

# Saver ‘n Saviour

## Final Project Report

**Team 10 members:** Dig Vijay Kumar Yarlagadda, Sushma Mitta, Pooja Shekhar, Chandra Sekhar Janyavula

### **Project Deployment:**

#### **User Manual:**

##### **Table of contents:**

1. Introduction
2. Usage of the system
3. Error handling
4. Interaction with the system
5. Deficiencies

### **1. Introduction**

A Reminder system is important in today's busy lifestyle. There comes a time where we wish to have someone to remind us of our tasks so that we can complete them on time. Today's innovation in technology is minimising of human with applications, in many cases.

It's also a reminder system that keeps you informed! Many a times, we pass by a grocery store or gas station unaware ,when we still was in need of grocery or gas, realizing only to go back to the store at a later time. It's a waste of time and resources. We felt the need for an application which can help us manage our day-to-day works, remind us of appointments, household needs based on time and location. Our reminder application, Saver ‘n Savior, is a hybrid application which lets the user create tasks,mark them complete once finished ,create reminders and gives notifications in terms of messages and voice mails.

With our application, users can create tasks ,mark them complete once finished, can set up a geo-fence of frequently visited places ,and get push notification while crossing the location, set up voicemails to be reminded of appointments.

### **2. Usage of the system**

First time user needs to signup before logging in. We have set up social logins like facebook,twitter and github to ease sign up activity for the users. In the home page user is presented with different tabs to

create task lists, track completed tasks, and create reminders based on time and location. A user can create a task list and mark the item complete whenever a task gets completed. All the completed tasks can be tracked and deleted if needed. The user can create a reminder for a particular task for a specific date and time. The user will be reminded about the task by sending message to the registered mobile number. It can also send a voice mail. The user can set up geofence to get reminded of a related task when he is moving towards or moving away from the location, based on his choice. When the user crosses that location, application sends a push notification about the task.

Device constraints:

The app can be run on Android devices running Android 5.0 or higher and iOS devices running iOS 7 or higher.

The permissions that app needs are Contacts and Location. To obtain location based reminders, the device needs to turn on the GPS all the time.

Design choice:

The app is user friendly, with simple clicks on tab at the bottom, users can manage their task lists and reminders.

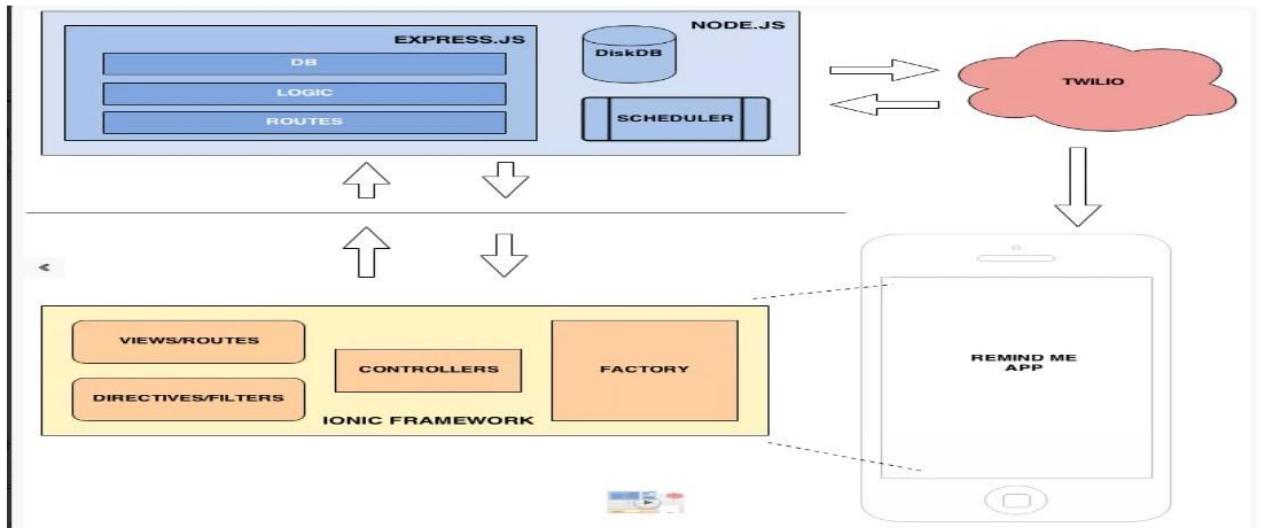
### **3. Error handling**

Maps not loading while creating Geolocation based reminders: Turn on device location and grant location permissions to the app.

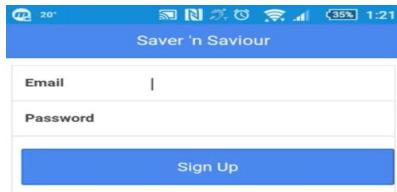
User needs to get a twilio number before using the app and get a token key and secret.

#### 4. User interactions

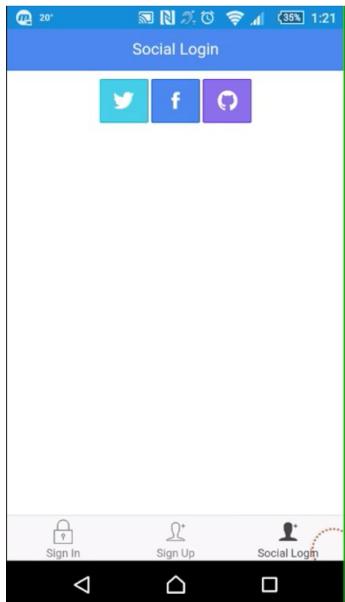
## Architecture



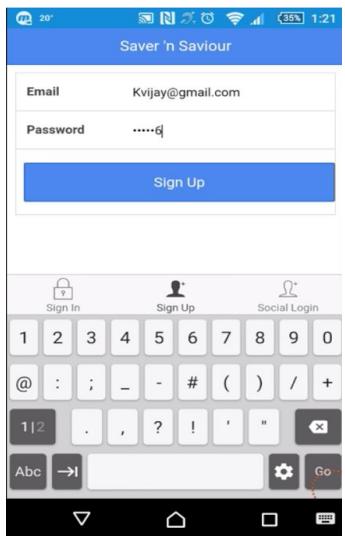
User logs in through below screen.



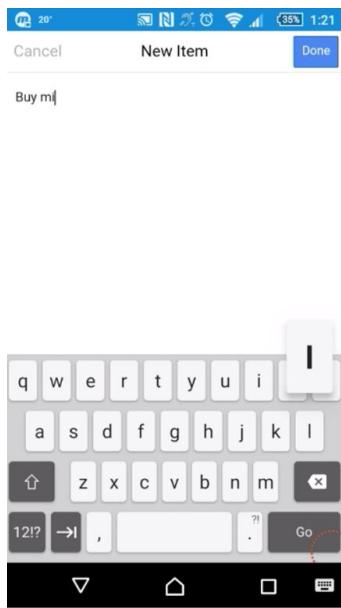
## Social logins



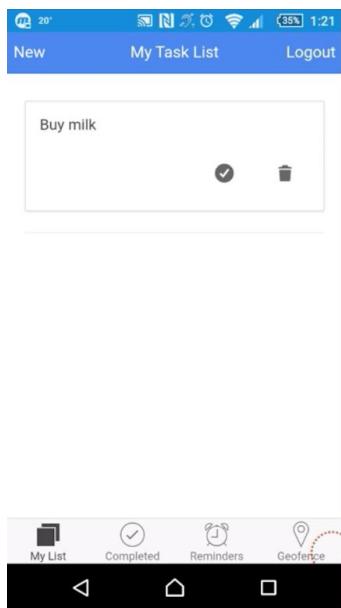
## User Sign Up Screen



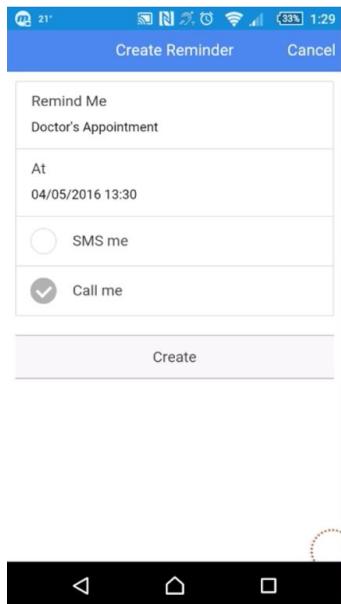
User creates a task.



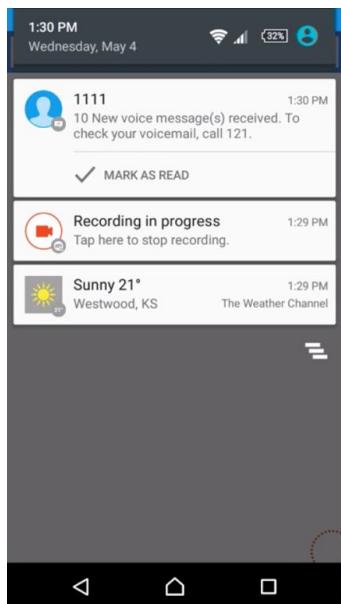
Task created.



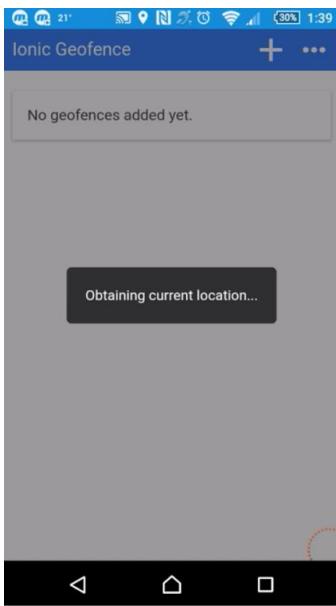
Time based reminder creation.



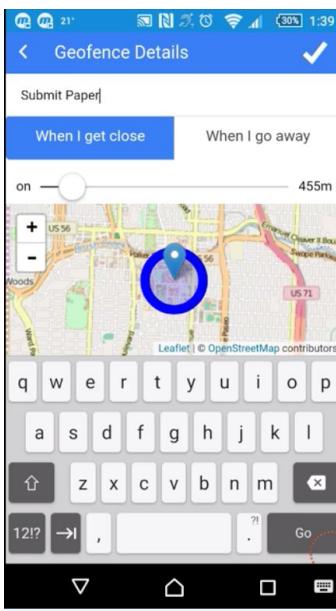
Voice mail received.



Create a new Geofence.



Edit the Geofence



## 6. Bugs and Deficiencies:

Bugs:

1. Sometimes switching between task creation and reminder creation features just doesn't work.

2. Even if you choose only to receive messages, app will send a voicemail.

Deficiencies:

1. Sharing of tasks is implemented, but not integrated into our application.

## Project Management:

Our initial estimation of the amount of effort we need to put into the project was proved wrong, we put in much more effort and time into our project. We tried to implement some features like speech recognition and gas station tracker, but we fell short of time. Except for those features, our project met all the requirements we have tried to implement and some additional features too.

Agile process was very helpful in developing our project features incrementally. The core features are improved in each increment and new features are added from time-to-time. Due to agile project development, we have complete interaction with our app from the beginning, it helped in resolving errors and helped in perfecting our core features of the application.

We very much hope to use agile development in our future projects too.

We have followed our project schedule for implementing core features, we only changed our schedule for additional features.

Management structure:

Sushma Mitta: Coding, unit testing, integration testing

Pooja Shekhar: Coding, unit testing, documentation, database maintenance

Dig Vijay Kumar Yarlagadda: Coding, unit testing, deployment

Chandrasekhar Janyavula: Documentation

One thing we would like to change if we develop this project realtime is using scalable technologies, our app would probably can't handle heavy loads.

