

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

3/18/2015

My Shared Diary

Android Application Project

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Project Group: PG2

Team Members

Chelle, Vishnu – 07

Arumalla, Chandra Mouli – 03

Lam, Sundar Sagar – 19

Salapaka, Phanideep – 28

I. Introduction

The idea of this project comes from the question which each one of us encounter at some point of our daily life “What have I done during last week/month/someday?”, and we don’t have an answer. We believe- “Human brains are very powerful to remember almost everything that happens but humans are not that powerful to restore that information from brain”. Here comes the necessity of having some other system to track and store our daily activities. This was done traditionally from ages by writing a “Dairy” mostly at the end of day about the events of that day. But now with the busy modern work life, it is difficult to find time to sit and write about ourselves. But still we have to track ourselves right?

This gives us the motivation to create something which will be with us throughout most of our daily time and track out events. Now the question comes what we have to create? Based on our understanding on this modern world we are making this statement “Smartphone is almost a human body part.” This motivated us to consider smartphone to track our daily activities.

II. Project Goal and Objectives

Overall goal:

Our main goal is to build an android application [9] which will replace the traditional system of writing dairies in books. We know that this is not a new idea but our objective is to make a part of our personal dairies to get shared between communities. A user can share his/her info as private information which will remain on his time track also share some common events or info with his communities like friends and family. Objective is to track ones individual life based on timestamp and location as well as track daily events of the person based on communities like family.

Specific Objective:

To be specific our objective is to answer two simple questions of an individual:

“What I have done in last month/week/specific date?” – Provide tracked input of user including metrics like travel, restaurants etc.

“What my community (e.g.: family) has done in last month/week/specific date?” – Provides all the shared information from the users belonging to a specific group.

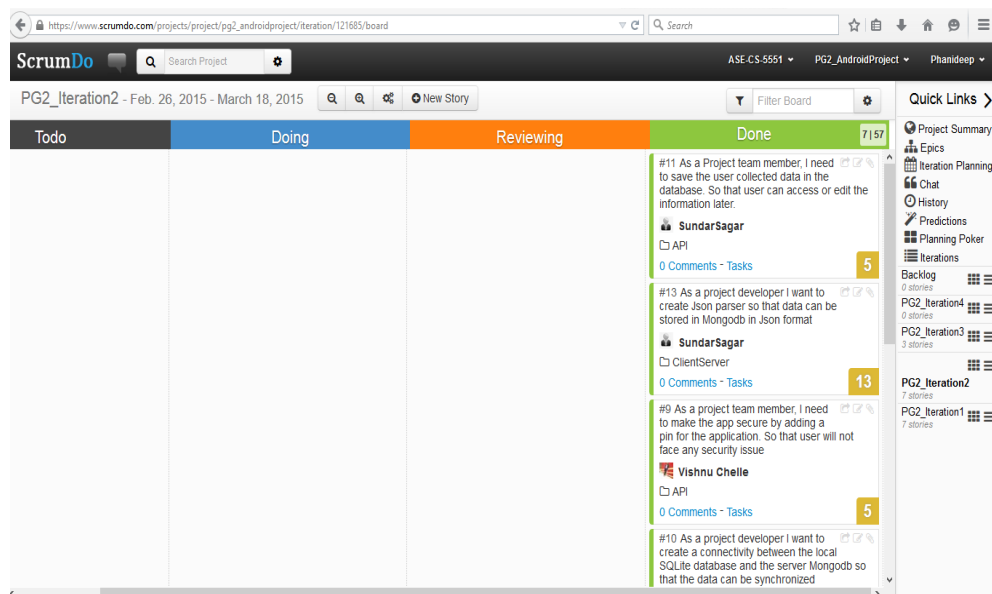
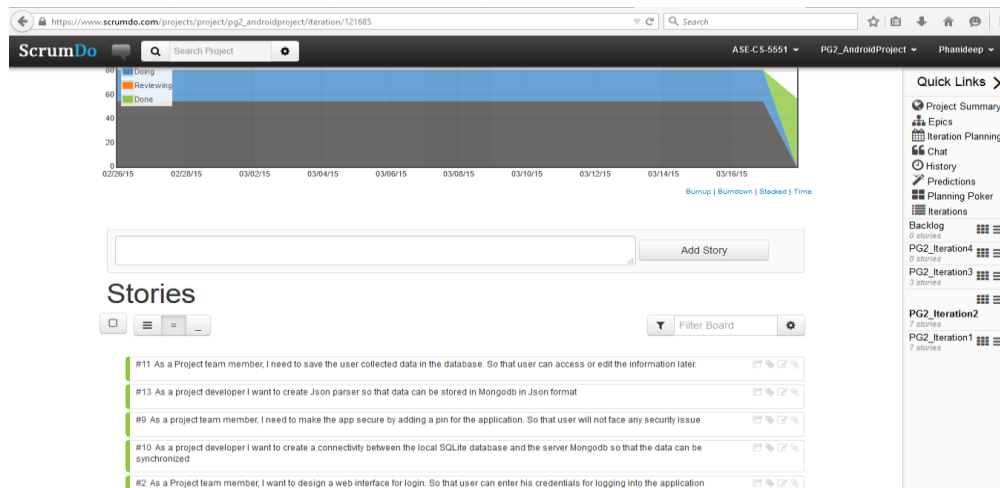
III. Import Existing Services/API:

Yelp API: Yelp API is a powerful service to search restaurants using geo location. Our idea of using this service in our project is to provide the user recommendations of restaurants during use's lunch and dinner time.

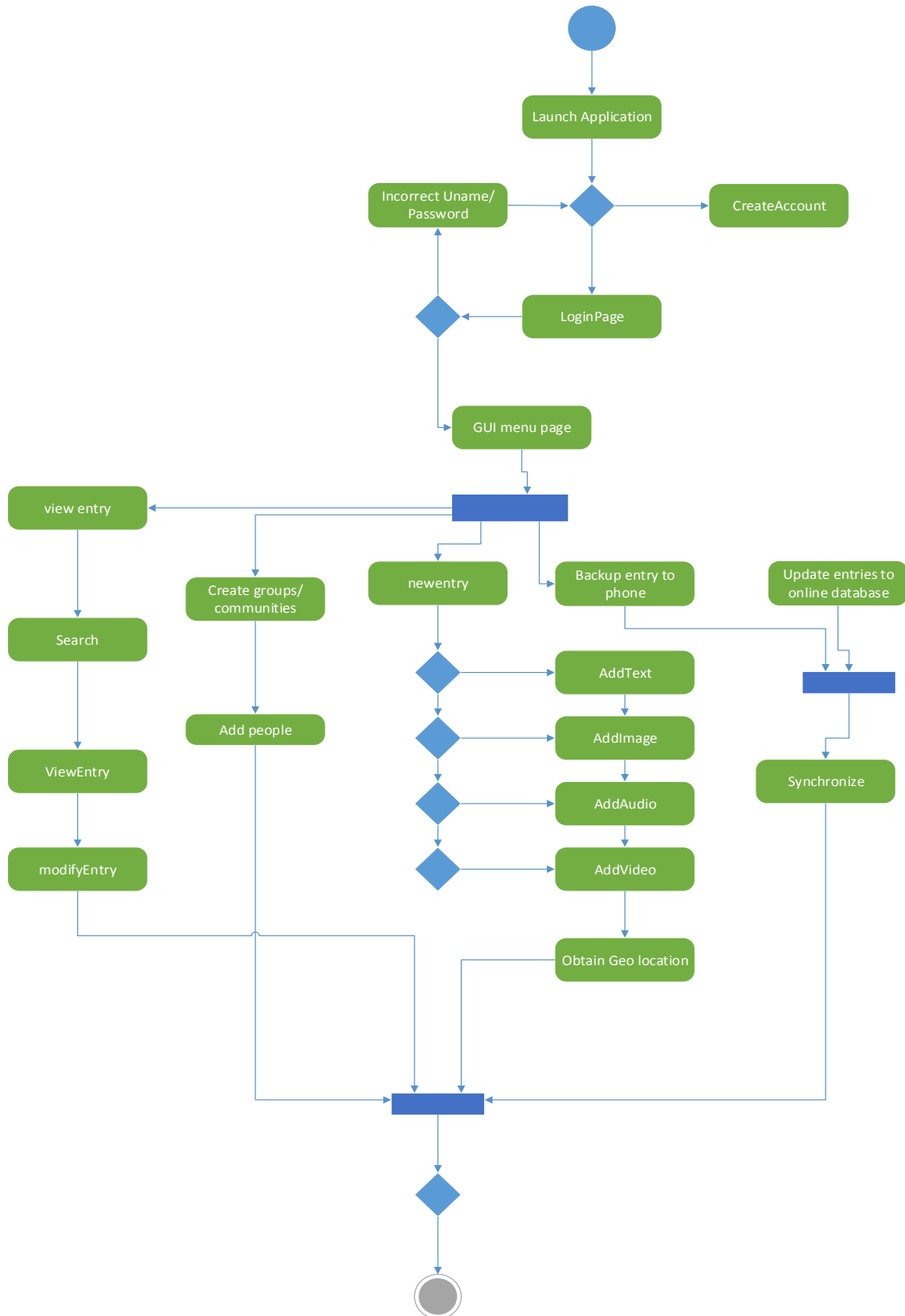
Google Maps API: We are getting the current location of the user using this service. The location details will be used to calculate user's travel metrics and also to store the user's input with respect to location.

