# CS551 Advance Software Engineering Project Plan

Rishabh Bhojak (G2) Bhargava Gellaboina (G2)

Due: February 11, 2014

# **Disaster-Scene Crowdsourcing Solution**

## I. Introduction

This is a cloud based Mobile Application that helps researchers to analyze scenes from major disasters like tornados, earthquakes, Tsunami, etc. and inference from basic attributes like damage rate, type of objects damaged, etc., that helps as a good inspection for civil infrastructure conditions.

Two different versions named: DS-Crowd and CISA
DS-Crowd – Analyst and Coordinator
CISA – Observer, Analyst and Coordinator

The intelligence of the 'crowd', who can be first-responders, professional inspectors, and the general public within or outside the disaster zone, to collect, analyze and manage disaster-scene data. Disaster scenes form extreme disasters, such as hurricanes, storms, surges, earthquakes, tsunamis, and tornadoes, are extremely complex. In the meantime, disaster scenes are perishable due to organized recovery efforts. Therefore, rapid post-disaster reconnaissance efforts are imperative in order to learn from disasters important knowledge, such as hazard effects, damage mechanisms, and vulnerabilities of built environments.

## II. Project Goal and Objectives

- **Overall Goal**

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.

- **Specific Objectives**

System Features:

- 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.
- Rest: Based on geolocation services get the meaningful data and visualize it on the same
- Machine Learning: To classify the images using Naive Bayes approach. This approach is tentative and subjected to change.

- **Significance**

There is no automated (i.e. computer-based machine vision or learning-based) solution to date that can analyze such complex damage scene images automatically. Visual (human-based) inspection is the general approach to characterize the damage information embedded in the natural scene. For any disaster scene image, we propose the machine learning aspect that can help to characterize the images of potential damage or effects for an object in an image: type of, location of, and degree of damage/effect to the object.

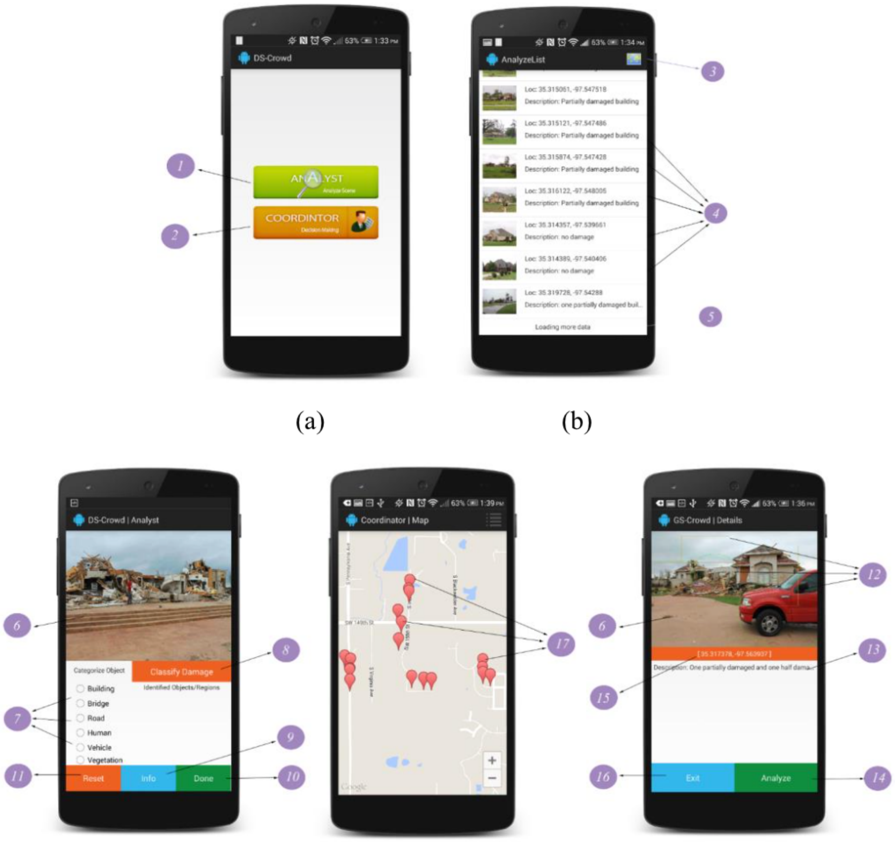
## III. Project Background and Related Work

Existing work

http://107.170.242.10:8080/serviceengine/

Android Application(DS-Crowd)

https://github.com/DIGiTLabHub/GS-Crowd



Loc: 35.315061, -97.547518
Description: Partially damaged building

Loc: 35.315121, -97.547486
Description: Partially damaged building

Loc: 35.315874, -97.547428
Description: Partially damaged building

Loc: 35.316122, -97.548005
Description: Partially damaged building

Loc: 35.314357, -97.539661
Description: no damage

Loc: 35.314389, -97.540406
Description: no damage

Loc: 35.319728, -97.54288
Description: one partially damaged buil...

Loading more data

Description: One partially damaged and one half dama...

(a) (b)

(c) (d) (e)

Figure 2. User interface (UI) design for the DS-Crowd application: (a) main menu; (b) list of available images; (c) crowd-analyst interface; (d) GIS marking of analyzed images; and (e) damage details from crowdsourcing with interfacing to further analysis.

**Table 1. List of UI elements.**

## Comparison

There is no such existing application that aims to create structured information for disaster-induced damage. Given the images, one can see that complex natural scenes besides distinct damage patterns are found in the images. Crowdsourcing approach can potentially provide the most rapid and reliable solution. This is a research project and it is not related to any existing solutions and thus, cannot be compared.

## IV. Proposed System

### 1) Requirement Specification

- **Functional**: Users needs an android phone and should be able to take and view the images using the application. Administrator should be able to view and analyze the images, add events in web application. As input properties, the android device should have an ID, images should have geotag location, image type, region, category, damage level, boundary. Output would be a map with geo location and information about the image(only for the android application).

**Non-functional**:

Requirements of System:

OS:  OS independent

RAM: 2GB or more

CentOS Server 5.11

Apache Tomcat Server 7

Java Version 1.7

MongoDB

Android with 4.1 or higher

API (19)

**Technical/Business Requirements**

Eclipse ADT

JavaScript, AJAX, JQuery, Jssor Slider, Bootstrap

MongoDB

CentOS

Java, Rest Api using Http Get, Http Post

HTML5

- **Business Process/Workflow analysis**

Administrator logs into the server, http://someserver.com:8080/serviceengine/

Administrator can add/remove events.

Users can view all the images and useful information related to these images

Users and Administrator can use the client version of the app

Administrator can use the admin version of the app to analyze the images.

