneId("-1");
scene.setImageData(Utils.getImage());
scene.setDescription(e_comment.getText().toString());
System.out.print(scene.getDescription());
System.out.println(scene);

double latlon[] = { 56.67, 23.223 };

/* Test API */

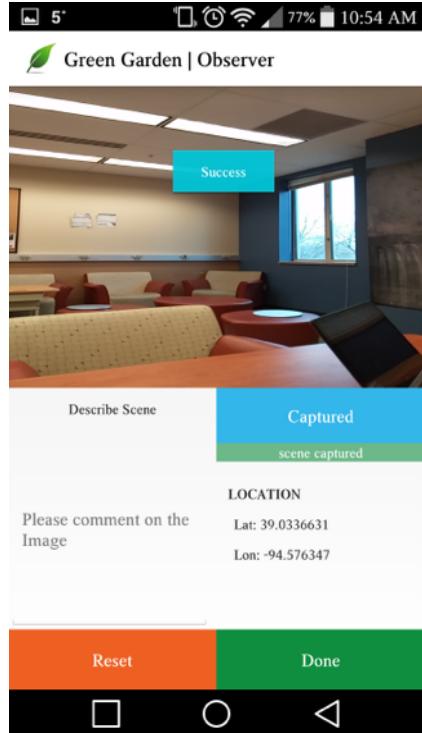
Boundary b = new Boundary(10, 20, 30, 34);
Region reg = new Region(9, b);
ArrayList<Region> al = new ArrayList<Region>();
al.add(reg);
al.add(new Region(2, b));
/* Analysis analysis = new Analysis("Android", al, latlon);

ArrayList<Analysis> analys = new ArrayList<Analysis>();
analys.add(analysis);
*/
Gson gson = new Gson();
gson.toJson(scene);

System.out.println("JSON SCENE ARRAY " + gson.toJson(scene));
AsyncTask<String, Void, String> a = new HttpAsyncService(ObserverActivity.this).execute(String.valueOf(Utils.UploadInfo),
Utils.UPLOAD_URL, gson.toJson(scene));
try {
Toast.makeText(getApplicationContext(), a.get().toString(),
Toast.LENGTH_SHORT).show();
Utils.image = null;
```

```
} catch (InterruptedException e) {  
    e.printStackTrace();  
} catch (ExecutionException e) {  
    e.printStackTrace();  
}  
}  
}
```

(Screenshot shows the Success statement)



Service #4 & 5

Problem Statement:

Analyzing Multiple Regions: There was no way for analyst to analyze multiple regions in an image. Only one region of image can be analyzed. As we know, there can be multiple regions that analyst would want to analyze.

Once we introduced Multiple Regions, there was no way to see the already analyzed regions in an image by the same analyst. Analyst should remember what regions he has analyzed already.

Proposed Solution

We introduced analyzing multiple regions, now analyst can analyze multiple regions in any image.

We add multiple region view, which can be seen in the implementation section.

Code

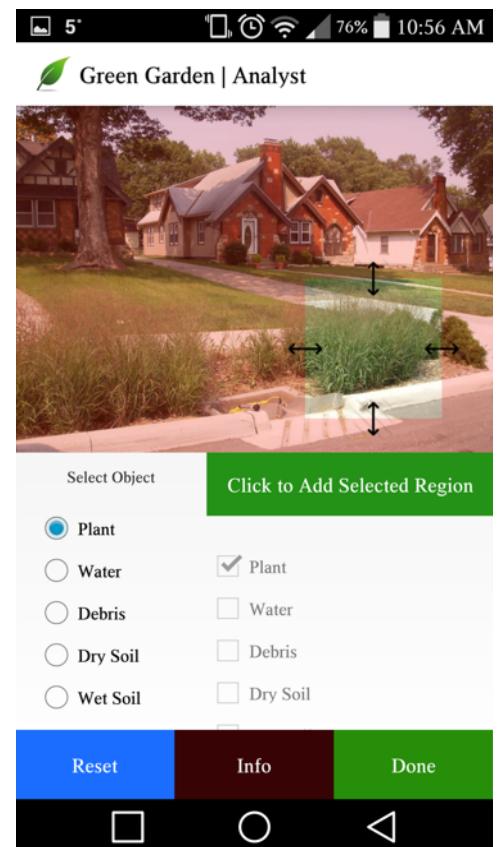
```
//Code to select Multiple regions and add it to them to the List item and upload it to  
the MongoDB  
public void addRegion(View view) {  
    RectF rect = cropView.getRect();  
  
    radioGroup = (RadioGroup) findViewById(R.id.radioObject);  
    int id = radioGroup.getCheckedRadioButtonId();  
    int type = 1;  
    String typeText= " ";  
    switch (id) {  
        case R.id.radioPlant:  
            type = Utils.REGION_PLANT;  
            typeText = "Plant";  
            break;  
        case R.id.radioWater:  
            type = Utils.REGION_WATER;  
            typeText = "Water";  
            break;  
        case R.id.radioDebris:  
            type = Utils.REGION_DEBRIS;  
            typeText = "Debris";  
            break;  
        case R.id.radioDry:  
            type = Utils.REGION_DRYSOIL;  
            typeText = "Dry Soil";  
            break;  
        case R.id.radioWet:  
            type = Utils.REGION_WETSOIL;  
            typeText = "Wet Soil";  
            break;  
        default:  
            break;  
    }  
    final Region region = new Region(type, new Boundary(rect.left, rect.top, rec-  
t.right, rect.bottom));
```