IV. Detailed Design and Services

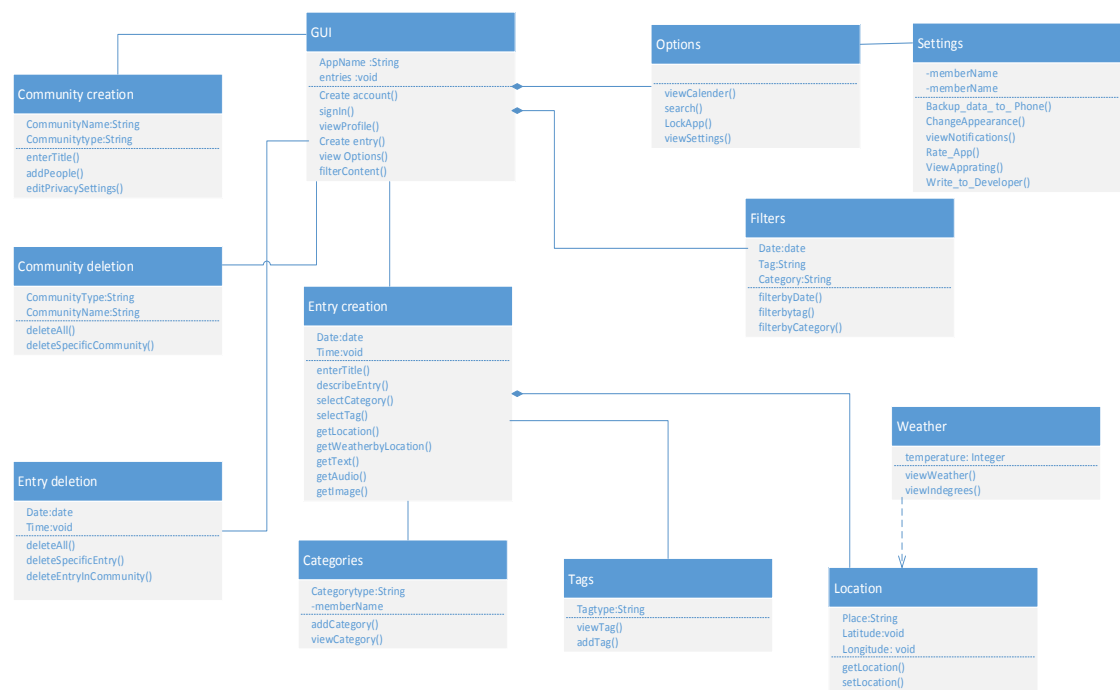
User Storied from ScrumDo:



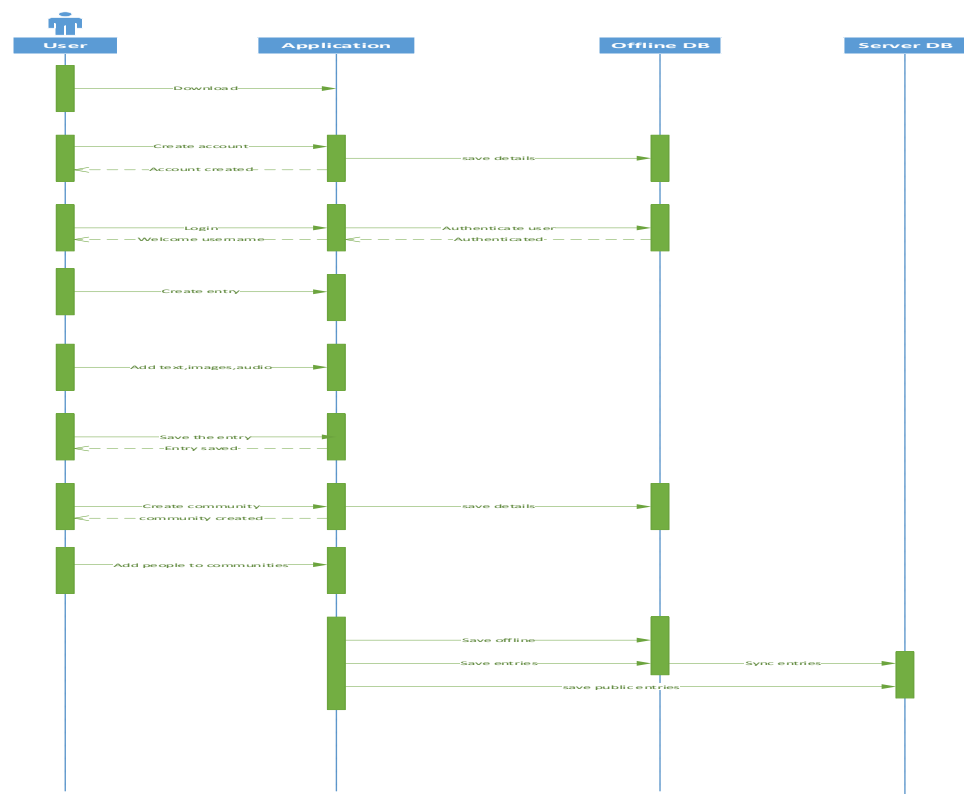
Activity Diagram:



Class Diagram:



Sequence Diagram:



Design of Mobile Client Interface:

Mobile client interface contains login and signup options.

User also has the advantage of logging into the application using google+ account.

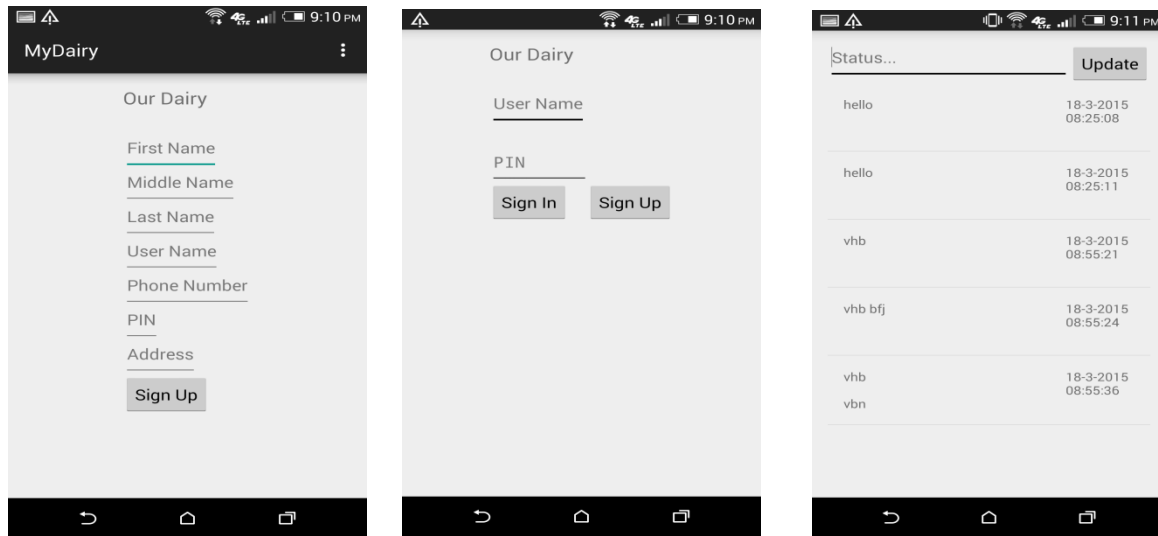
Once the user logs into the application, an entry screen of the dairy will be displayed.