## Activity Diagram

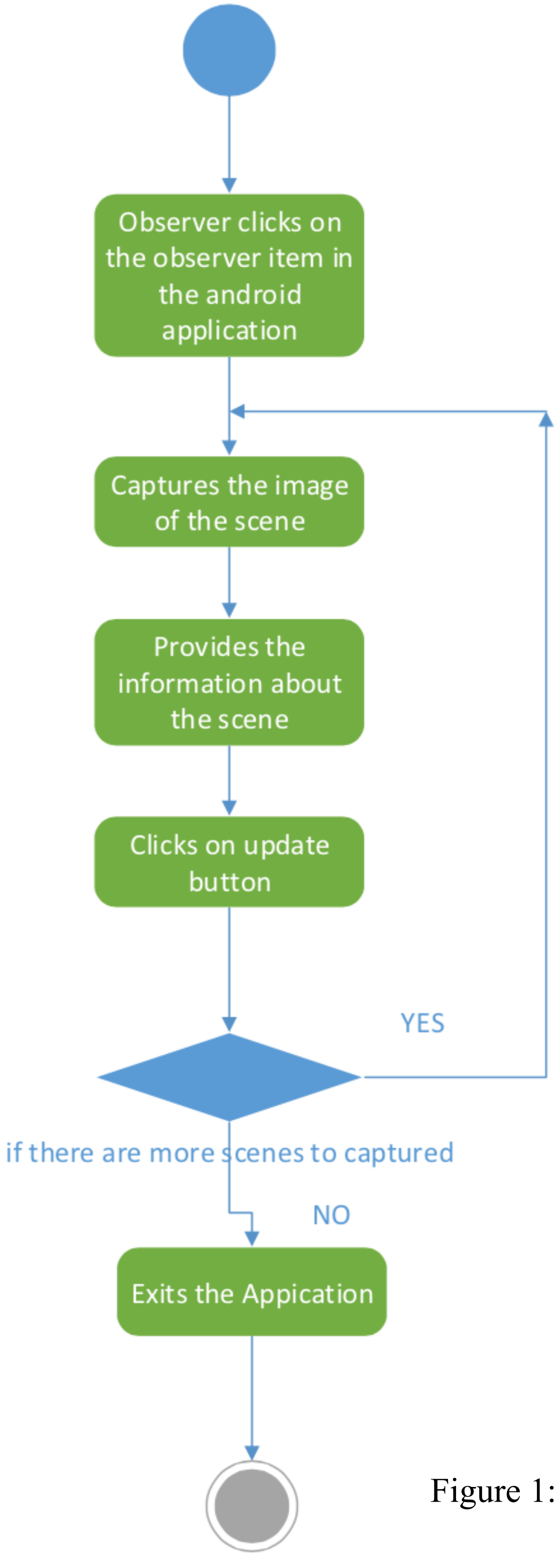

Observer clicks on the observer item in the android application
Captures the image of the scene
Provides the information about the scene
Clicks on update button
YES
if there are more scenes to captured
NO
Exits the Appication

Figure 1: Observer

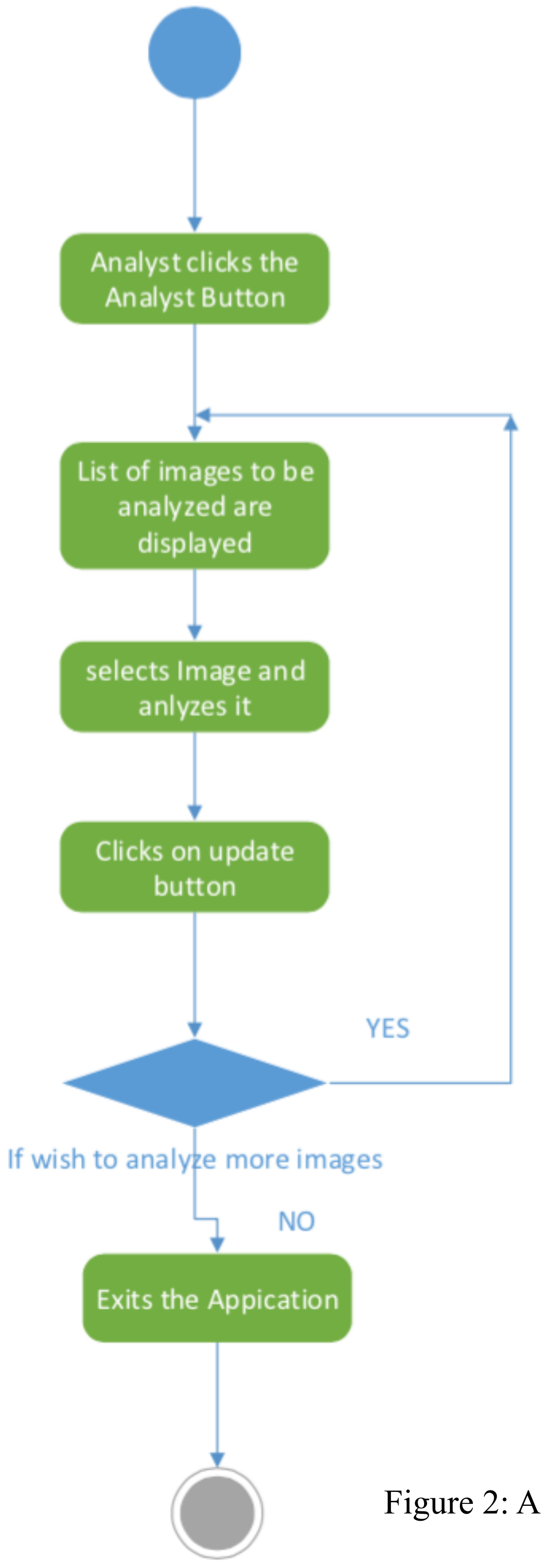

Analyst clicks the Analyst Button
List of images to be analyzed are displayed
selects Image and anlyzes it
Clicks on update button
YES
If wish to analyze more images
NO
Exits the Appication