```

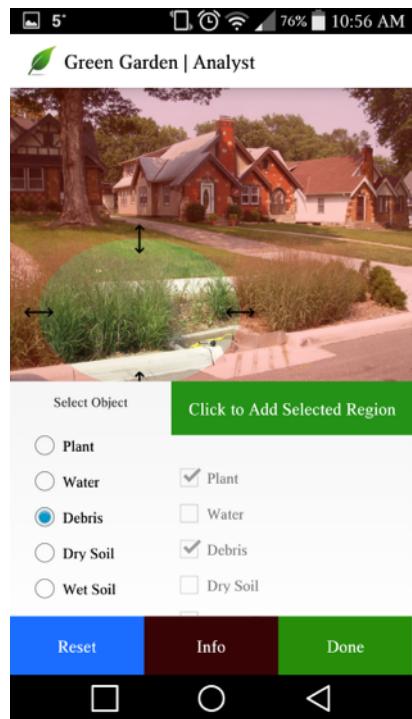
selectedRegions.add(region);
CheckBox mCheckBox;
if(typeText.contains("Plant"))
{
    mCheckBox = (CheckBox) findViewById(R.id.plant);
    mCheckBox.setChecked(true); //to check
}
else if(typeText.contains("Water"))
{
    mCheckBox = (CheckBox) findViewById(R.id.water);
    mCheckBox.setChecked(true);
}
else if(typeText.contains("Debris"))
{
    mCheckBox = (CheckBox) findViewById(R.id.debris);
    mCheckBox.setChecked(true);
}
else if(typeText.contains("Dry"))
{
    mCheckBox = (CheckBox) findViewById(R.id.drysoil);
    mCheckBox.setChecked(true);
}
else if(typeText.contains("Wet"))
{
    mCheckBox = (CheckBox) findViewById(R.id.wetsoil);
    mCheckBox.setChecked(true);
}

```

(Screenshot shows the plant has been already analyzed, another instance of event plant is being analyzed by analyst, analyst can select any events and analyze an image many times, layout tells the status of things analyzed)



(Now Debris has also been analyzed)



Service #6

Problem Statement

Coordinator Map View: Getting the image data from mongoDB and displaying on Google Maps was not working. Coordinator can only see his current location but the image data is not loaded.

Proposed Solution

We added more functionality for Coordinator in Map View, he can now load the images from its location in map. Implementation can be seen in implementation section.

Code

```
// Code to get the image data from the MongoDB

protected String doInBackground(String... params) {
    ByteArrayOutputStream out = null;
    HttpClient client = new DefaultHttpClient();
    String url = Utils.HOST + Utils.RES_LIST_ITEMS + "?start=" + sceneList.size() +
    "&limit=20";
    System.out.println(" Url: " + url);
    HttpGet get = new HttpGet(url);
    HttpResponse response;
    try {
        response = client.execute(get);
        out = new ByteArrayOutputStream();
        response.getEntity().writeTo(out);
        out.close();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        get.abort();
    }
    return out.toString();
}
// Post processing after retrieving the data

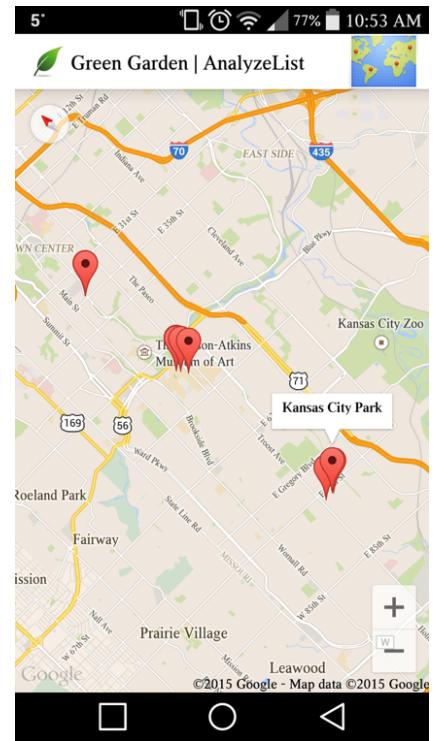
@Override
protected void onPostExecute(String result) {
    dialogShow = false;
    System.out.println("Data from server: " + result);
    if(populateSceneData(result)){
        //Choose the required data to display on map using below function call
        populateMarkersToMap();
        MapsInitializer.initialize(obj.getActivity());
    }