The entry screen of the dairy contains adding an entry to the diary and deletion of an existing entry.

User can create a security pin lock the app.

Below are the screenshots for the above scenarios:



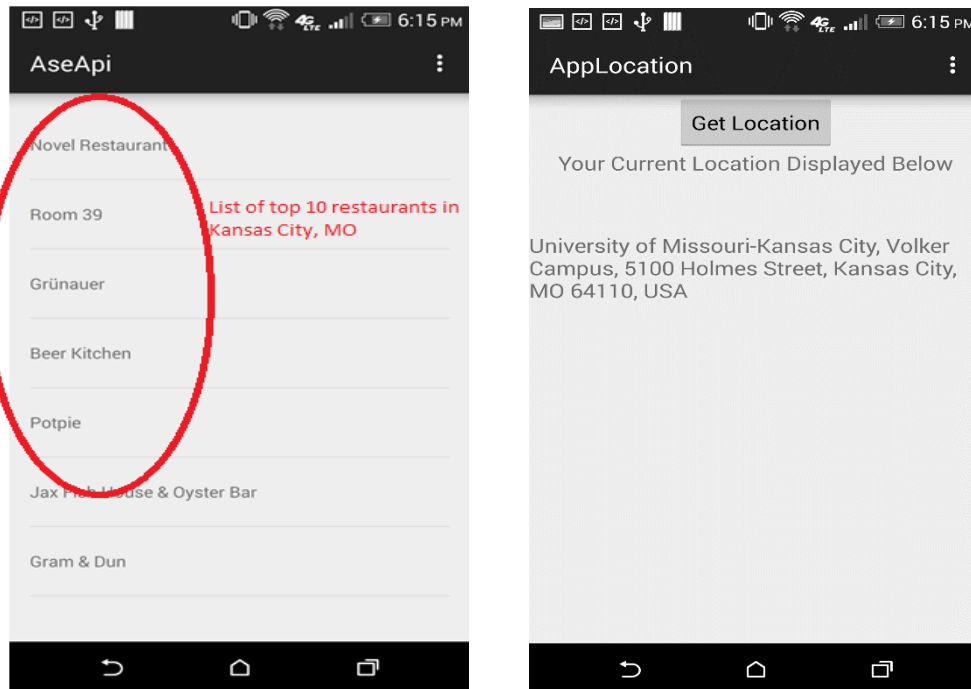


V. Implementation of REST Services

For the first iteration we have divided our work into two parts mainly as mentioned below.

1. Implement a simple dairy template with basic inputs like text and store them in the database using timestamp.
2. Play around with existing API's which we will be using in coming iterations. We have developed two sample application for both the API's which we are using to test whether the API's are working as expected.

Yelp API: AseApi application which provides Top 10 restaurant in any search area. This application is developed using Yelp API service. We are making a HTTP POST request to Yelp API and retrieving information. We will get a list of businesses in the searched area in JSON format. For the time being we are only displaying only name of the business in the below application.



Google Maps API: Below application AppLocation will fetch the current location of the user at any point of time when the user's GPS is on. The application uses Android GPS service to fetch the latitude and longitude of the user and then we will make a HTTP POST request to Google API with the latitude and longitude to get the current address of the user. Google Maps API provides results in a JSON array format with all possible locations. In below application we are displaying the formatted_address of the first location in the array.

MONGO database as backend server:

We have chosen mongo db as our server database for its flexible nature. The data model of mongo db helps you to store data of any structure and dynamically modify the schema. Mongolab.com provides sample database to load some data. A collection is created in this database to load the data from SQLite database to mongo. The lab also provides a unique API key for our account which is used as an authentication to load our data. The code included in the querybuilder will convert the data in the SQLite table to JSON format through parsing. The retrieval in mongo lab also shows the values in JSON format as well as it loads the values into it in JSON.

Generation of Unique API Key

API Key

The resources in your MongoLab account can be accessed via the MongoLab REST API.