Figure 2: Analyst

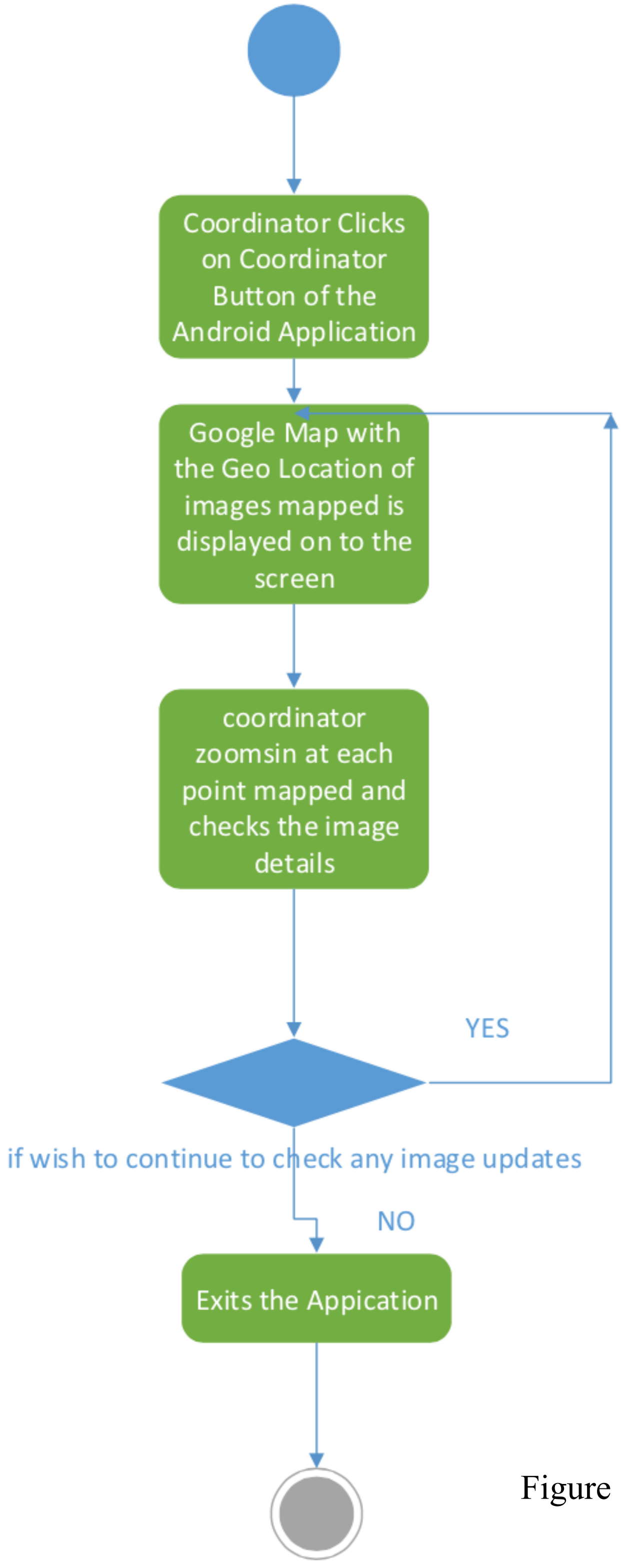

Coordinator Clicks on Coordinator Button of the Android Application
Google Map with the Geo Location of images mapped is displayed on to the screen
coordinator zoomsin at each point mapped and checks the image details
YES
if wish to continue to check any image updates
NO
Exits the Appication

Figure 3: Coordinator

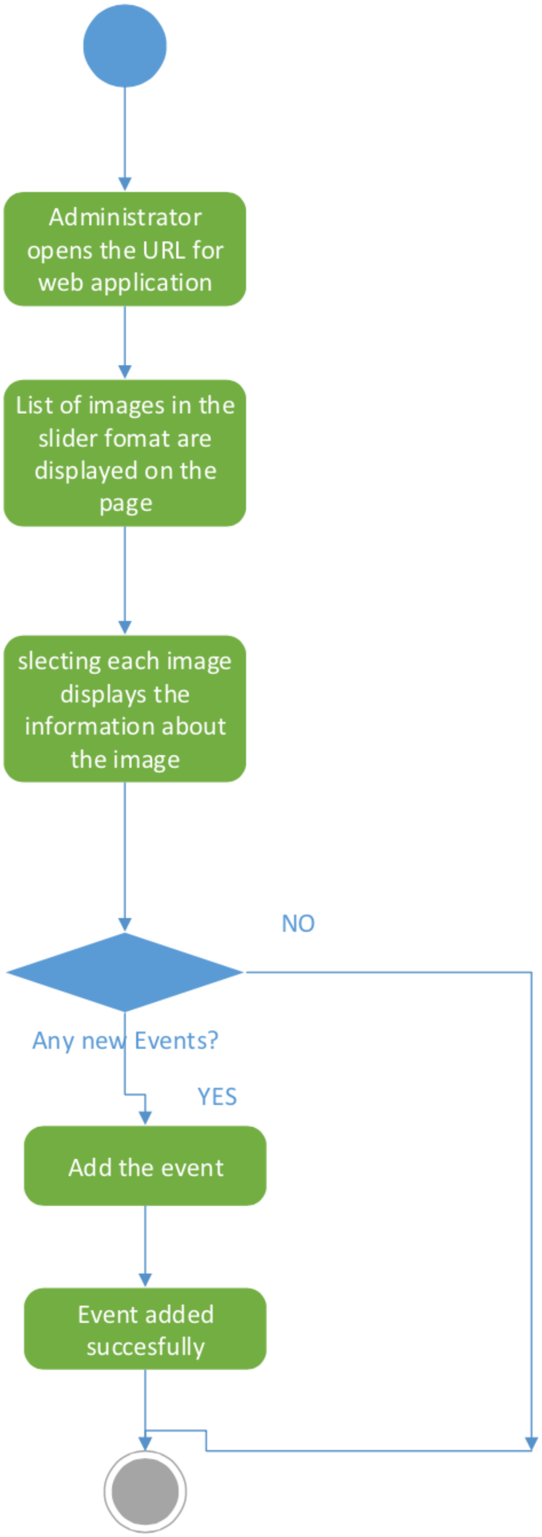

Administrator opens the URL for web application
List of images in the slider fomat are displayed on the page
slecting each image displays the information about the image
NO
Any new Events?
YES
Add the event
Event added succesfully

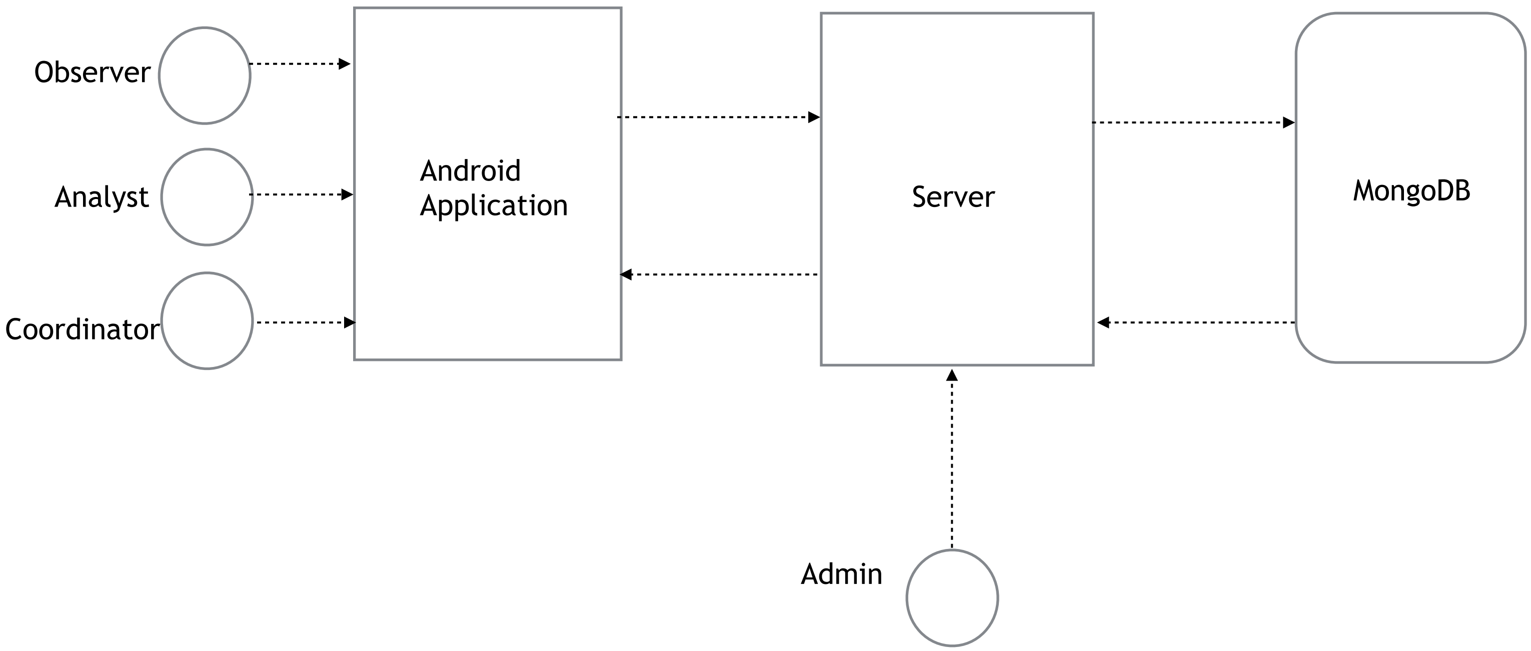
Figure 4: Administrator

## 2) Framework Specification:

- **Assumptions and Principles**

  People near the disaster location are using this application and uses it to capture disaster scenes and uploads the images into the server.