    // Updates the location and zoom of the MapView
    System.out.println("Before CAMMAP");
    CameraUpdate cameraUpdate = CameraUpdateFactory.newLatLngZoom(new
    LatLng(sceneList.get(0).getLocation()[0], sceneList.get(0).getLocation()[1]), 15);
    mMap.animateCamera(cameraUpdate);
```

```
//return obj.rootView;
}

// Populate the required data from the retrieved data

private boolean populateSceneData(String data) {
if (data == null) {
//listView.removeFooterView(loadMoreView);
return false;
}
JSONArray jsonList;
try {
jsonList = new JSONArray(data);
if (jsonList.length() == 0) {
loadmore = false;
//listView.removeFooterView(loadMoreView);
} else {
if (jsonList.length() < 10) {
//listView.removeFooterView(loadMoreView);
loadmore = false;
}
for (int i = 0; i < jsonList.length(); i++) {
JSONObject sceneJSON = jsonList.getJSONObject(i);
// Image Mapping
Image image = new Gson().fromJson(sceneJSON.getString("imageData"),
Image.class);
// Location Mapping
String sceneld = sceneJSON.getString("_id");
String locationString = sceneJSON.getString("location");
double location[] = extractLocation(locationString);
// Description Mapping
String description = sceneJSON.getString("description");
Scene scene = new Scene(sceneld, image, description, location);
sceneList.add(scene);
}
} catch (JSONException e) {
e.printStackTrace();
}return true; }
```



(Screenshot shows the image data loaded to map and image description too, for instance here Kansas City Park)

Service #7

Problem Statement

Image Holder View: Selecting the regions and playing with the square/ellipse to do the same and stretching/shrinking it was a pain. Never worked for first time.

Proposed Solution

Changed the resizing style and holder view for selecting and analyzing the regions. Implementation can be seen in implementation section.

Code