Ionic was introduced a little too late in the class. We suggest introducing Ionic early in the class, students would benefit using it with the project.

Link to agile management console:

Boards: [https://github.com/SCE-UMKC/AESP16\\_Saver\\_n\\_Savior\\_Team10#boards?milestoneId=1](https://github.com/SCE-UMKC/AESP16_Saver_n_Savior_Team10#boards?milestoneId=1)

# **First Increment Report**

## **Project Goal and Objectives:**

### **Overall goal:**

To create an application that is capable of tracking all the household needs and gas refills at best prices.

### **Specific Objectives:**

The primary focus of this project is creating a reminder application that is capable of tracking all the needs of essentials like groceries, gas refills based on its usage. This will also find a store where an item is available at the cheapest price.

### **Specific features:**

The salient features of the application are:

- To do lists for household needs
- Notifications for grocery refill
- System wide search for groceries, restaurants and gas stations
- Track of shopping history
- Sharing shopping lists with family members and friends
- Complete information about grocery stores nearby, including store timings, distance to store etc.,
- Voice reminders
- Optimization of travel routes for best possible prices or least amount of time required.

## **Project Plan**

### **Schedule:**

The project is divided into equal parts such as design, implementation, testing and deployment.

The below screenshots shows the schedule of the project and contributions of each team members.

## Issues:

This screenshot shows the GitHub Issues page for the repository `digvijaykumaryarlagadda / ASE_Team_10_Spring16`. The page displays 7 open issues and 3 closed issues. The open issues are:

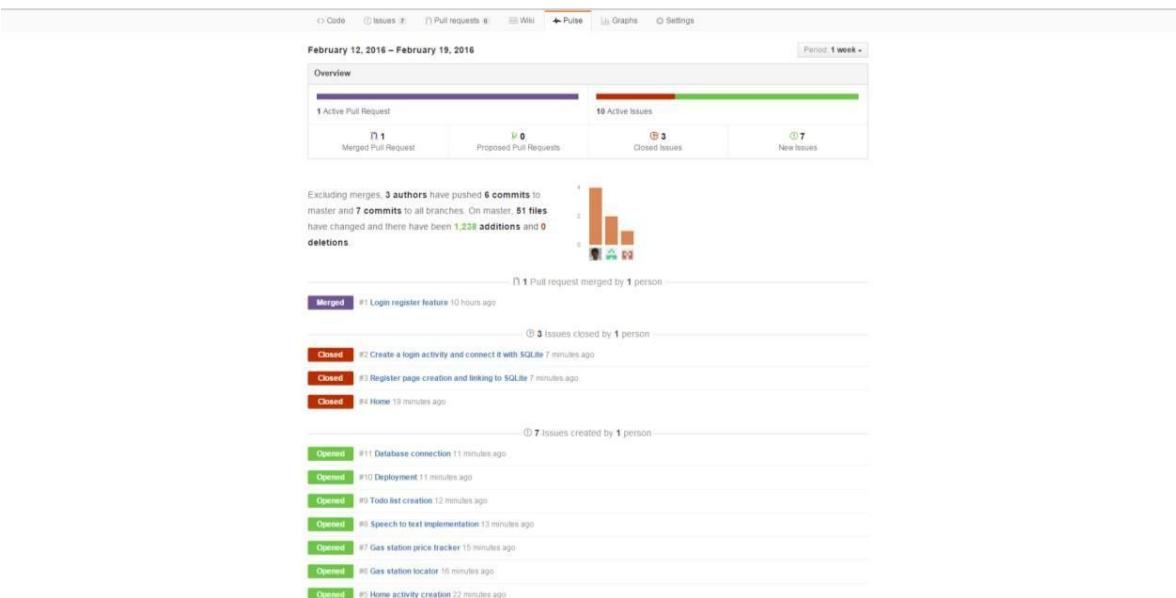
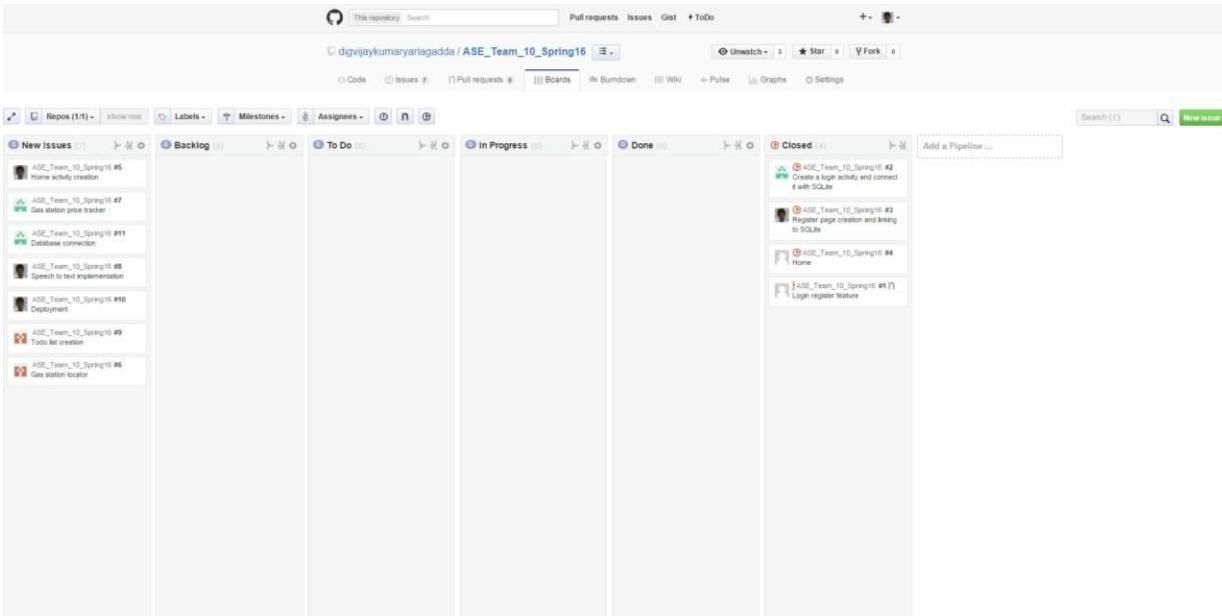
- Database connection** (#11) opened 4 minutes ago by `digvijaykumaryarlagadda` · Increment IV
- Deployment** (#10) opened 4 minutes ago by `digvijaykumaryarlagadda` · Increment IV
- Todo list creation** (#9) opened 4 minutes ago by `digvijaykumaryarlagadda` · Increment III
- Speech to text implementation** (#8) opened 6 minutes ago by `digvijaykumaryarlagadda` · Increment III
- Gas station price tracker** (#7) opened 7 minutes ago by `digvijaykumaryarlagadda` · Increment III
- Gas station locator** (#6) opened 8 minutes ago by `digvijaykumaryarlagadda` · Increment III
- Home activity creation** (#5) opened 15 minutes ago by `digvijaykumaryarlagadda` · Increment II

The GitHub interface includes standard navigation and search tools at the top, and footer links for Status, API, Training, Shop, Blog, About, and Pricing.

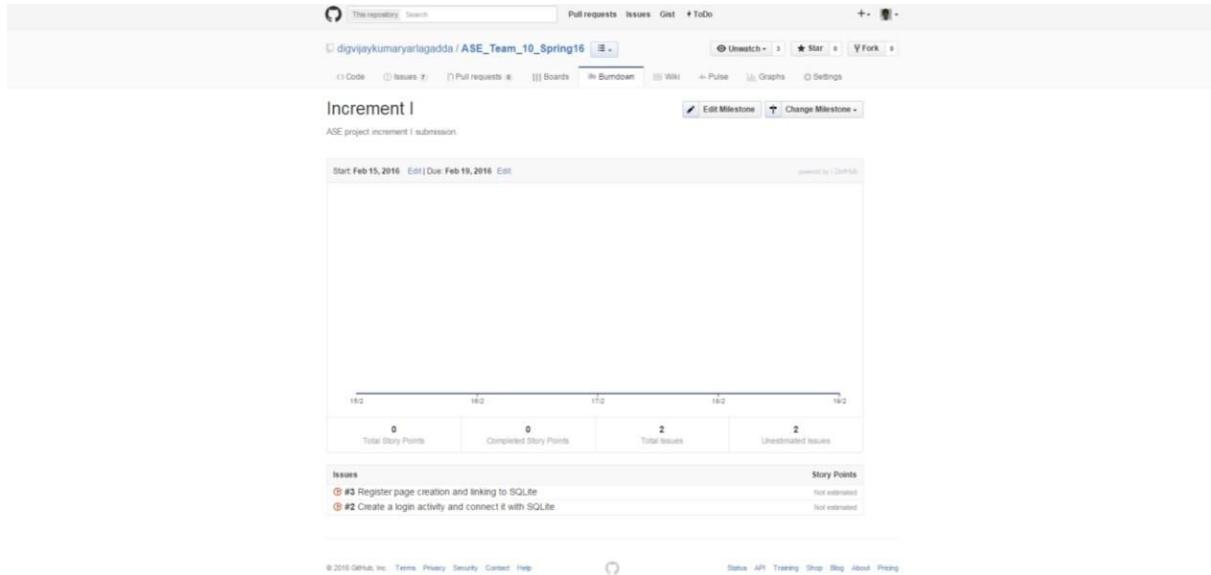
This screenshot shows the GitHub Issues page for the same repository, but with a different filter applied. The page displays 7 open issues and 3 closed issues. The closed issues are:

- Home** (#4) opened 17 minutes ago by `digvijaykumaryarlagadda`
- Register page creation and linking to SQLite** (#3) opened 18 minutes ago by `digvijaykumaryarlagadda` · Increment I
- Create a login activity and connect it with SQLite** (#2) opened 21 minutes ago by `digvijaykumaryarlagadda` · Increment I

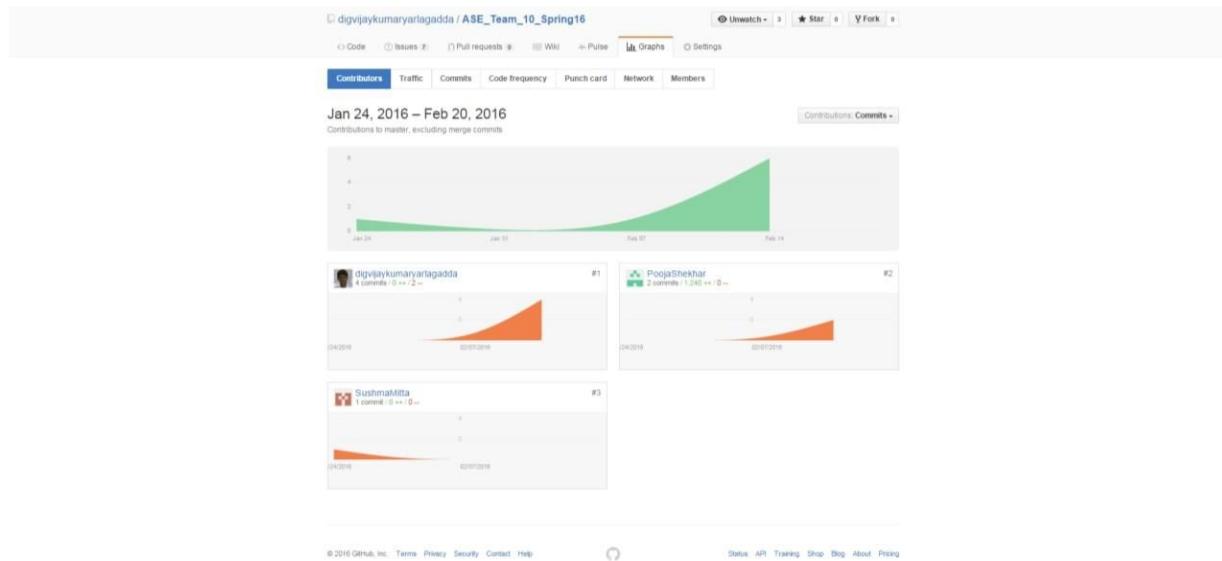
The GitHub interface includes standard navigation and search tools at the top, and footer links for Status, API, Training, Shop, Blog, About, and Pricing.