- **System Architecture Diagram**



## 3) System Specification

- **Existing Services**

  https://jersey.java.net/apidocs/1.18/jersey/index.html

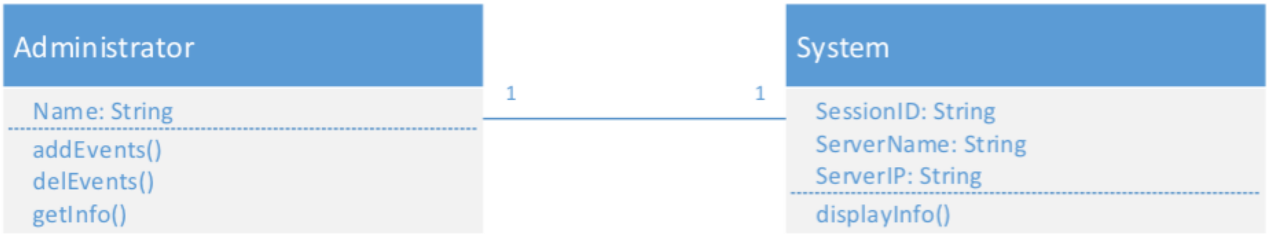
  https://developers.google.com/maps/documentation/android/

- **New Services to be build**

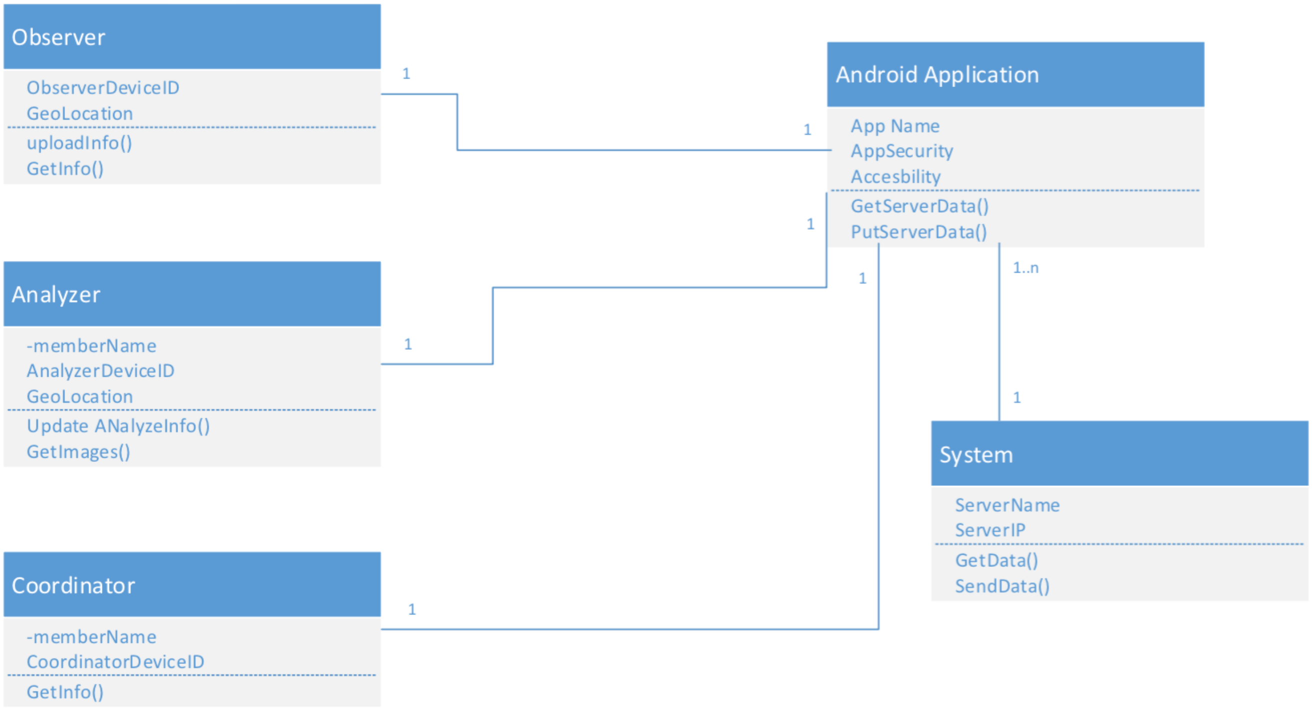
Restful Service to get image data and text data from server