```
private class ViewHolder {
```

```
    ImageView image;
```

```
    TextView locality;
```

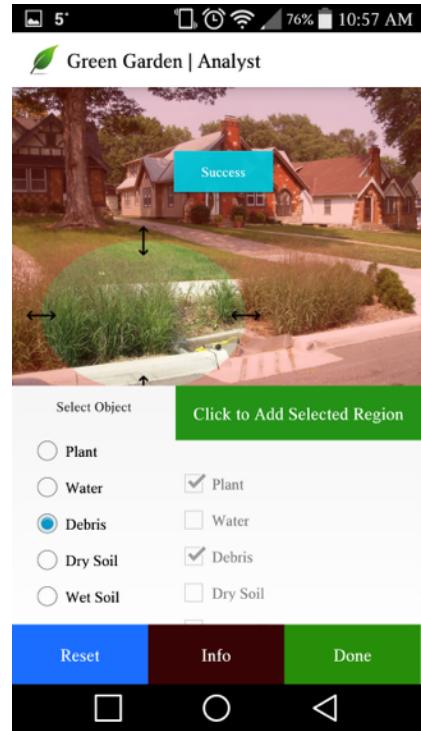
```

}

holder = new ViewHolder();
holder.image = (ImageView) convertView
    .findViewById(R.id.imageicon);
holder.locality = (TextView) convertView
    .findViewById(R.id.locality);

```

(Screenshot shows the resizing tools, more clear and they are easy to use)



Service #8

Problem Statement

Add health status of an image to the analyst, and purpose a new machine learning solution to it. Analyze the health of the green garden by checking the intensity of color of green garden in images based on temporal and spatial, automatically using supervised machine learning technique.

Not yet implemented
In “Doing” phase.

Results using our machine learning technique

| Naive Bayes | Decision Tree | Decision Table | Part | Data Set(Size) | Experiments |
|-------------|---------------|----------------|--------|----------------|-------------------------|
| 91.67% | 91.67% | 91.67% | 91.67% | 5 images | |
| 96.20% | 97.47% | 96.20% | 97.47% | 80 images | |
| 90.80% | 90.80% | 90.80% | 90.80% | 80 im | With Dead garden Images |

Test Cases & Performance Evaluation

System Testing: Manually test all functions and check for bugs, in case bugs are found, debugging is done to check for bugs and fixes are proposed and implemented.

Unit Testing: We performed both NUnit & JUnit testing
NUnit testing to test the services

Test cases generated and tested for,

1. To retrieve the images from mongoDB(REST)
2. To retrieve the image IDs from mongoDB(REST)
3. Submit Observer Data to MongoDB(REST)

Evaluation:

Application was installed on 15 phones and it was used by 15 people at same time, no major bugs and performance issue found except the analyst list image loading takes time when all 15 phones are pulling the data at same time. It is mainly because of server limitation(4GB memory). All the major bugs from increment one were found and fixed, performance of application got improve drastically.

Testing

Unit test cases

NUnit

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using NUnit.Framework;
using System.IO;
namespace Testws
{
    [TestFixture]
    public class Service
    {
        [Test]
        public void restGetImages()
        {
            //To retrieve the images

            WebRequest req = WebRequest.Create(@"http://lasir.umkc.edu:8080/green-
garden/service/webresources/ggarden/itemslist?start=0&limit=1");
            req.Method = "GET";
            req.ContentType = "application/json";
            req.Method = "GET";
            //req.ContentLength = 0;
            HttpWebResponse resp = req.GetResponse() as HttpWebResponse;
            if (resp.StatusCode == HttpStatusCode.OK)
            {
                using (Stream respStream = resp.GetResponseStream())
                {
                    StreamReader reader = new StreamReader(respStream, Encoding.UTF8);
                    Assert.AreEqual(HttpStatusCode.OK, resp.StatusCode);
                }
            }
        }
    }
}
```

```

}

[TestMethod]
public void restGetImageIds()
{
    // To retrieve the Image IDs
    WebRequest req = WebRequest.Create(@"http://lasir.umkc.edu:8080/green-
garden/webresources/ggarden/sceneids?start=0&limit=1");
    req.Method = "GET";
    req.ContentType = "application/json";
    //req.Method = "POST";
    //req.ContentLength = 0;
    HttpWebResponse resp = req.GetResponse() as HttpWebResponse;
    if (resp.StatusCode == HttpStatusCode.OK)
    {
        using (Stream respStream = resp.GetResponseStream())
        {
            StreamReader reader = new StreamReader(respStream, Encoding.UTF8);
            Assert.AreEqual("54f3eed2e4b0ceb891145128", reader.ReadToEnd());
        }
    }
}

[TestMethod]
public void postObserverData()
{
    //Post the Observer Data to MongoDB

    string data = "{ color: \"red\", value: \"#f00\" }";
    WebRequest req = WebRequest.Create(@"http://lasir.umkc.edu:8080/green-
gardenservice/webresources/ggarden/observerdata");
    req.Method = "POST";
    var data1 = Encoding.ASCII.GetBytes(data);
    req.ContentLength = data1.Length;
    req.ContentType = "application/json";
    using (var stream = req.GetRequestStream())
    {
        stream.Write(data1, 0, data1.Length);
    }
}

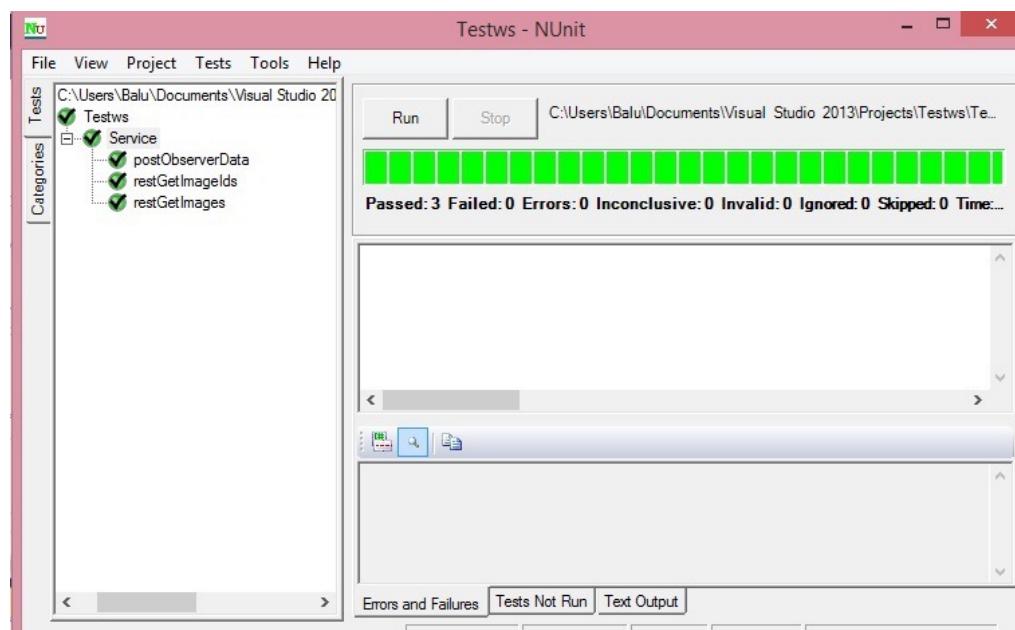
```

```

HttpWebResponse resp = req.GetResponse() as HttpWebResponse;
if (resp.StatusCode == HttpStatusCode.OK)
{
    using (Stream respStream = resp.GetResponseStream())
    {
        StreamReader reader = new StreamReader(respStream, Encoding.UTF8);
        Assert.AreEqual(HttpStatusCode.OK, resp.StatusCode);
    }
}
else
{
    Assert.AreEqual(null, resp.StatusCode);
}

}
}
}