## Burndown chart:



## Team members contributions:



## Design:

The architecture of the project has been planned in the first increment. The login and register page has been created in Android Studio and the application is linked to a SQLite server.

Existing services:

- [1]. <https://github.com/SimplicityApks/ReminderDatePicker>
- [2]. <https://github.com/wdkapps/FillUp>

APIs:

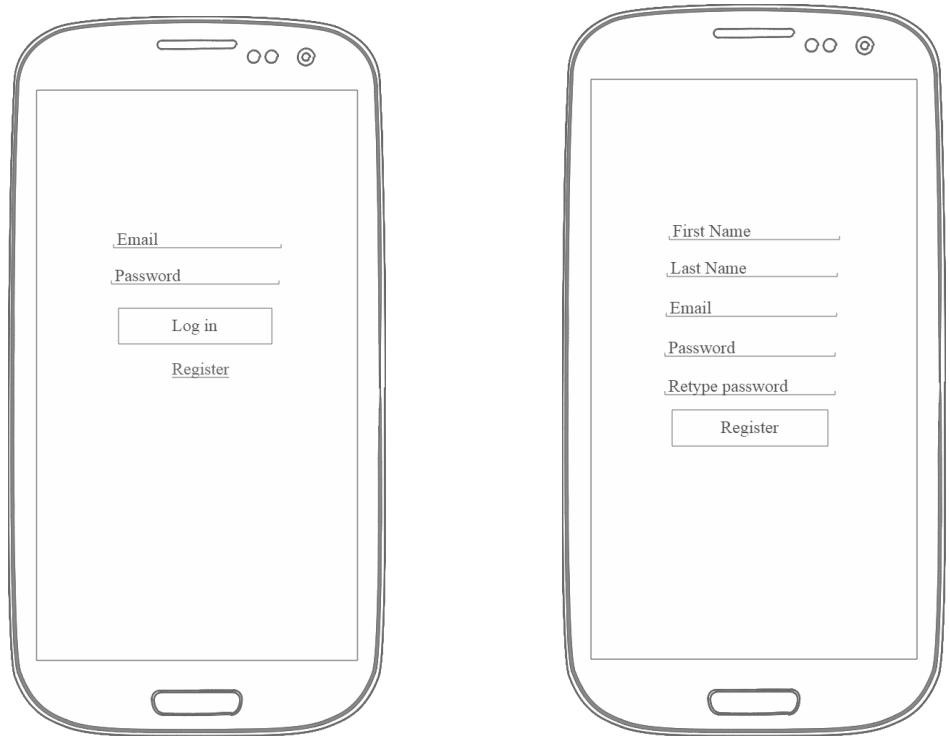
The APIs which will be used in this project are

- Google Places API for obtaining nearby restaurants for a selected route
- Yelp API for restaurant reviews
- Yellow API to obtain gas prices
- Web speech API for speech to text conversion

Mockups:

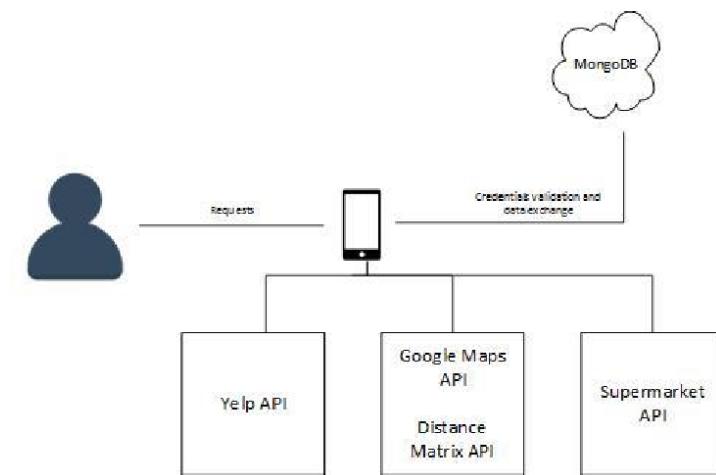


### Wireframes:

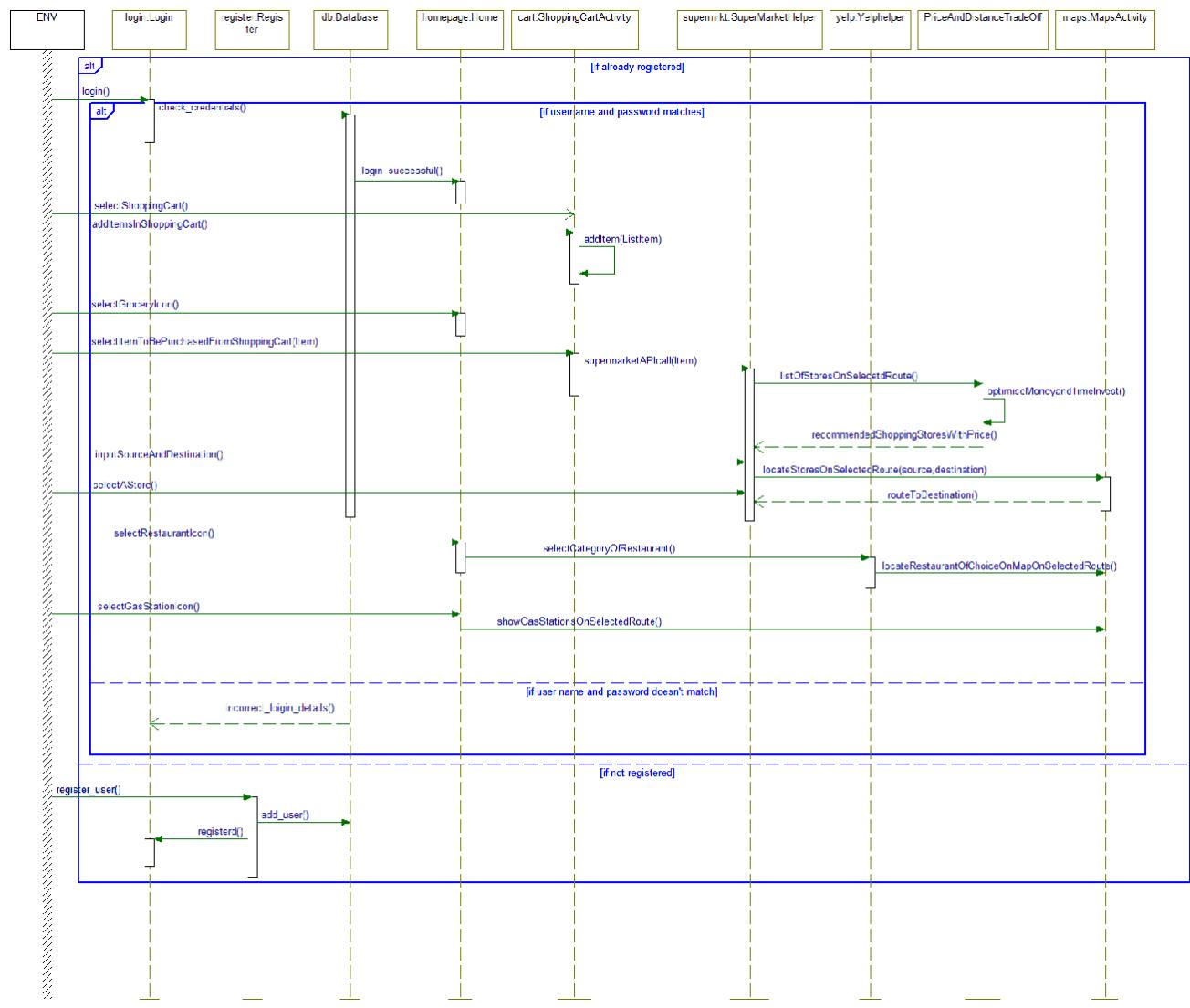


### Architecture diagram:

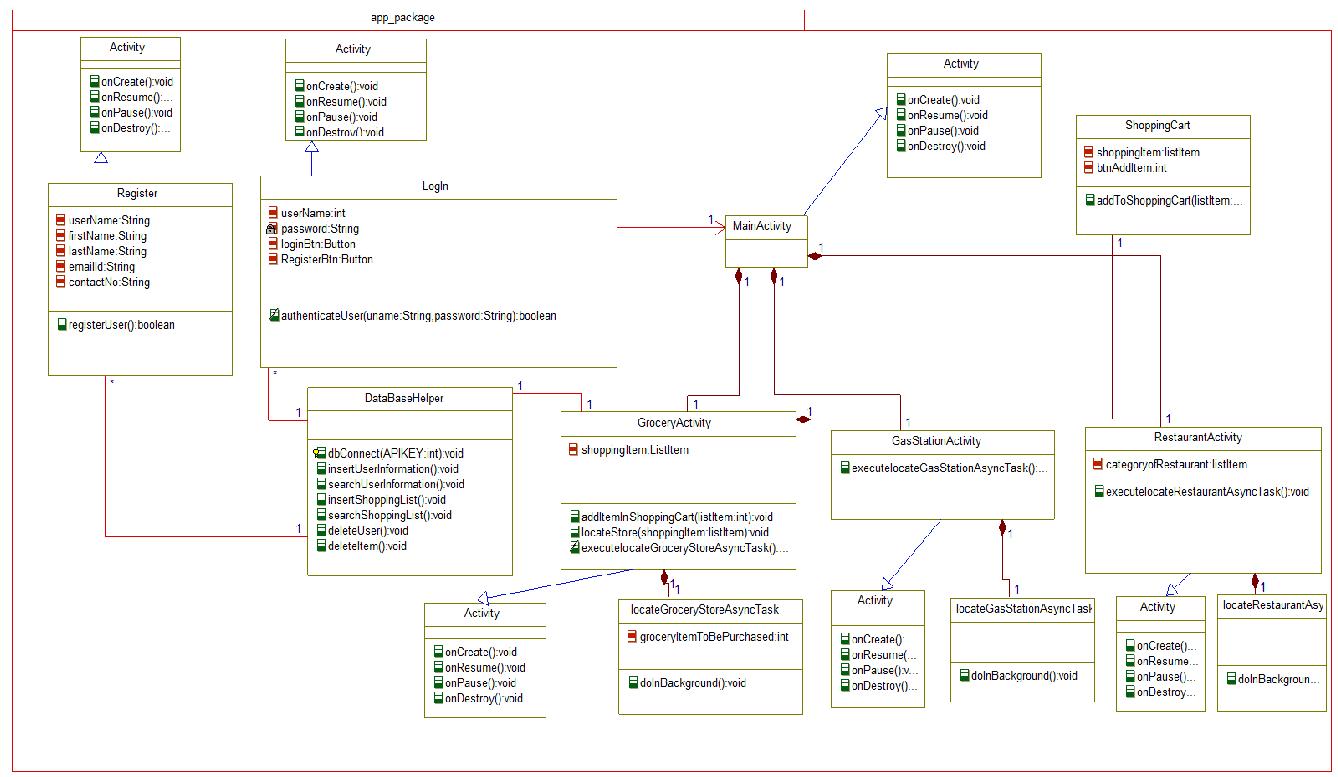
The app accepts user inputs, maps user's request for grocery item search, gas station search, restaurants search belonging to a particular category to the relevant API's shown below and provides responses to the user. The app interacts with MongoDB whenever any data is required.