Show the geo location on the map

- **Class diagram**

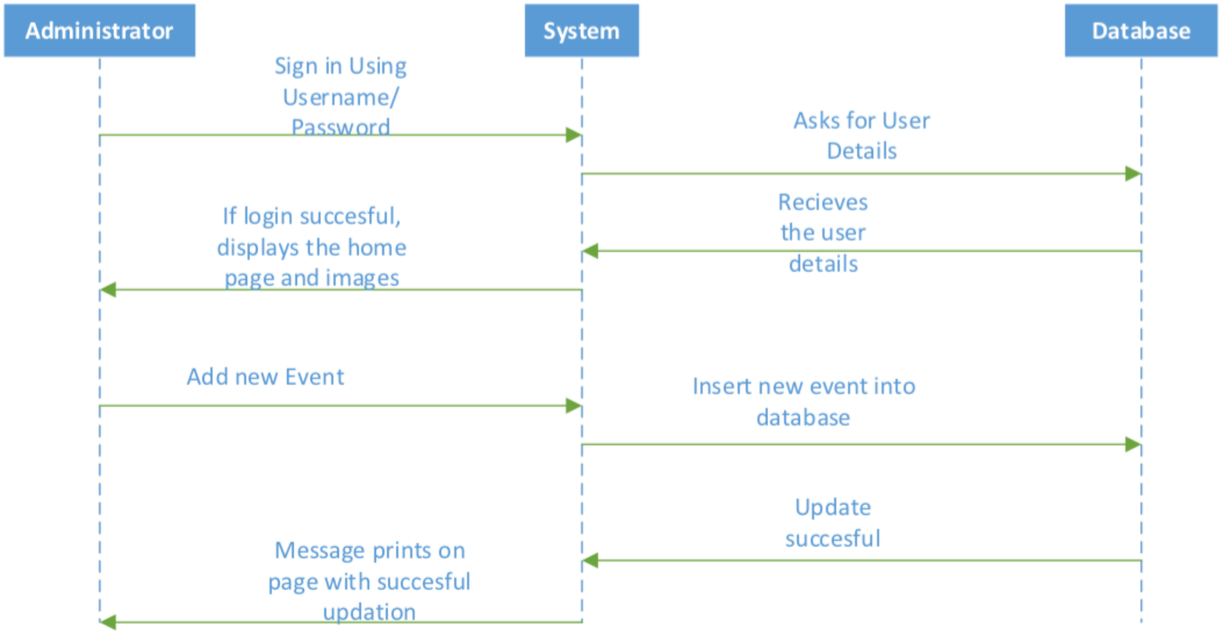


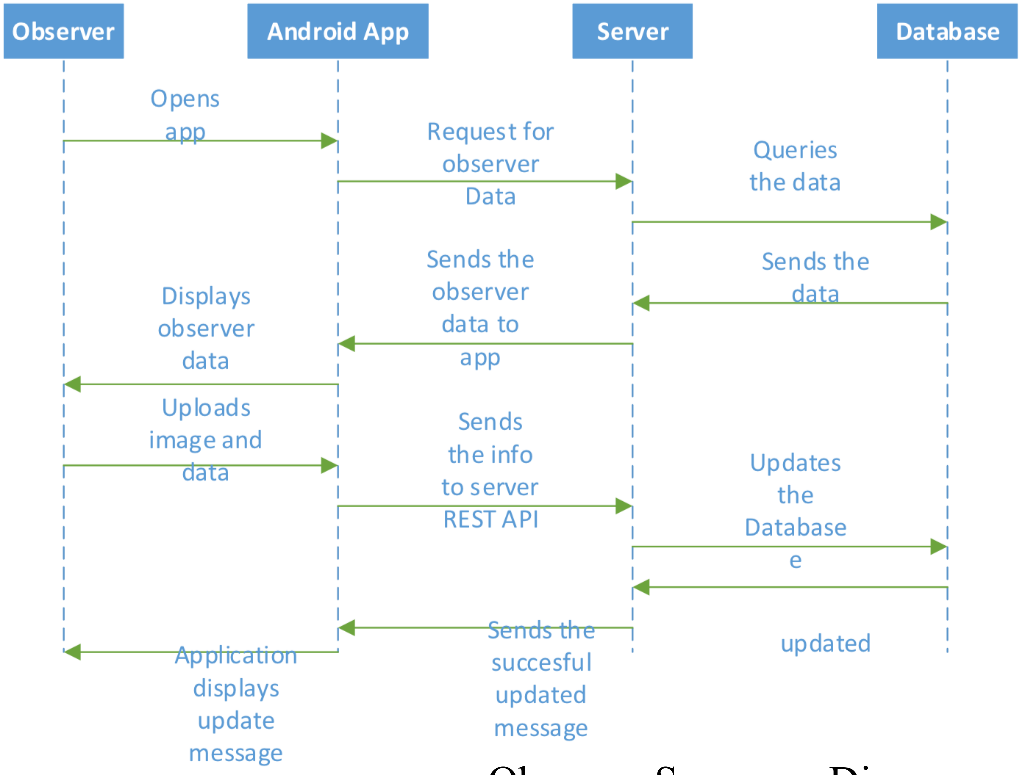
Class diagram of image, text and geotag retrieval & application.