```



JUnit

```
package com.cisa.androidapp.test;

import java.util.ArrayList;
import java.util.List;

import android.test.ActivityInstrumentationTestCase2;
import android.widget.RadioButton;
import android.widget.TextView;

import com.cisa.androidapp.*;
public class Test1 extends ActivityInstrumentationTestCase2<MainActivity> {

    private MainActivity mActivity;
    RadioButton rb;
    private String resourceString;

    public Test1(Class<MainActivity> activityClass) {
        super(activityClass);
        // TODO Auto-generated constructor stub
    }
    @Override protected void setUp() throws Exception {
        super.setUp();
        mActivity = this.getActivity();
        List<String> names = new ArrayList<String>();
        rb = (RadioButton) mActivity.findViewById(com.cisa.androidapp.R.id.plant);
        rb.setChecked(true);

    }
    public void testPreconditions() {
        assertNotNull(rb); }
    public void testText() {
        assertEquals(1,rb.isChecked());
    }
}
```

Deployment

ScrumDo

<https://www.scrumdo.com/projects/project/dscrowd/iteration/121774>

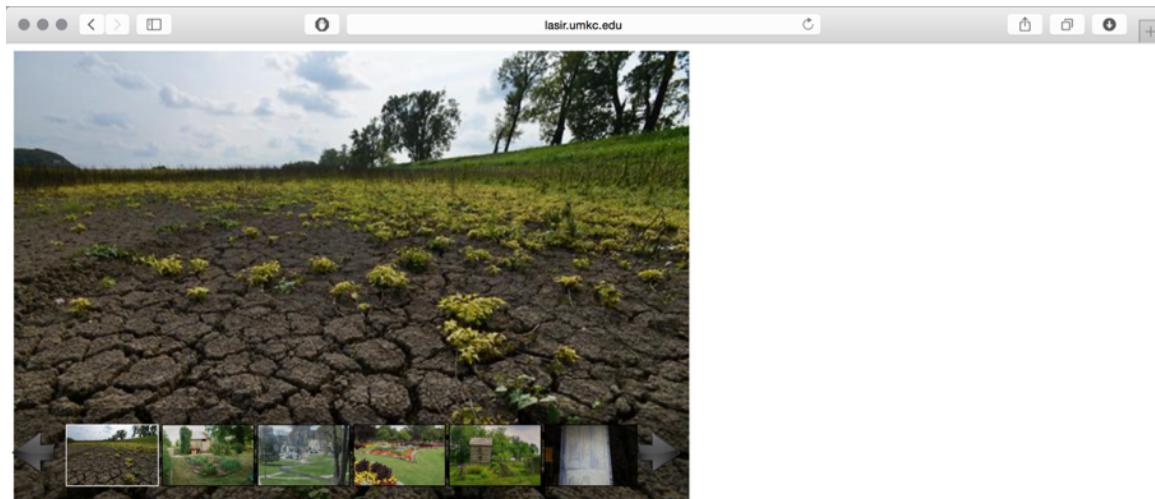
<http://lasir.umkc.edu:8080/>

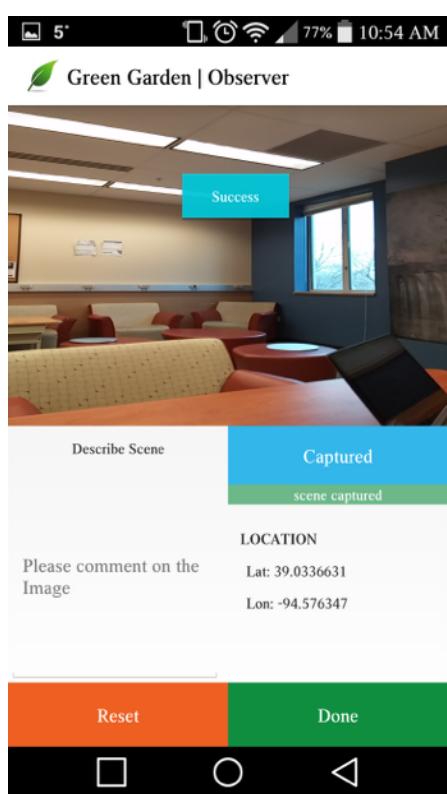
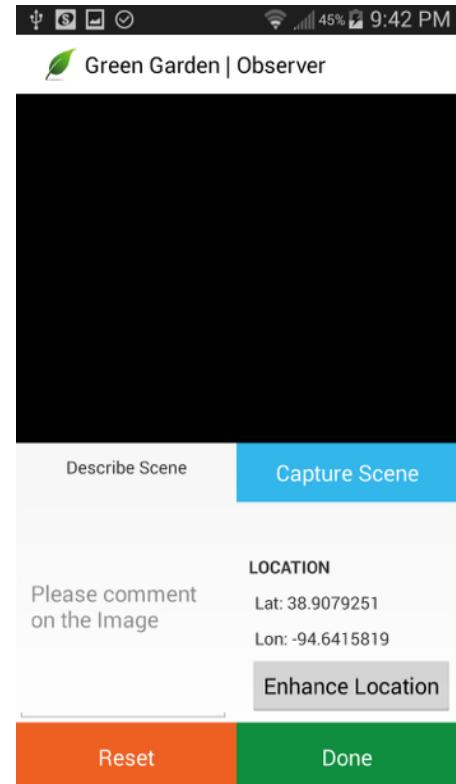
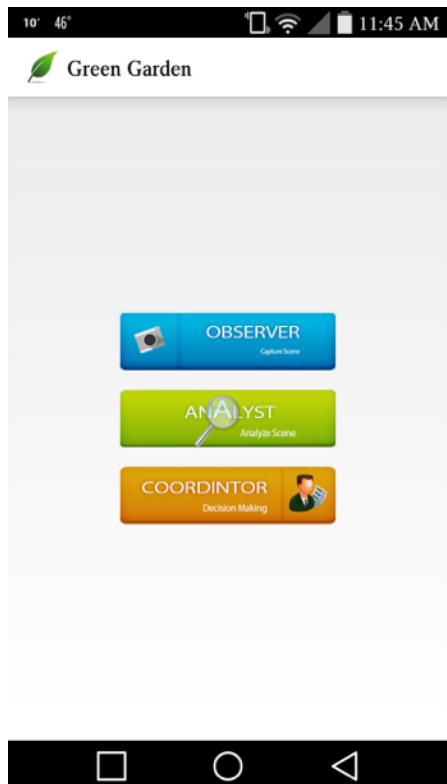
Mobile App Website

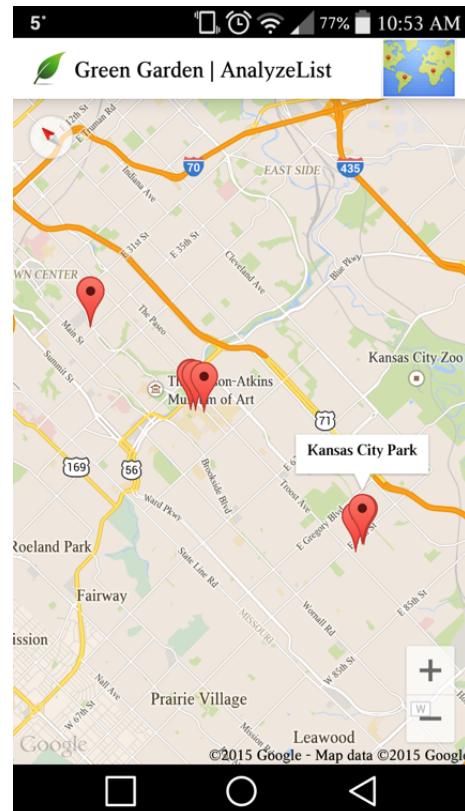
<http://lasir.umkc.edu:8080/greengarden/>

Report

Screenshots







ScrumDo Search Project ASE-Project DSCROWD rbx44

Increment 2 - Feb. 25, 2015 - March 18, 2015

| Todo | Doing | Reviewing | Done |
|---|---|---|---|
| 1 40 | 1 8 | 2 33 | 8 58 |