## UML Sequence diagram:



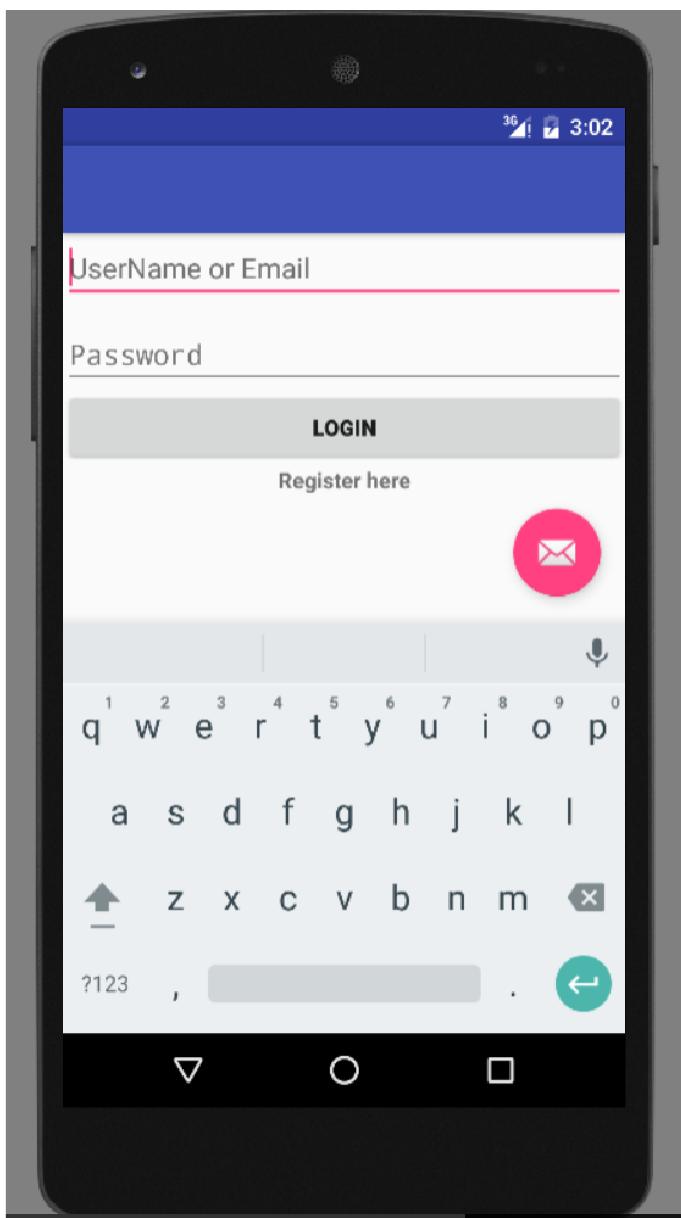
## UML Class diagram:

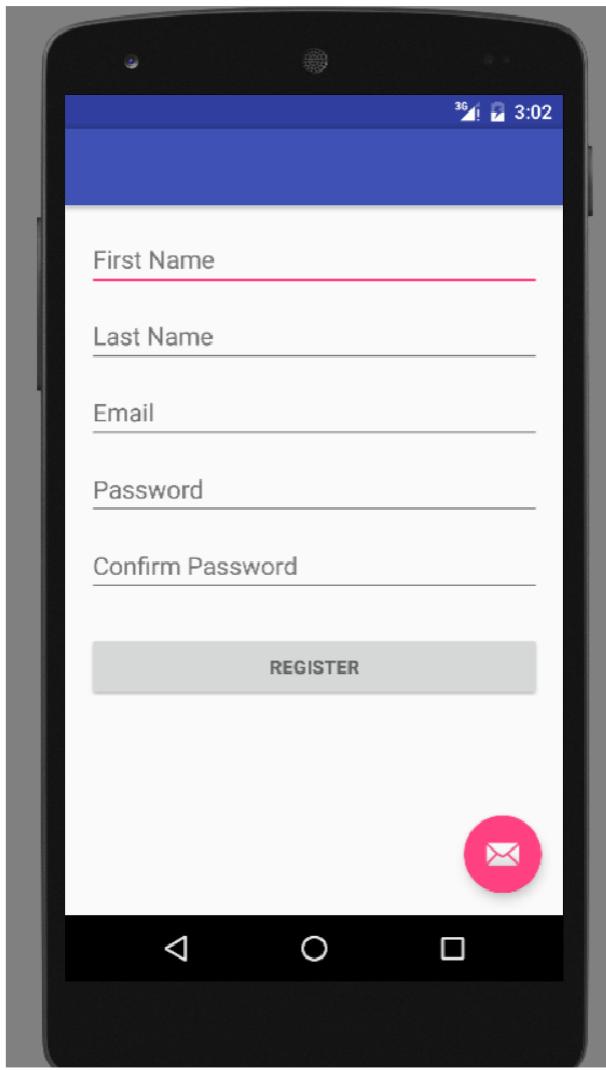


## Deployment:

The application in current form uses SQLite. However we have established connection with MongoDB platform and this will be used for later phases of the project.

Project screenshots:

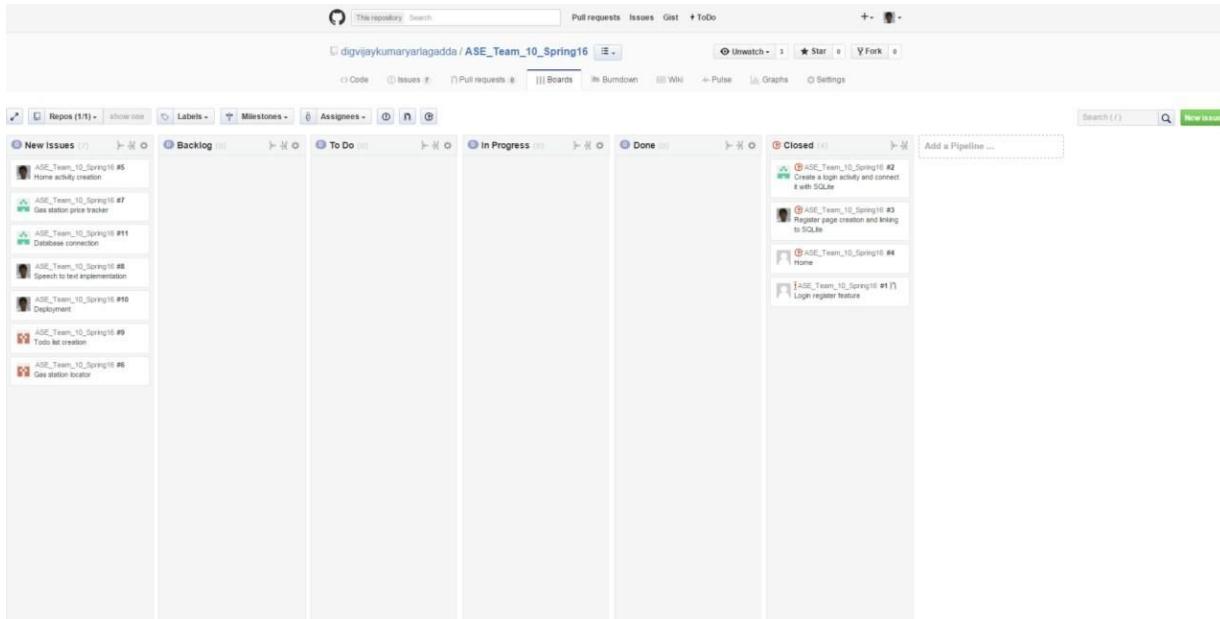




Project GitHub repository:

[https://github.com/digvijaykumaryarlagadda/ASE\\_Team\\_10\\_Spring16](https://github.com/digvijaykumaryarlagadda/ASE_Team_10_Spring16)

## Project Management:



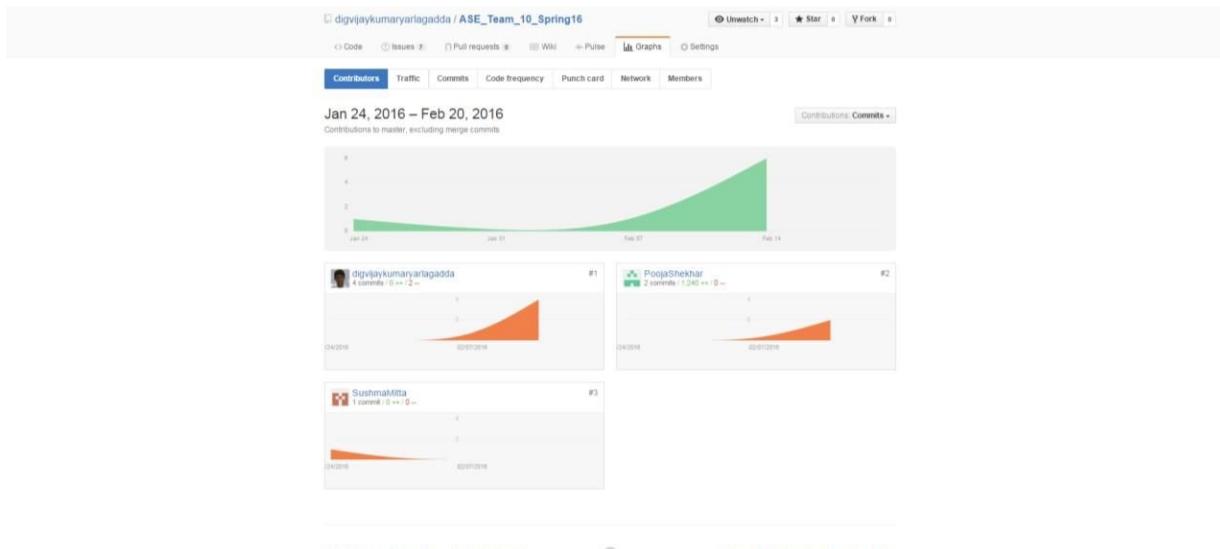
## Team members contribution:

Pooja Shekhar (25%) – Login page creation, setting up MongoDB connection, UML diagrams

Dig Vijay Kumar Yarlagadda (25%) – Wireframes, UML diagrams, Project report, Android app coding

Sushma Mitta (25%) – Register page creation and setting up SQLite, UML diagrams

Chandra Sekhar Janyavula (25%) – Project report and Architecture diagram



Bibliography:

- [1]. <http://www.mygasfeed.com/keys/intro>
- [2]. <https://www.yelp.com/developers/documentation/v2/overview>
- [3]. <https://developers.google.com/web/updates/2013/01/Voice-Driven-Web-Apps-Introduction-to-the-Web-Speech-API?hl=en>

## **Second Increment Report**

### Introduction:

The application Saver 'n Saviour provides users a convenient way to manage household tasks. It also provides maps to nearby, restaurants, grocery stores and gas stations.

In this increment, we have developed location based reminders: users are reminded whenever they reaches nearby a grocery store.