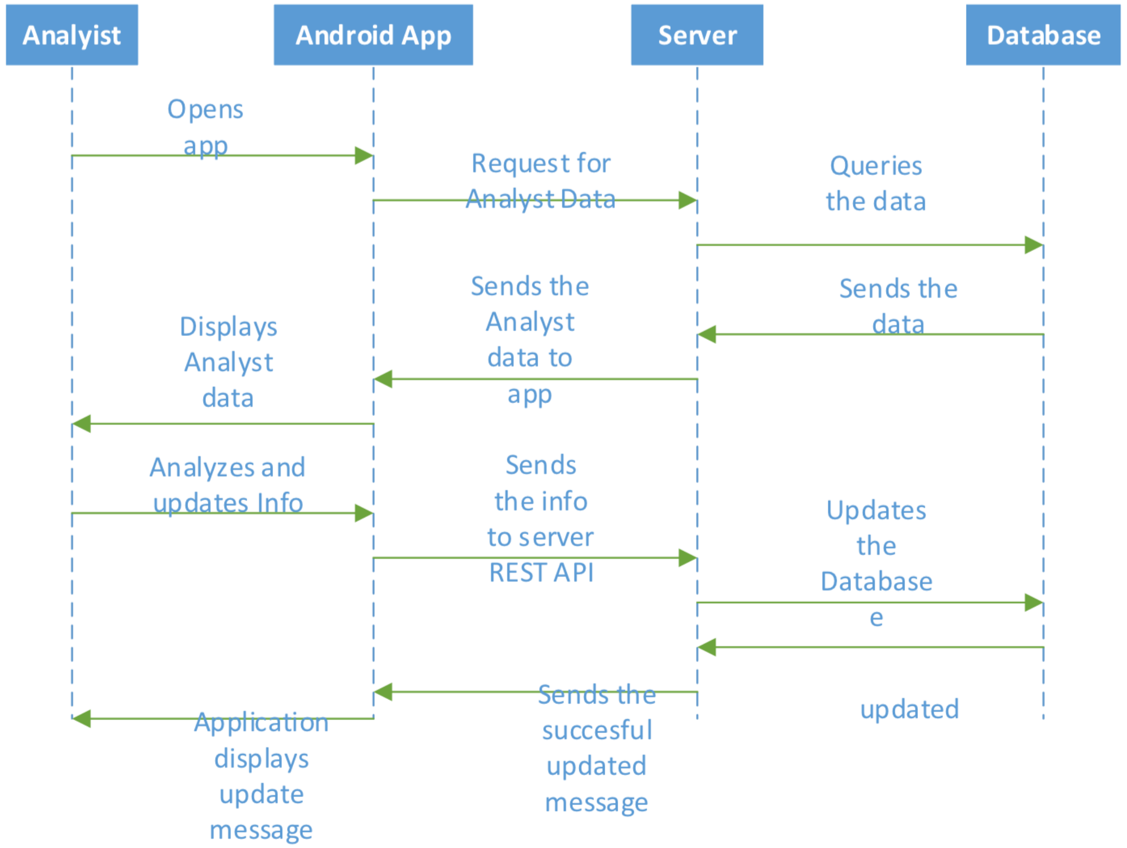
- **Sequence diagram**

Administrator Sequence Diagram

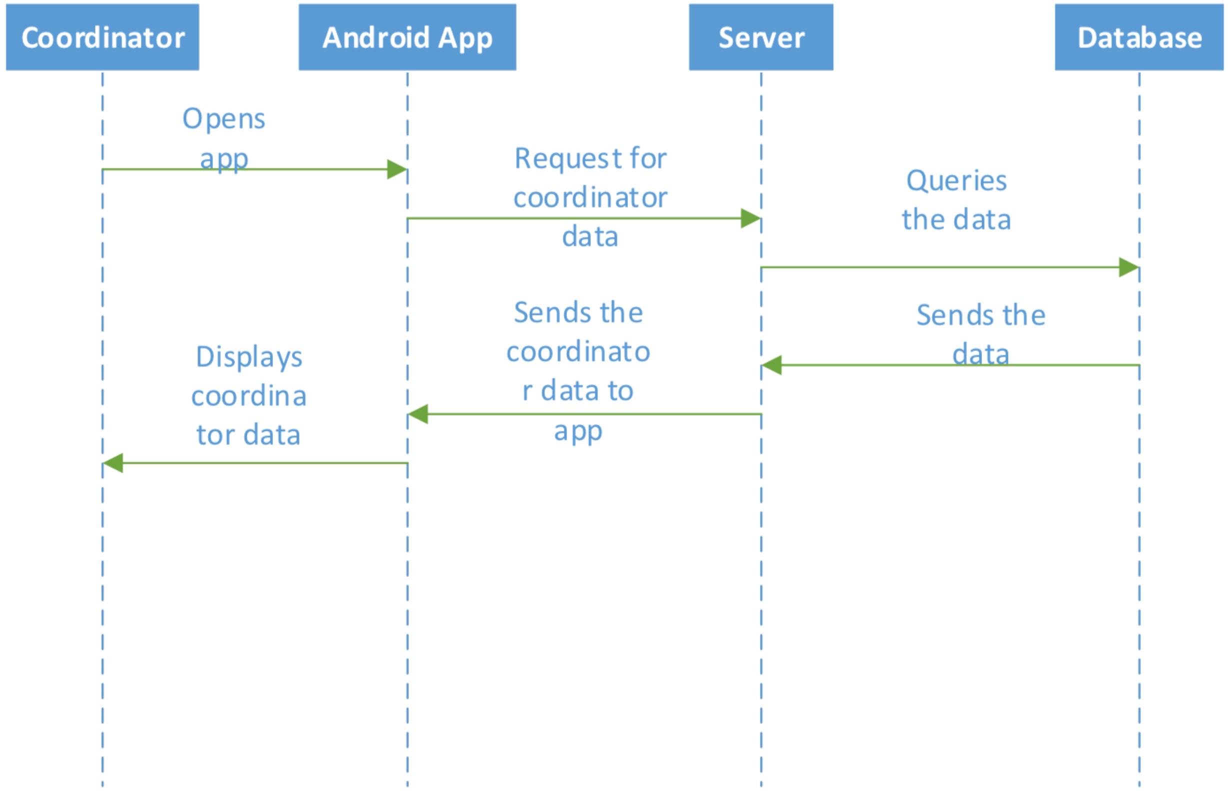

Administrator
System
Database
Sign in Using Username/ Password
Asks for User Details
Recieves the user details
If login succesful, displays the home page and images
Add new Event
Insert new event into database
Update succesful
Message prints on page with succesful updation


Observer
Android App
Server
Database
Opens app
Request for observer Data
Queries the data
Sends the data
Sends the observer data to app
Displays observer data
Uploads image and data
Sends the info to server REST API
Updates the Database e
updated
Sends the succesful updated message
Application displays update message

Observer Sequence Diagram


Analyist
Android App
Server
Database
Opens app
Request for Analyst Data
Queries the data
Sends the data
Sends the Analyst data to app
Displays Analyst data
Analyzes and updates Info
Sends the info to server REST API
Updates the Database
e
updated
Sends the succesful updated message
Application displays update message

Analyst Sequence Diagram


Coordinator
Android App
Server
Database
Opens app
Request for coordinator data
Queries the data
Sends the data
Sends the coordinator data to app
Displays coordinator data

Coordinator Sequence Diagram

- **Service Specification**
  - **Operational description**

It will make the Http request to get the ID of the images.

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

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

Using the Rest services we will get the geo location of the images and will visualize it in google maps.

Using geotag location on maps, other features will be added but we are still thinking about them.

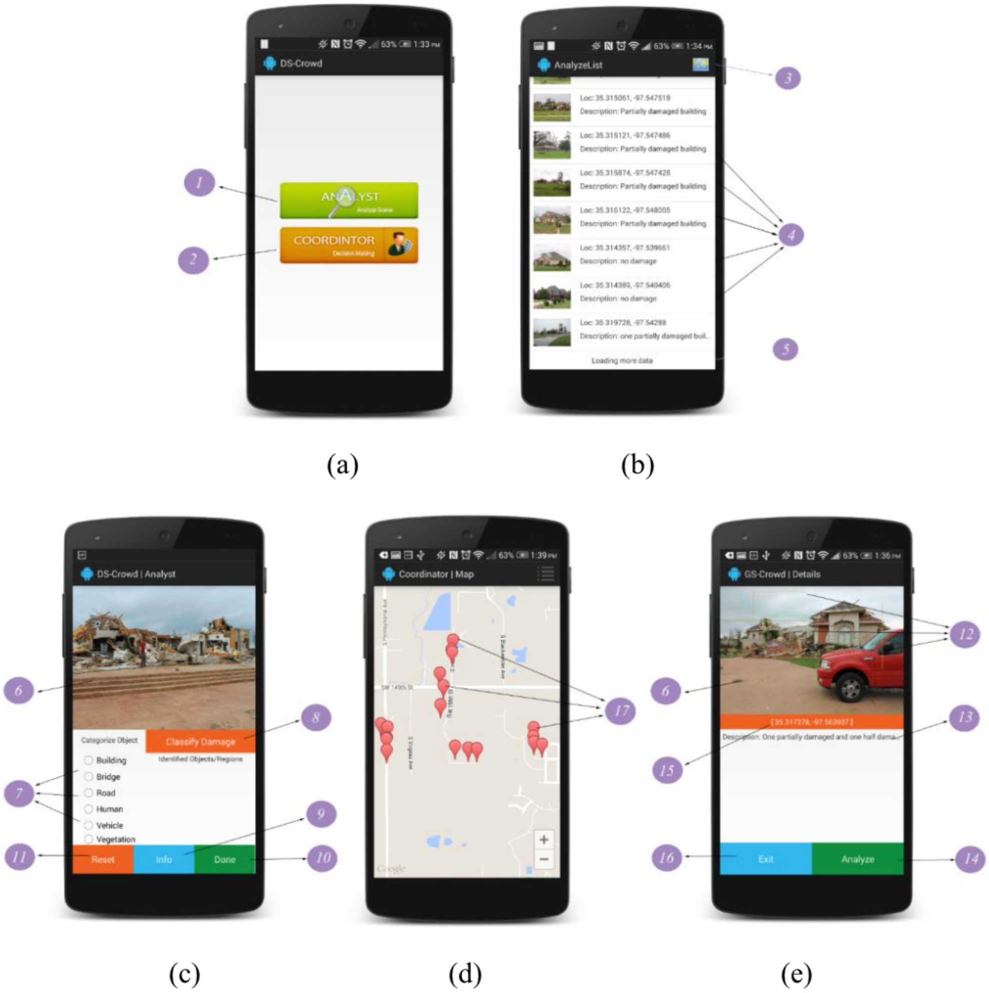
- **Input/output for services**

Showing the geotag location of the pictures in google map. Using the location for other features which users would like.

The display consists of data which is based on priorities, accessibility and other factors.

- **Constraints/exceptions**
  - Server memory is low and CPU Usage is high
  - Server crashing, flexible requirements
  - Port 8080 not accessible on internet

## - Service flow/alternative flow



Loc: 35.315061, -97.547518
Description: Partially damaged building

Loc: 35.315121, -97.547486
Description: Partially damaged building

Loc: 35.315874, -97.547428
Description: Partially damaged building

Loc: 35.316122, -97.548005
Description: Partially damaged building

Loc: 35.314357, -97.539661
Description: no damage

Loc: 35.314389, -97.540406
Description: no damage

Loc: 35.319728, -97.54288
Description: one partially damaged buil...

Loading more data

[ 35.317378, -97.563937 ]
Description: One partially damaged and one half dama...

(a) (b)

(c) (d) (e)

Figure 2. User interface (UI) design for the DS-Crowd application: (a) main menu; (b) list of available images; (c) crowd-analyst interface; (d) GIS marking of analyzed images; and (e) damage details from crowdsourcing with interfacing to further analysis.