| Developer I want to add an image to the analyst, a new machine learning solution to it. | #13 As a developer we want to do introduce the caching for the first time. rbx44, bgz82 cache cacheenabled AnalystList analystload performance 0 Comments - Tasks | #20 As a developer I want to write test cases for the application to do unit & integration testing. rbx44, bgz82 junittesting testing testcases 0 Comments - Tasks | #19 As a developer I want to test the performance of the application using 15 phones at same time. rbx44, bgz82 systemtesting performanceevaluation 0 Comments - Tasks |
| gz82 g gardenhealth greenish tensity histogram analyst ility Tasks | 40 | 8 | 5 |
| | | 13 | 13 |
| | | | 8 |
| | | | |

Quick Links

- Project Summary
- Epics
- Iteration Planning
- Chat
- History
- Predictions
- Planning Poker
- Iterations
- Backlog 0 stories
- Increment 4 0 stories
- Increment 3 0 stories
- Increment 2 12 stories
- Increment 1 8 stories

Stories

#20 As a developer I want to write test cases for the application to do unit & integration testing.
Unit testing by writing jUnit Test Cases for various test cases
Reviewing Tasks | 0 Comments junittesting testing testcases □ Testing rbx44, bgz82 20

#19 As a developer I want to test the performance of the application using 15 phones at same time.
System testing was done to do the thorough evaluation of the performance by installing the application on 15 phones and running at some time. No major flaws found except analyst list loading getting slower due to the server limitation
Done Tasks | 0 Comments systemtesting performanceevaluation rbx44, bgz82 8