In order to use the API, you must code your clients to present an API Key to the server on each API request. The key should be presented using an HTTP query parameter called 'apiKey' as in the following example:

```
https://api.mongolab.com/api/1/databases?apiKey=<your-api-key>
```

Each user of your MongoLab account has a separate API Key. The current key for this user is displayed below. You may regenerate the key for this user at any time.

Current key: `ueLZAN3-odIb4OEB7uKPdItxBuaG_Pz9`

Database Creation:



Welcome Plans & Features Pricing Docs & Support Account [Log out](#)

{ user: "phanideep.salapaka", account: "Phanideep" }

[Home](#)

MongoDB Deployments

Create from backup Clone existing Create new

Development and Utility Single-node deployments intended for environments that do not require high-availability.

NAME	PLAN	RAM	SIZE	FILE SIZE
<div>code101</div> <div>Ok: Database is up and running.</div>	Sandbox	shared	11.58 KB	16.00 MB

CLOUD: AWS US-EAST-1
DATABASES: 1 DOCUMENTS: 18

Collection created inside the database:

Collections Users Stats Backups Tools

Collections

+ Add collection

NAME	DOCUMENTS	CAPPED?	SIZE
docs101	14	false	11.27 KB

System Collections

NAME	DOCUMENTS	SIZE
system.indexes	1	0.11 KB

Key values stored in JSON format

Home: (code101) (user: 'phanideep.salapaka', account: 'Phanideep')

Collection: docs101

Documents | Indexes | Stats | Tools

Documents

Start new search

All Documents

Display mode: list | table (edit table view)

records / page 10 [1 - 10 of 14] next > last >>

```
{
  "_id": {
    "$oid": "5507a758e4b0603bec0c8ab1"
  },
  "document": {
    "userName": "phanideep",
    "Timestamp": "1426733796436",
    "text": "Today is Sunday, We had dinner at restaurant"
  },
  "safe": true
}
```

Documents (aka Objects)

From the "Documents" tab you can browse and search for objects in this collection. All standard query constructs are supported except for map/reduce queries. To use map/reduce, use the MongoDB shell (note that temporary result collections will be viewable in Mongolab).

You can also add, edit, and delete individual documents from here. Bulk collection updates are not yet supported in this UI (although they are supported in the shell).

```
{
  "_id": {
    "$oid": "5507a75ae4b0603bec0c8ab5"
  },
  "document": {
    "userName": "Mouli",
    "Timestamp": "1426733798489",
    "text": "I had a fight with my friend today. He is my best friend "
  },
  "safe": true
}
```

```
{
  "_id": {
    "$oid": "5507a75be4b097f4fd5a8914"
  },
  "document": {
    "userName": "Sagar",
    "Timestamp": "1426733799868",
    "text": "He is really cunning. I never had a crush on his girlfriend "
  },
  "safe": true
}
```

```
{
  "_id": {
    "$oid": "5507a75ce4b0603bec0c8abb"
  },
  "document": {
    "userName": "Vishnu",
    "Timestamp": "1426733799899",
    "text": "This semester is really tuff. I am trying my best to get through.."
  },
  "safe": true
}
```

VI. Testing

Test case 1: Login using valid username and password.

Expected result: User should be able to login.

Result: Pass

Test case 2: Options in the entry screen of the dairy.

Expected result: Once user clicks the options button on the screen, he should be able to see add and edit item entry options.

Result: Pass

Test case 3: Clicking create an entry in the dairy.

Expected result: Once user implements creating an entry, He should be forwarded to a screen having title and body columns.

Result: Pas

Test case 4: Creation of an entry.

Expected result: User should be able to save the entry of title and body by giving the submit option in the entry creation screen.

Result: Pass

Test case 5: Saved entries should be displayed to the user.

Expected result: User should able to see the entries made by him on the screen once he saves the entries.