### Objectives:

The primary objectives are:

- Create task list
- Establish MongoDB connection to store user data
- Create location based reminders

### Features:

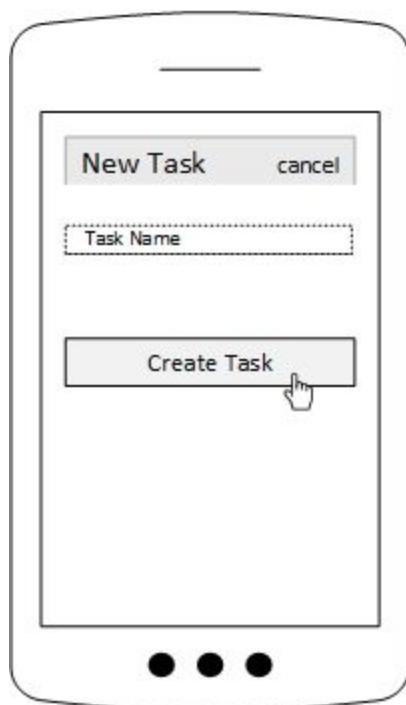
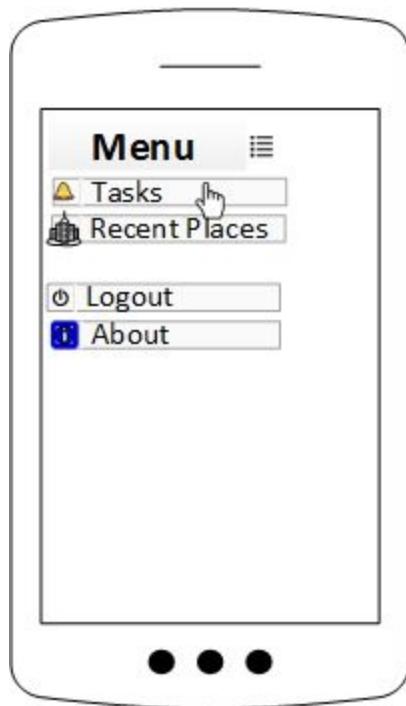
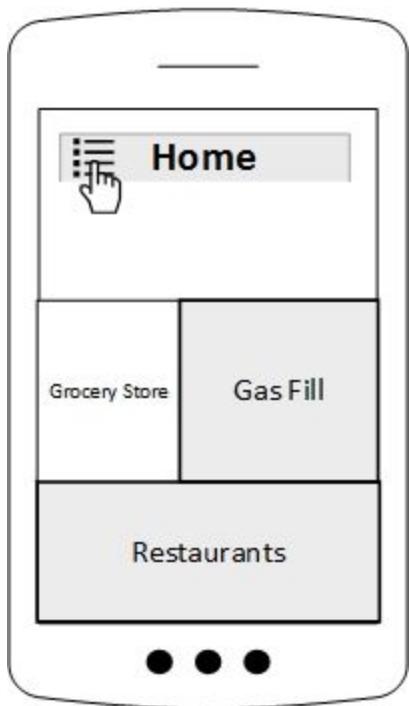
The salient features of the application so far are:

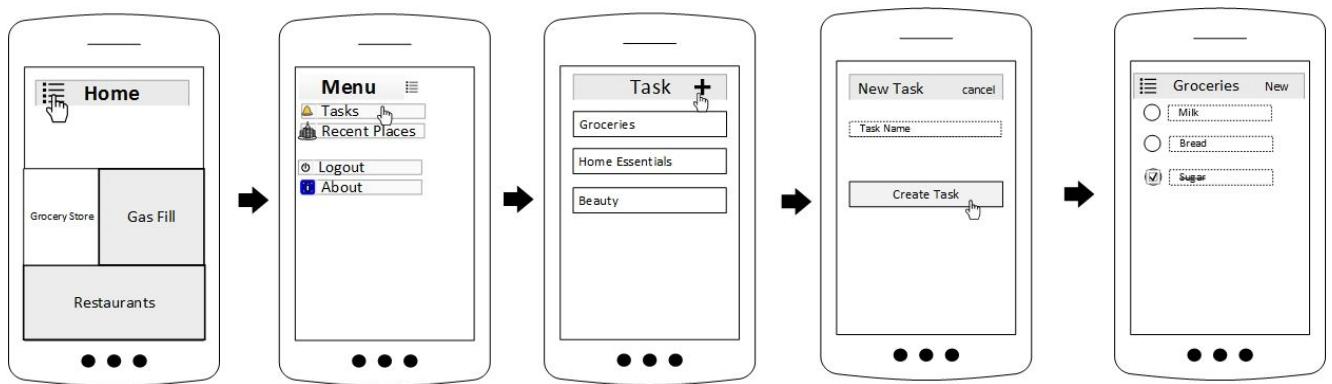
- Register and Login features
- Create to do lists for different categories and modify, delete or check mark tasks
- Store and synchronize all the user data with database
- Location based reminders

### Existing Services/API:

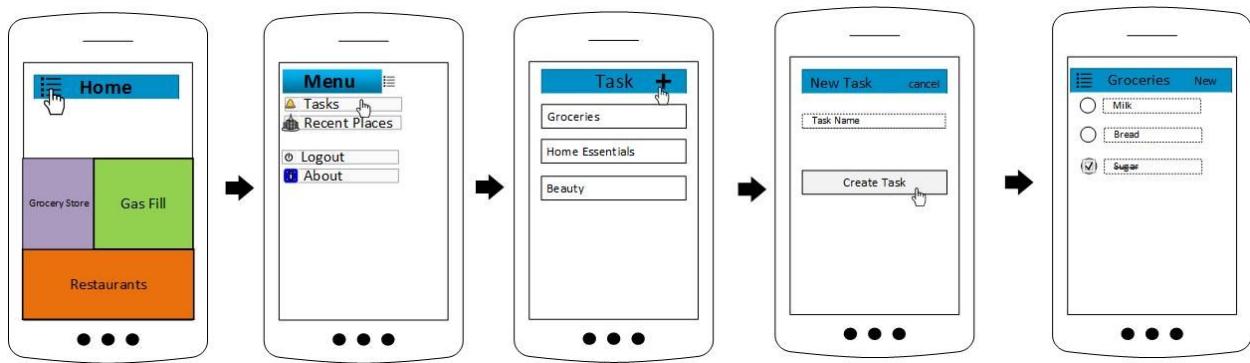
- Mongo Lab
- Google Places API
- Walmart API

Wireframes:

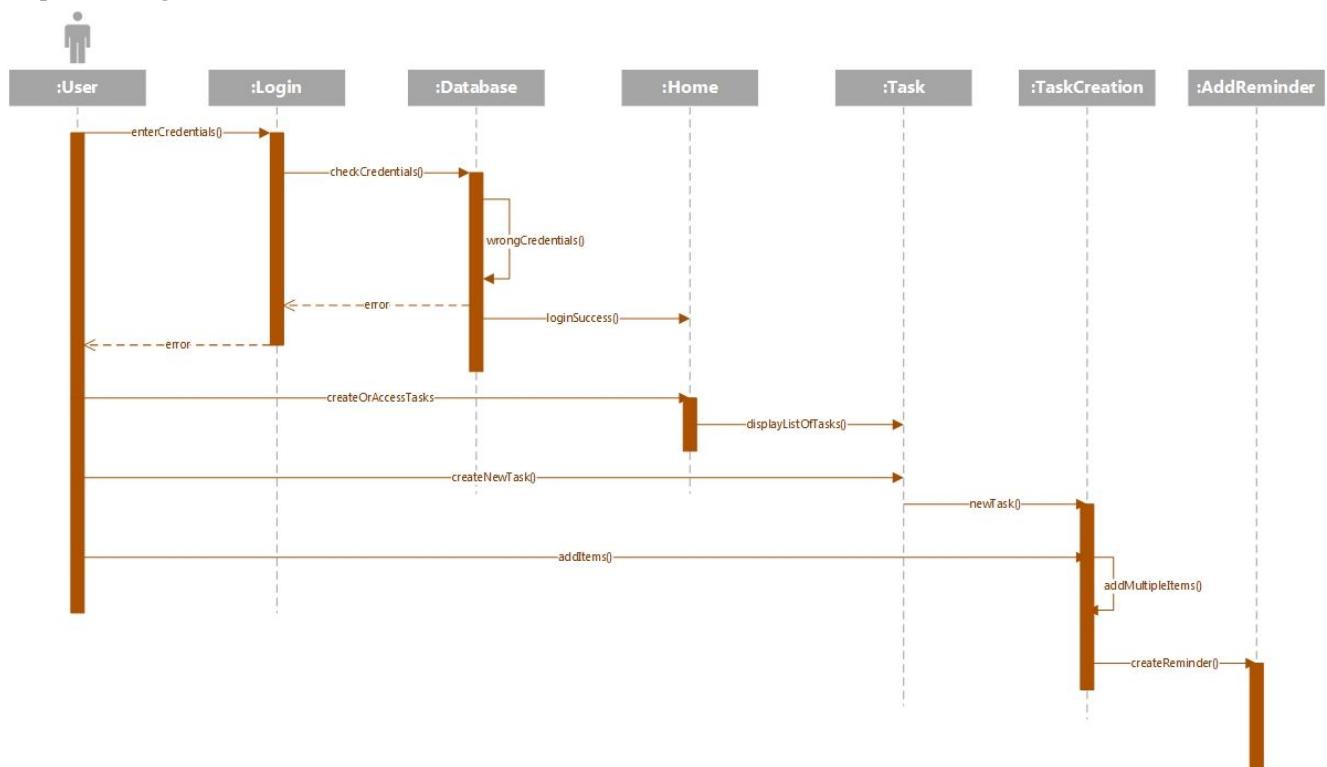




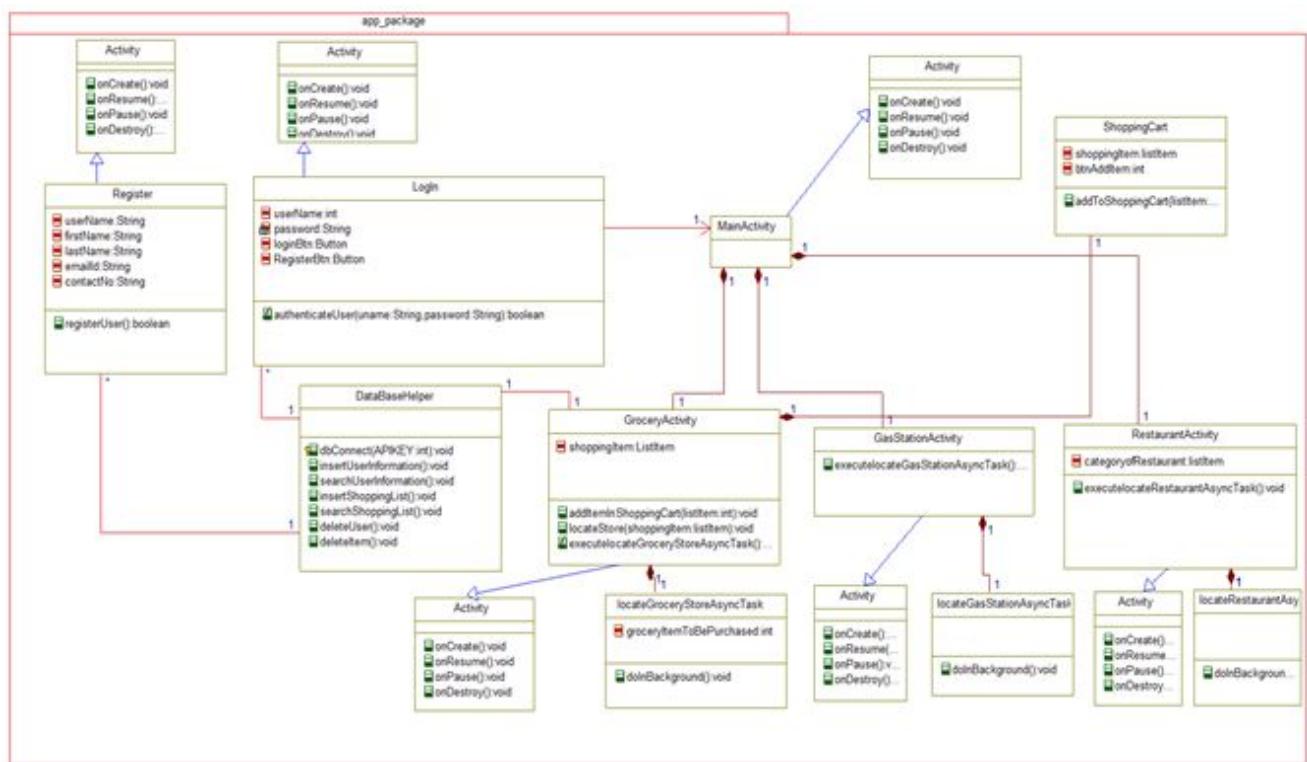
### Mockup:



### Sequence diagram:



## Class diagram:



## Use case diagram:



Architecture diagram:

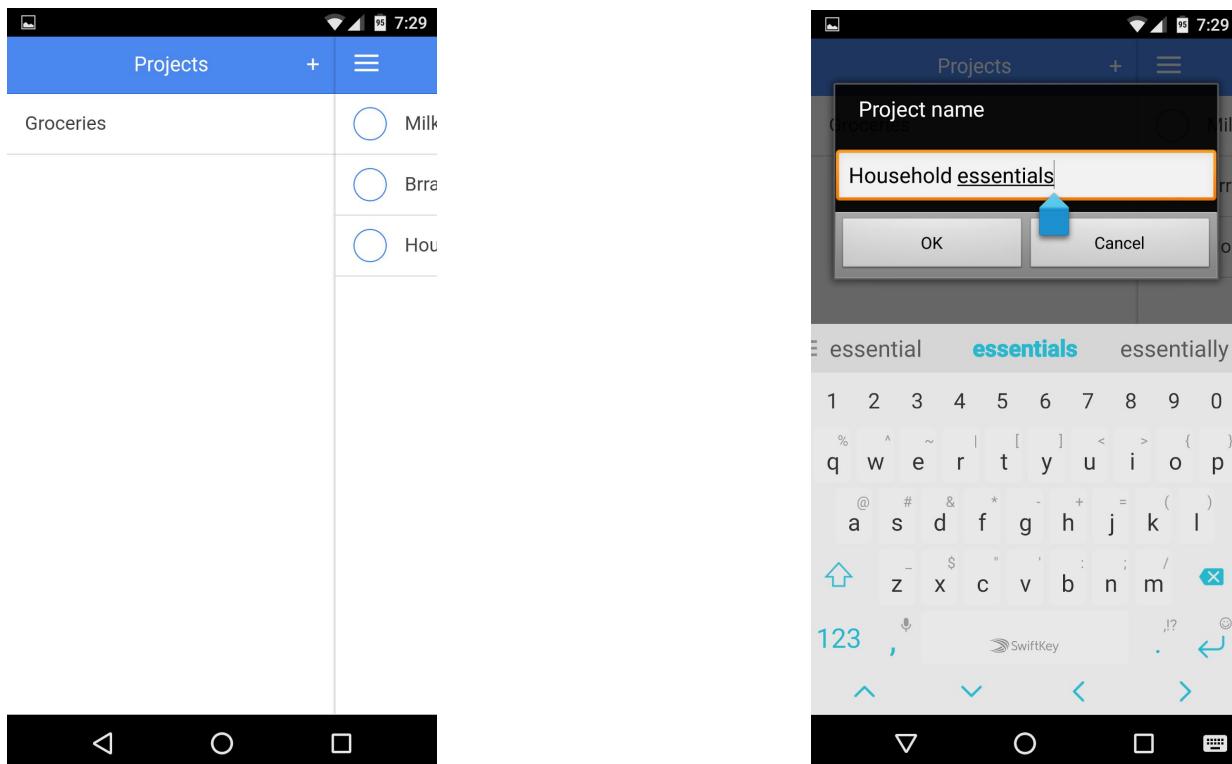
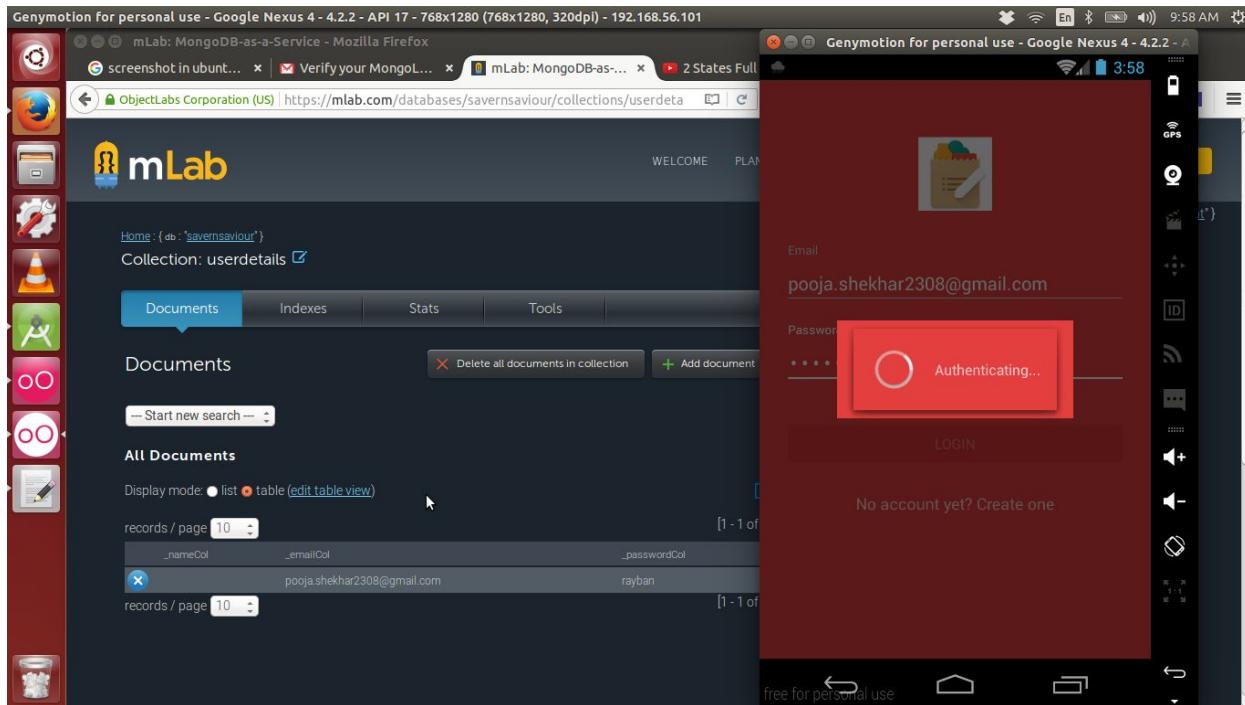


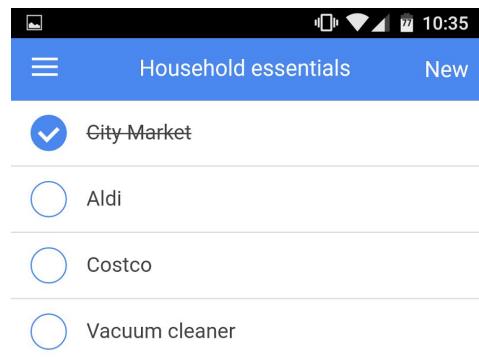
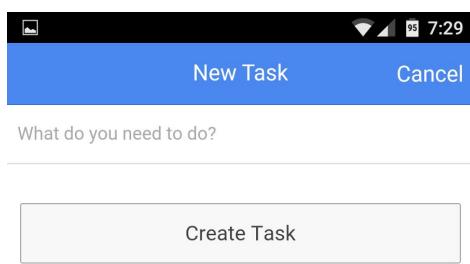
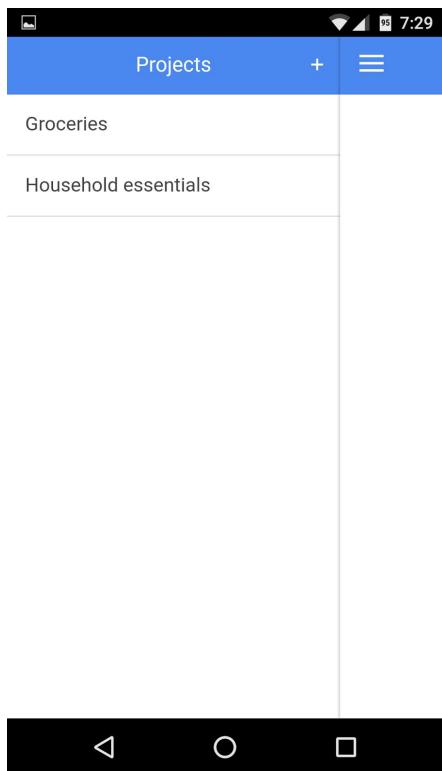
Implementation:

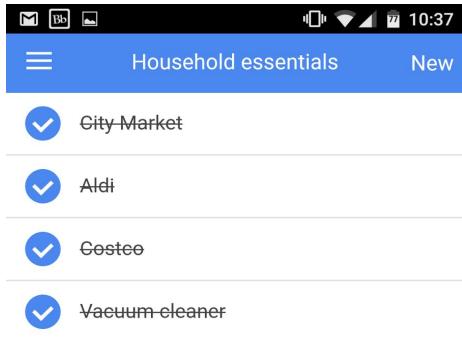
This android application has been implemented using ionic framework which is based on Angular.js. The data has been hosted on a MongoDB database.

Deployment:

The application has been deployed to Nexus 5 running Android 6.0. The below screenshots shows the creation, updation and deletion of to do lists along with data updation into MongoDB server.