#18 As a developer I want to add health_status of an image to the analyst, and purpose a new machine learning solution to it.
Analyze the health of the green garden by checking the temporal & intensity of color of green garden in images automatically using supervised machine learning technique.
Todo Tasks | 0 Comments machinelearning gardenhealth greenish temporal intensity histogram analyst □ Functionality rbx44, bgz82 40

#17 As a developer I want to change the image holder view to select the regions in an image to analyze.
Selecting the regions and playing with holder to stretch or shrink to an particular region was difficult, We changed the holder view, which makes it easy to select regions easily for analyst.
Done Tasks | 0 Comments Imageholder selectregions analyst □ Bugs rbx44, bgz82 5

#16 As a developer, I want to fix Coordinator loading Google maps problem and add functionality to show where the images were taken in the map and load them.
Previously Coordinator was broken, Coordinator can now view the images and their location in google map and load the images from there.
Reviewing Tasks | 0 Comments googlemaps loadimages coordinator □ Bugs rbx44, bgz82 13

#15 As a developer I want to add functionality to Analyst to analyze multiple regions in an image and provide a layout which makes it easier to represent.
Earlier, Analyst were able to analyze only one part of the image. Analyst can analyze multiple regions in an image and can view the already analyzed regions in an image in the side, no need to remember what you did.
Done Tasks | 0 Comments multipleregions addlayout analyzeregions analyst □ Functionality rbx44, bgz82 13

#14 As a developer, I want to fix the Observer image capture and also add Status(ack back from Mongo) to make sure that the image was loaded to MongoDB.
Previously, only the images loaded into mongoDB were analyzed, analyzing the image captured by the application crashed the application.
Done Tasks | 0 Comments bugs observerimage codeerror □ None rbx44, bgz82 8

#13 As a developer we want to do introduce the caching for the first time.
Caching helps from not loading the thumbnails all the time, for one session it loads only one and then it is cached, which also increases the performance of the application by n times. New images are loaded by load more data only. Good for multiple users running at same time.
Doing Tasks | 0 Comments cache cacheenabled AnalystList analystlistload performance □ rbx44, bgz82 8

#12 As a developer I want to load the thumbnails of the images instead of loading the whole image from MongoDB in AnalystList
Another fix to increase the overall experience of the application and fixing the crashing of application, instead of loading all the images from MongoDB, now it loads only the thumbnails of the application which significantly increases the load on the server and thus improves the performance of application more than 10 times.
Done Tasks | 0 Comments thumbnailload AnalystList appcrash bugs □ rbx44, bgz82 5

#11 As a developer I want to resize the images taken by the observer to improve the speed of loading and retrieving in AnalystList
The AnalystList loading was taking too much time, for multiple users loading at same time, application was crashing. Resizing the image taken by observer before sending it to mongoDB to 600 by 600.
Done Tasks | 0 Comments imageresize analystlistload speed performance bugs □ None rbx44, bgz82 3

#10 As a developer, we want to address the bugs and performance issues to improve the overall experience of the application.
Due to the slow server and programming issues found in the application, application was not usable, bugs found in image size, loading of analyst list, observer taking pictures(status), multiple regions analyzing, coordinator, image holder.
Done Tasks | 0 Comments fixes bugs □ rbz82 3

#9 As a developer I want to thorough test the application manually and look for bugs and improvements
System testing, look for major bugs and improving the performance of the application as a whole, fixes and do the evaluation.
Done Tasks | 0 Comments bugs flaws fixes improvements finds □ None rbx44 13

Implementation Status Report

Work Completed

Story #9: As a developer I want to thorough test the application manually and look for bugs and improvements.

System testing, look for major bugs and improving the performance of the application as a whole, fixes and do the evaluation.

Status: Done

Responsibility: Rishabh

Time Taken: 2 Hours

Contributions: Rishabh(80%) & Bhargava(20%)

Story #10: As a developer, we want to address the bugs and performance issues to improve the overall experience of the application.

Due to the slow server and programming issues found in the application, application was not usable, bugs found in image size, loading of analyst list, observer taking pictures(status), multiple regions analyzing, coordinator, image holder.

Status: Done

Responsibility: Bhargava

Time Taken: 2 Hours

Contributions: Bhargava(80%) & Rishabh(20%)

Story #11: As a developer I want to resize the images taken by the observer to improve the speed of loading and retrieving in AnalystList

The AnalystList loading was taking too much time, for multiple users loading at same time, application was crashing. Resizing the image taken by observer before sending it to mongoDB to 600 by 600.

Status: Done

Responsibility: Rishabh & Bhargava

Time Taken: 3 Hours

Contributions: Bhargava(60%) & Rishabh(40%)

Story #12: As a developer I want to load the thumbnails of the images instead of loading the whole image from MongoDB in AnalystList.