Result: Pass

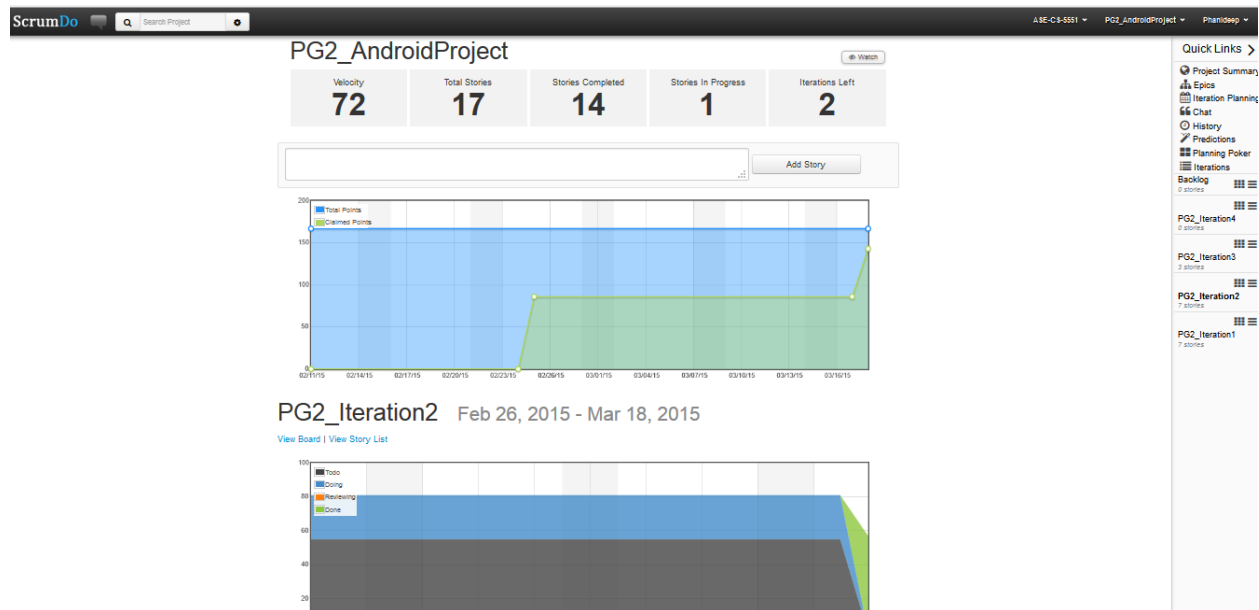
Test case 6: Deletion of an entry.

Expected result: User should be able to delete an entry by using this option.

Result: In progress

VII. Deployment

ScrumDo URL: http://www.scrumdo.com/projects/project/pg2_androidproject/summary



GitHub Link of the Project: https://github.com/vishnuchelle/My_Shared_Diary

We have uploaded the repost both in GitHub and Blackboard.

VIII. Project Management

The project is maintained in ScrumDo. Task assignments and descriptions are updated in ScrumDo stories. Below is an over view of the tasks performed by each individual.

ScrumDo URL: http://www.scrumdo.com/projects/project/pg2_androidproject/summary

Vishnu Chelle and Chandra Mouli: Create user interface. Store text data in the local SQL server. Create pin to lock the app. Save entries with timestamp.

Phanideep: Convert data to JSON format to store it in MONGO database.

Sundar Sagar: Created a connection between local SQL and MONGO database.

IX. Work To Be Completed

Creating user communities and share data across communities. As of now we have established mongodb server so our next task is to create communities. Future work will also include integrating individually implemented Yelp and Google Maps API's in to the same application.

Considering the time left we are removing audio recording feature for the user but will be considered under future work. For now our UI is basic, so our future work will also include developing a more user friendly UI. Background analytic function will be implemented in Iteration 3.

X. Bibliography

- [1] <http://www.techhive.com/article/2599838/the-best-apps-for-taking-notes.html>
<http://en.wikipedia.org/wiki/Evernote>
<https://evernote.com/>
- [2] <http://privatediary.net/>
<https://play.google.com/store/apps/details?id=app.diaryfree&hl=en>
- [3] <https://play.google.com/store/apps/details?id=com.aiguo.handydiary&hl=en>
- [4] <https://play.google.com/store/apps/details?id=com.android10.diarylog&hl=en>
- [5] <https://developers.facebook.com/docs/android>
- [6] <https://developers.google.com/maps/documentation/android/>
- [7] <http://www.yelp.com/developers/documentation>
- [8] <http://docs.mongodb.org/manual/>
- [9] <https://developer.android.com/training/basics/firstapp/index.html>