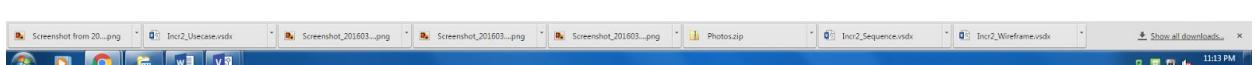
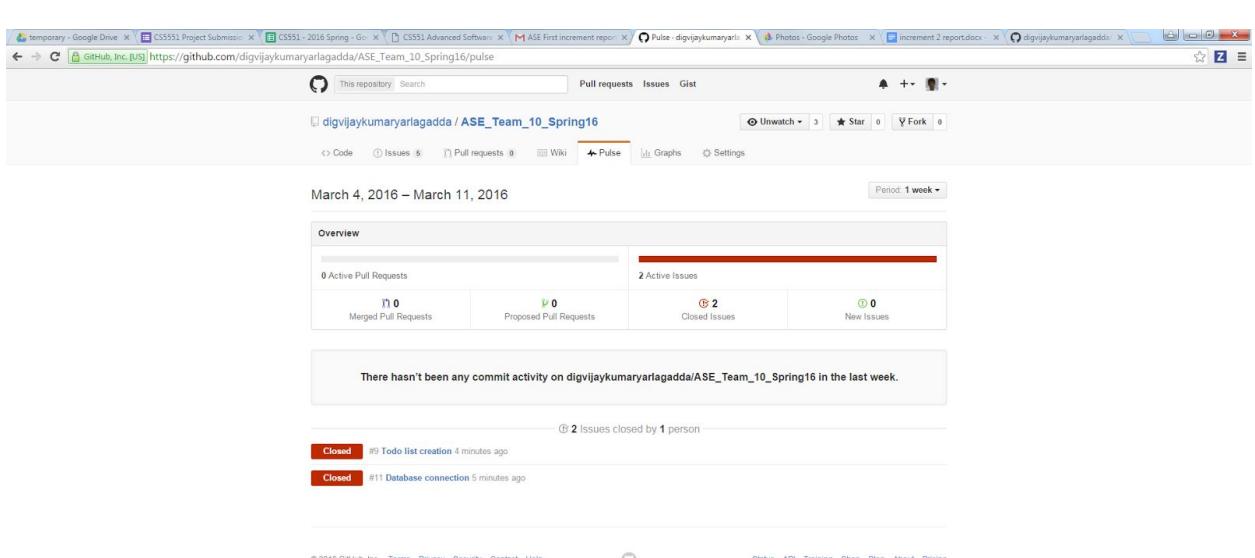
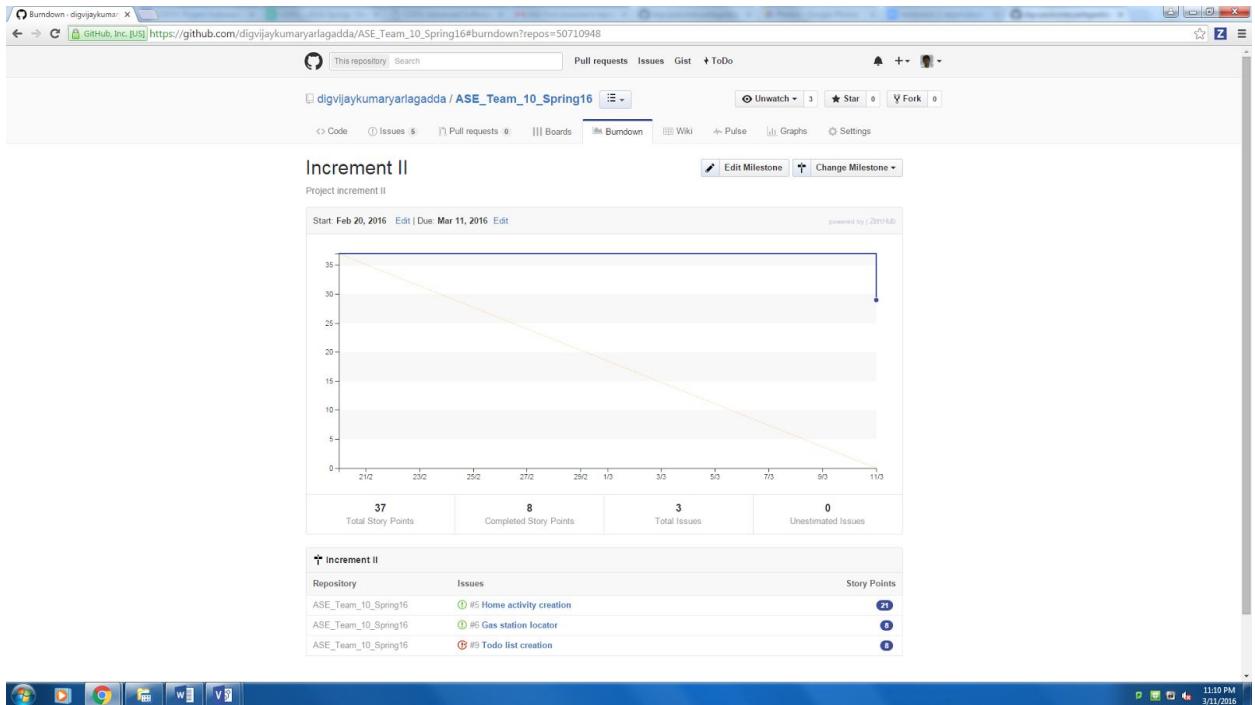


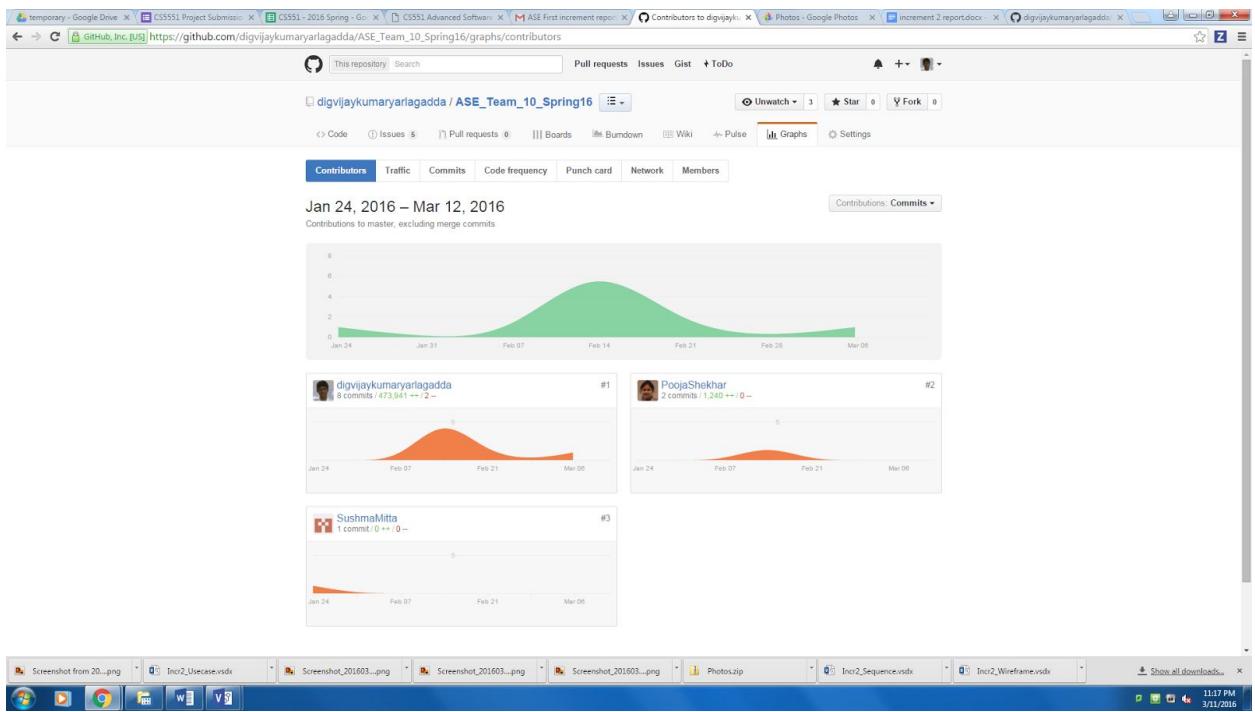




### Project management:







## Project GitHub repository:

[https://github.com/digvijaykumaryarlagadda/ASE\\_Team\\_10\\_Spring16](https://github.com/digvijaykumaryarlagadda/ASE_Team_10_Spring16)

## Team members contribution:

Pooja Shekhar (25%) – Login page creation, setting up MongoDB connection, UML diagrams  
 Dig Vijay Kumar Yarlagadda (25%) – Android app coding using Ionic framework, UML diagrams, Project report  
 Sushma Mitta (25%) – Register page creation, UML and Architecture diagrams  
 Chandra Sekhar Janyavula (25%) – Project documentation

## Bibliography:

- [1]. <http://ionicframework.com/>
- [2]. <https://github.com/driftyco/ionic-todo>

# Third Increment Report

## **Introduction:**

The application Saver 'n Saviour aims to provide users with a suitable way to manage different tasks. It helps users with reminders and also provides options like routes to nearby restaurants, grocery stores and gas stations.

In this increment, we have extended our application to send push notification of task reminders and also sharing of the task lists using device native applications like mail and chat applications. We have also included maps to search for nearby stores.

## **Objectives:**

The primary objectives are:

- Send reminders for created tasks
- Share the created task lists
- Search nearby grocery stores on map

## **Features:**

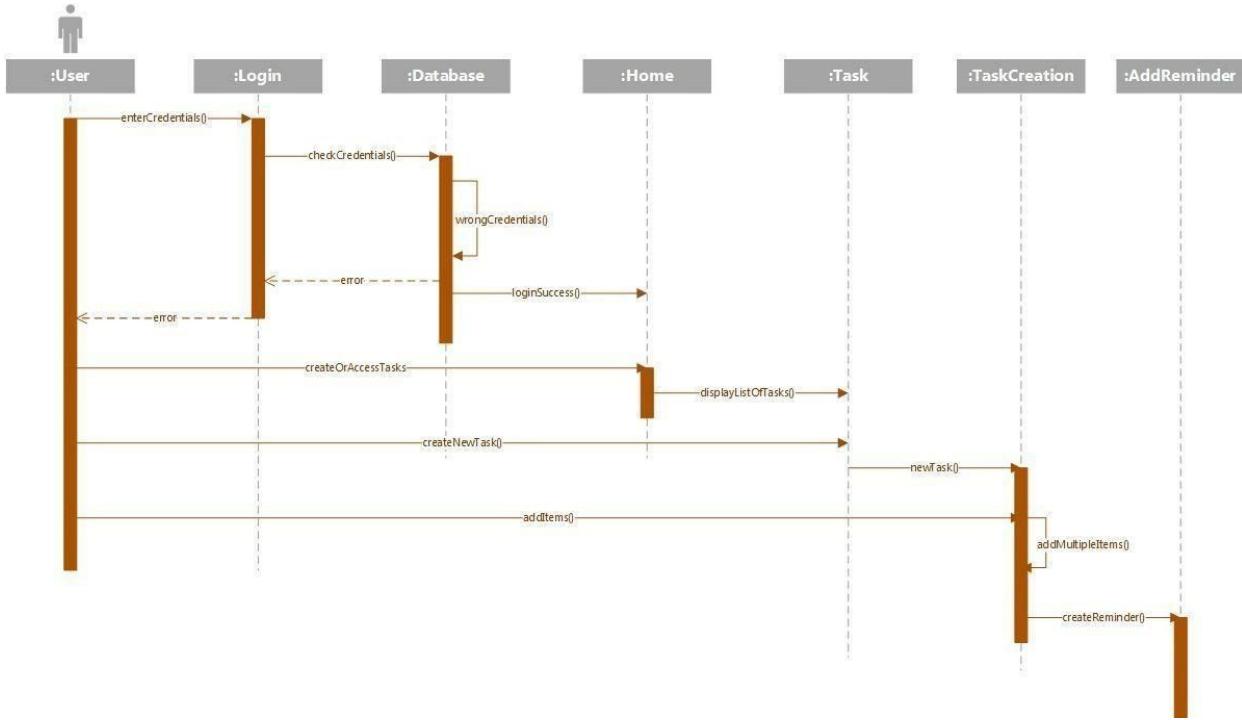
The salient features of the application so far are:

- Register and login features along with social login feature
- Create to do lists for different categories and modify, delete or check mark tasks
- Store and synchronize all the user data with database
- Send push notifications for the tasks created
- Share the list through mail or chat applications (platform dependent)
- Search nearby stores