Another fix to increase the overall experience of the application and fixing the crashing of application, instead of loading all the images from MongoDB, now it loads only the thumbnails of the application which significantly increases the load

on the server and thus improves the performance of application more than 10 times.

Status: Done

Responsibility: Rishabh & Bhargava

Time Taken: 5 Hours

Contributions: Rishabh(60%) & Bhargava(40%)

Story #14 As a developer, I want to fix the Observer image capture and also add Status(ack back from Mongo) to make sure that the image was loaded to MongoDB.

Previously, only the images loaded into mongoDB were analyzed, analyzing the image captured by the application crashed the application.

Status: Done

Responsibility: Bhargava & Rishabh

Time Taken: 3 Hours

Contributions: Bhargava(50%) & Rishabh(50%)

Story #15 As a developer I want to add functionality to Analyst to analyze multiple regions in an image and provide a layout which makes it easier to represent.

Earlier, Analyst were able to analyze only one part of the image. Analyst can analyze multiple regions in an image and can view the already analyzed regions in an image in the side, no need to remember what you did.

Status: Done

Responsibility: Bhargava & Rishabh

Time Taken: 6 Hours

Contributions: Bhargava(50%) & Rishabh(50%)

Story #16 As a developer, I want to fix Coordinator loading Google maps problem and add functionality to show where the images were taken in the map and load them.

Previously Coordinator was broken, Coordinator can now view the images and their location in google map and load the images from there.

Status: Reviewing

Responsibility: Bhargava & Rishabh

Time Taken: 5 Hours

Contributions: Bhargava(50%) & Rishabh(50%)

Story #17 As a developer I want to change the image holder view to select the regions in an image to analyze.

Selecting the regions and playing with holder to stretch or shrink to an particular region was difficult, We changed the holder view, which makes it easy to select regions easily for analyst.

Status: Done

Responsibility: Rishabh & Bhargava

Time Taken: 2 Hours

Contributions: Rishabh(50%) & Bhargava(50%)

Story #19 As a developer I want to test the performance of the application using 15 phones at same time.

System testing was done to do the thorough evaluation of the performance by installing the application on 15 phones and running at some time. No major flaws found except analyst list loading getting slower due to the server limitation.

Status: Done

Responsibility: Rishabh & Bhargava

Time Taken: 8 Hours

Contributions: Rishabh(50%) & Bhargava(50%)

Story #20 As a developer I want to write test cases for the application to do unit & integration testing. Unit testing by writing jUnit Test Cases for various test cases.

Status: Done

Status: Reviewing

Responsibility: Rishabh & Bhargava

Time Taken: 10 Hours

Contributions: Rishabh(50%) & Bhargava(50%)

Work to be completed.

Story #13 As a developer we want to do introduce the caching for the first time.

Caching helps from not loading the thumbnails all the time, for one session it loads only one and then it is cached, which also increases the performance of the application by n times. New images are loaded by load more data only. Good for multiple users running at same time.

Status: Doing

Responsibility: Rishabh & Bhargava

Time Taken: 4 Hours

Contributions: Rishabh(50%) & Bhargava(50%)

Story #18 As a developer I want to add health_status of an image to the analyst, and purpose a new machine learning solution to it.

Analyze the health of the green garden by checking the temporal & intensity of color of green garden in images automatically using supervised machine learning technique.

Status: To Do

Responsibility: Rishabh & Bhargava

Time Taken: 30 Hours

Contributions: Rishabh(50%) & Bhargava(50%)

Total Time Spent: 56 Hours

Issues/Concerns

Doing more to what could be possibly a public application, to find users and taking their input to make it better.

No background in Image processing. Have to read a lot, went through so many papers and codes to understand and to be able to use in application

Another difficulty is team size(2 people) and the amount of work that is required to carry the endeavor is a lot & timing issues.

Dealing with very flexible requirements Low server memory and high CPU usage.

References

- [1] <http://107.170.242.10:8080/serviceengine/>
- [2] <https://github.com/DIGiTLabHub>
- [3] <https://jersey.java.net/apidocs/1.18/jersey/index.html>
- [4] <https://developers.google.com/maps/documentation/android/>
- [5] <http://stackoverflow.com>
- [6] Guo-Hong, Shao, "Application Development Research Based on Android Platform," Intelligent Computation
- [8] <http://www.iscramlive.org/ISCRAM2012/proceedings/149.pdf>
- [9] <http://mashable.com/2013/07/29/fema-app-disaster-relief/>
- [10] <http://www.europeanbusinessreview.com/?p=4911>
- [11] <https://play.google.com/store/apps/details?id=com.arabiagis.disasterapp&hl=en>

[12] https://play.google.com/store/apps/details?id=kr-ishna.disaster_premiumversion&hl=en

[13] <http://www.vogella.com/tutorials/android.html>

[14]<http://developer.android.com/tools/projects>

[15]