**Table 1. List of UI elements.**

No alternative flow(s)

### - Priorities

Make the user side application ready

Code enhancement to use the server memory effiencetly

Adding analyst and coordinator in the app.

Adding functionalities of all users

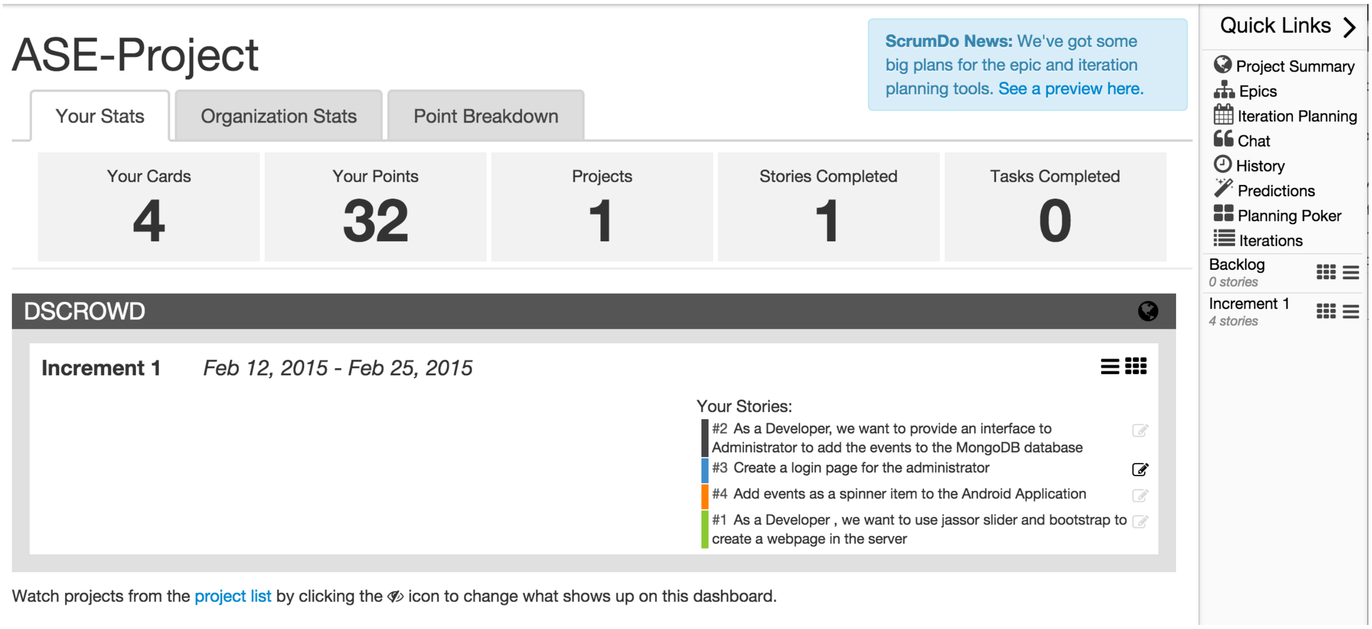
Administrator login in web

Admin can add/remote events for analyzers.

Integrating everything as a whole application

## V. Plan by Services

Scrum



ASE-Project

**ScrumDo News:** We've got some big plans for the epic and iteration planning tools. See a preview here.

| Your Cards | Your Points | Projects | Stories Completed | Tasks Completed |
|---|---|---|---|---|
| 4 | 32 | 1 | 1 | 0 |

DSCROWD

**Increment 1** *Feb 12, 2015 - Feb 25, 2015*

Your Stories:

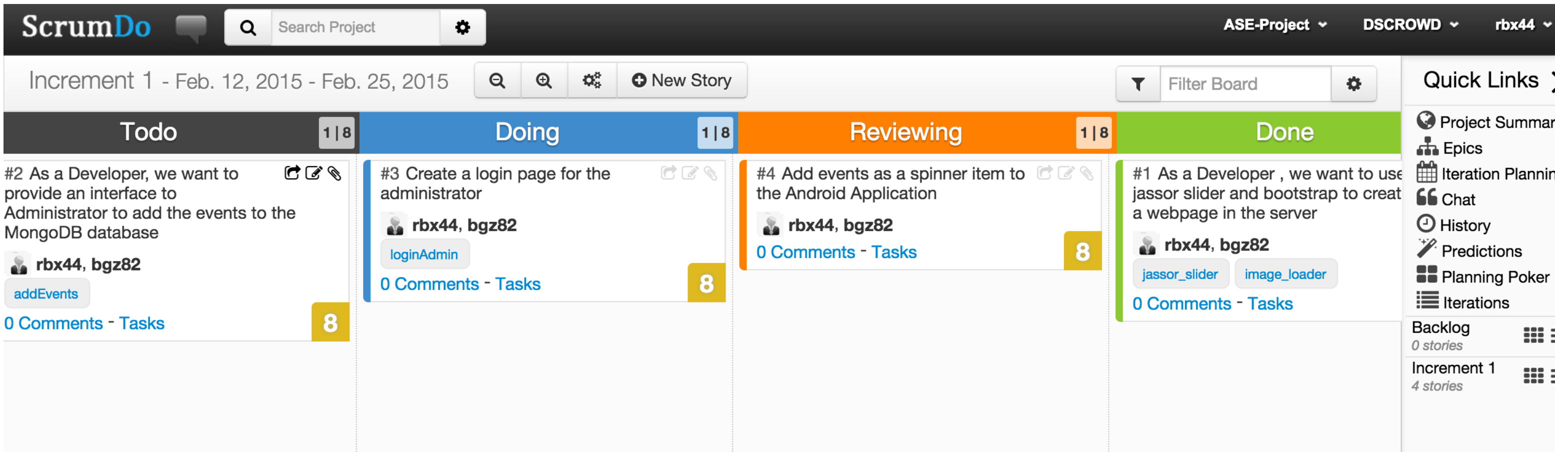
#2 As a Developer, we want to provide an interface to Administrator to add the events to the MongoDB database

#3 Create a login page for the administrator

#4 Add events as a spinner item to the Android Application

#1 As a Developer , we want to use jassor slider and bootstrap to create a webpage in the server

Watch projects from the project list by clicking the icon to change what shows up on this dashboard.

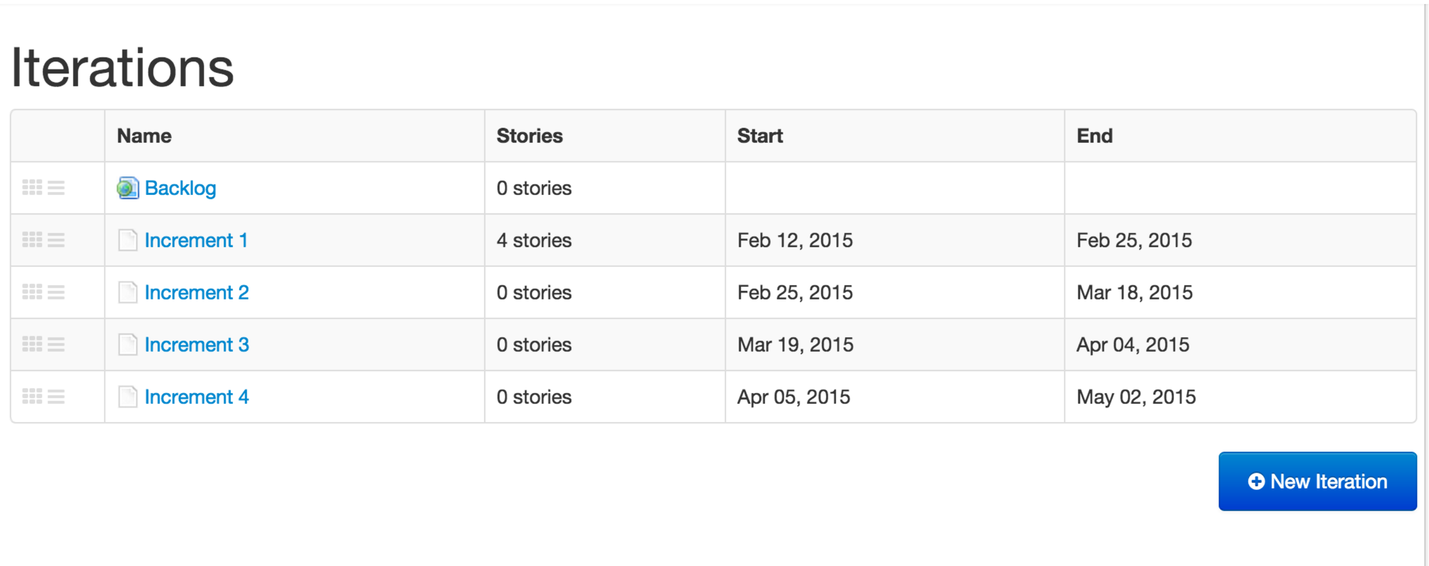


Increment 1 - Feb. 12, 2015 - Feb. 25, 2015

| Todo 1 \| 8 | Doing 1 \| 8 | Reviewing 1 \| 8 | Done |
|---|---|---|---|
| #2 As a Developer, we want to provide an interface to Administrator to add the events to the MongoDB database<br>rbx44, bgz82<br>addEvents<br>0 Comments - Tasks<br>8 | #3 Create a login page for the administrator<br>rbx44, bgz82<br>loginAdmin<br>0 Comments - Tasks<br>8 | #4 Add events as a spinner item to the Android Application<br>rbx44, bgz82<br>0 Comments - Tasks<br>8 | #1 As a Developer , we want to use jassor slider and bootstrap to creat a webpage in the server<br>rbx44, bgz82<br>jassor_slider image_loader<br>0 Comments - Tasks |

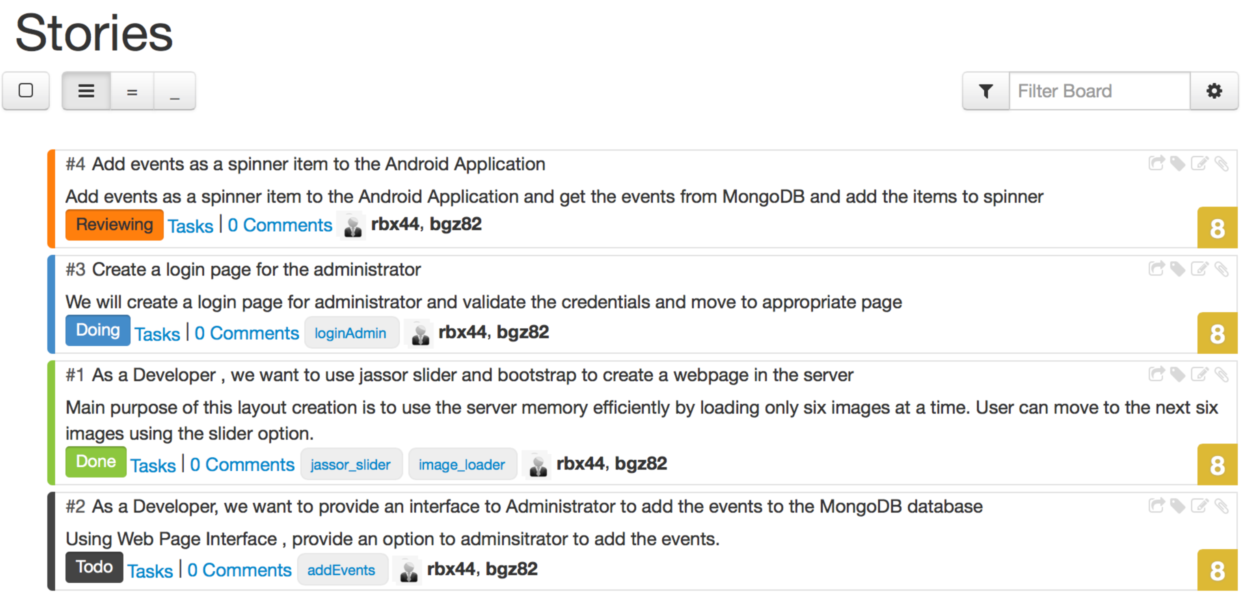
Backlog
*0 stories*

Increment 1
*4 stories*



# Iterations

| | Name | Stories | Start | End |
|---|---|---|---|---|
| | Backlog | 0 stories | | |
| | Increment 1 | 4 stories | Feb 12, 2015 | Feb 25, 2015 |
| | Increment 2 | 0 stories | Feb 25, 2015 | Mar 18, 2015 |
| | Increment 3 | 0 stories | Mar 19, 2015 | Apr 04, 2015 |
| | Increment 4 | 0 stories | Apr 05, 2015 | May 02, 2015 |



# Stories

#4 Add events as a spinner item to the Android Application

Add events as a spinner item to the Android Application and get the events from MongoDB and add the items to spinner

Reviewing Tasks | 0 Comments rbx44, bgz82 8

#3 Create a login page for the administrator

We will create a login page for administrator and validate the credentials and move to appropriate page

Doing Tasks | 0 Comments loginAdmin rbx44, bgz82 8

#1 As a Developer , we want to use jassor slider and bootstrap to create a webpage in the server

Main purpose of this layout creation is to 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.

Done Tasks | 0 Comments jassor_slider image_loader rbx44, bgz82 8

#2 As a Developer, we want to provide an interface to Administrator to add the events to the MongoDB database

Using Web Page Interface , provide an option to adminsitrator to add the events.

Todo Tasks | 0 Comments addEvents rbx44, bgz82 8 https://www.scrumdo.com/projects/project/dscrowd/iteration/121689/board

https://www.scrumdo.com/projects/project/dscrowd/planning

VI. Risk management

- **Technological and Architectural Requirements**

  Not familiar with fronted technologies like Ajax, Jquery

  None to very less knowledge about disaster management

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

  Dealing with very flexible requirements

  Low server memory and high CPU usage.

VII. Posting the class google site:

https://docs.google.com/spreadsheets/d/1QtbhKeCep4SvzP5gCYs1iGPyKQ-xhooCoG-qLR5uY18/edit#gid=2115159985

## VIII. Bibliography

[1] http://107.170.242.10:8080/serviceengine/

[2] Implemented Android Application(DS-Crowd)

[3] https://github.com/DIGiTLabHub/GS-Crowd

[4] https://jersey.java.net/apidocs/1.18/jersey/index.html

[5] https://developers.google.com/maps/documentation/android/

[6] http://stackoverflow.com

[7] 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=krishna.disaster_premiumversion&hl=en