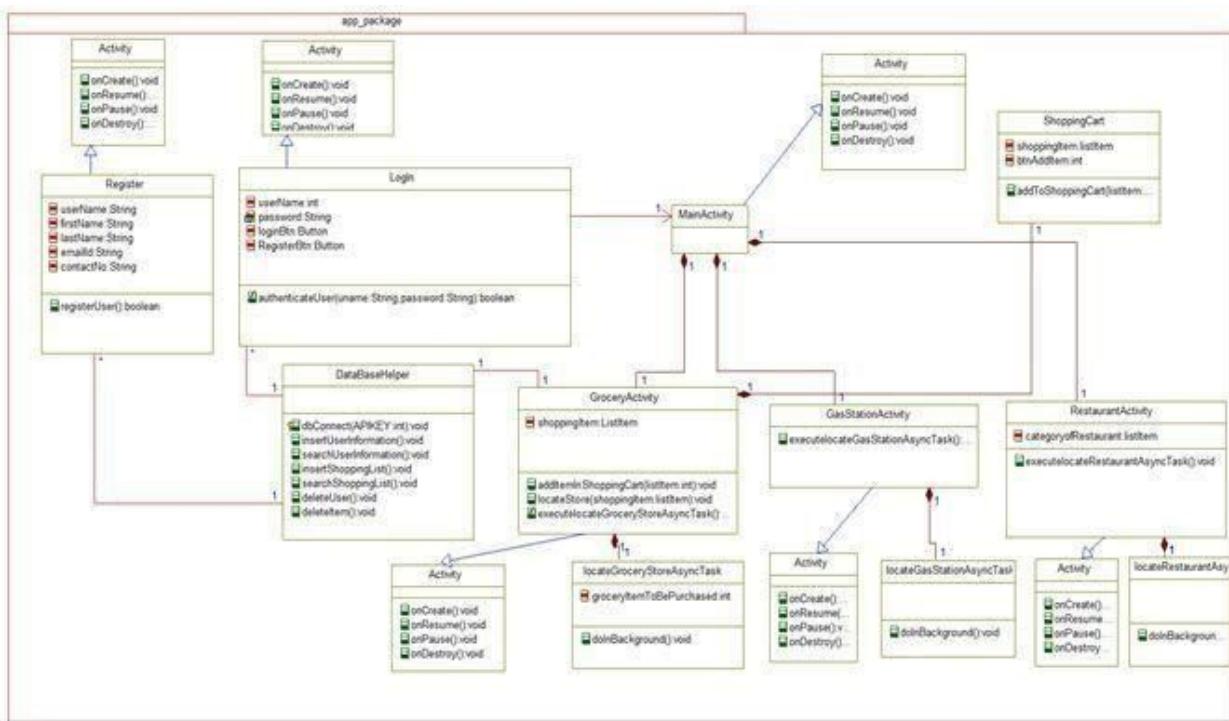
## **Existing Services/API:**

- Mongo Lab
- Google Places API
- Yelp API
- Super Market API

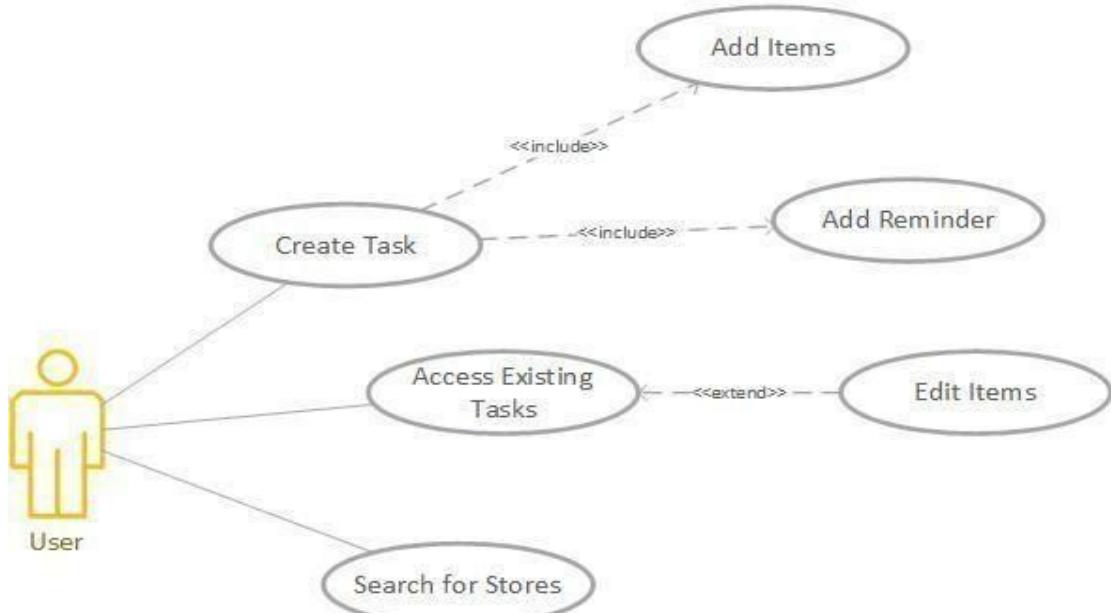
## Sequence Diagram:



## Class diagram:



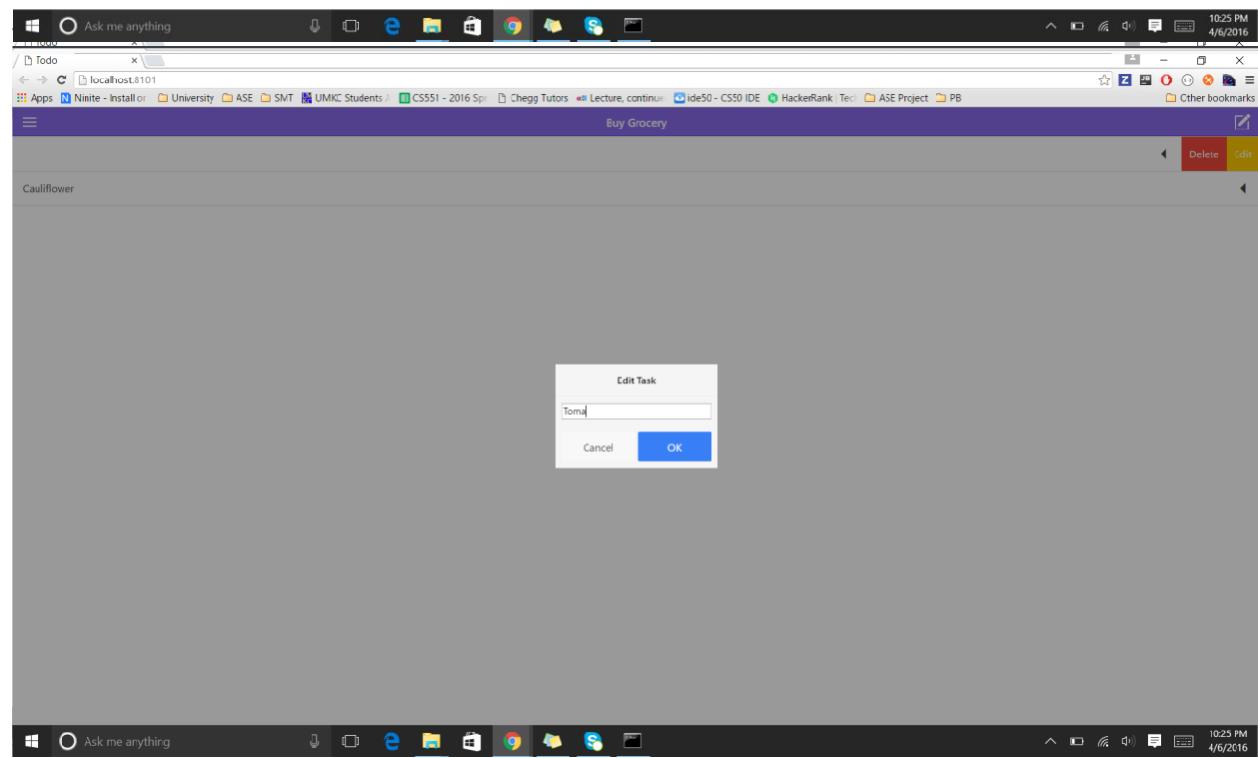
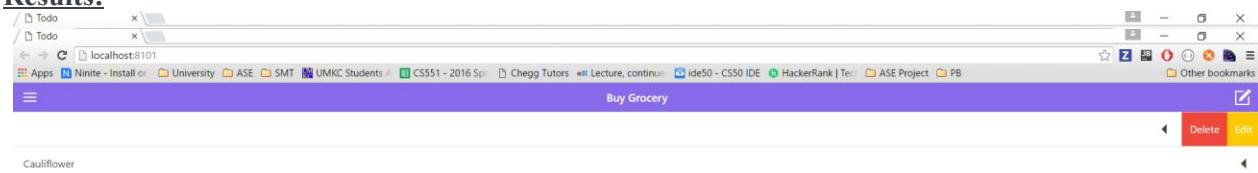
Use case Diagram:

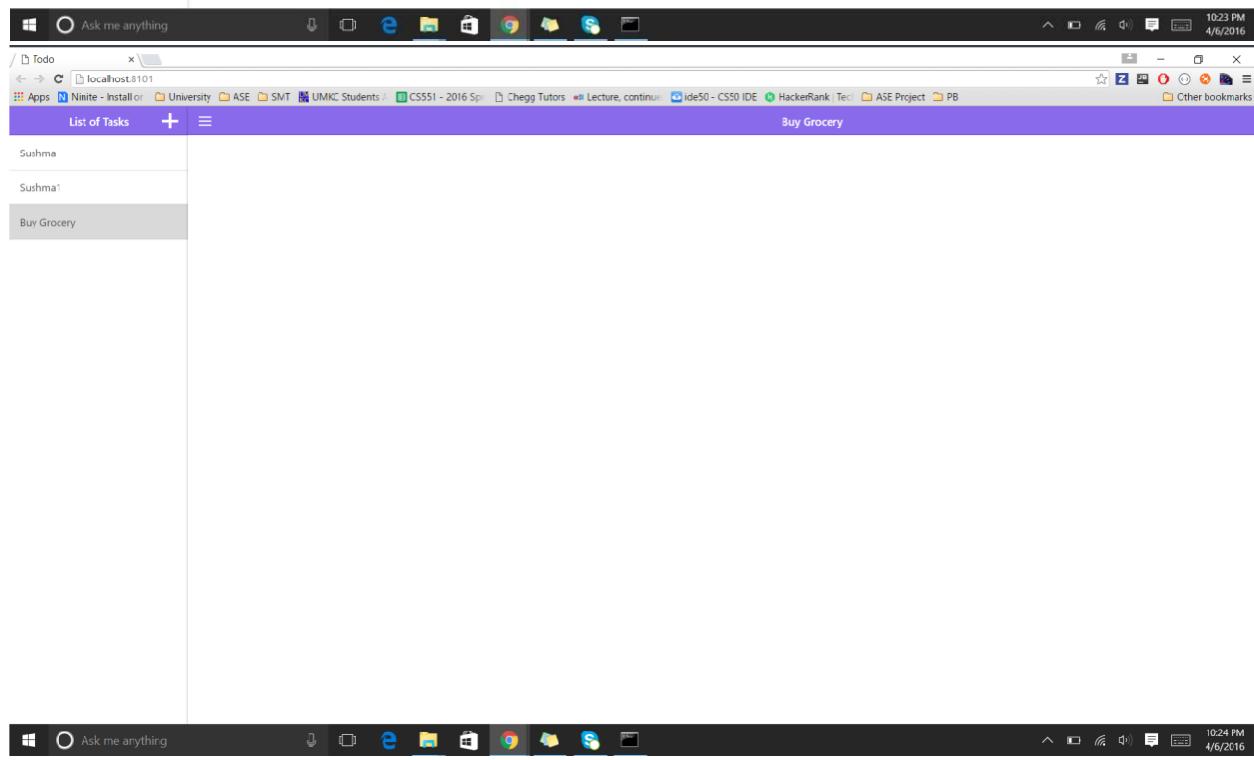
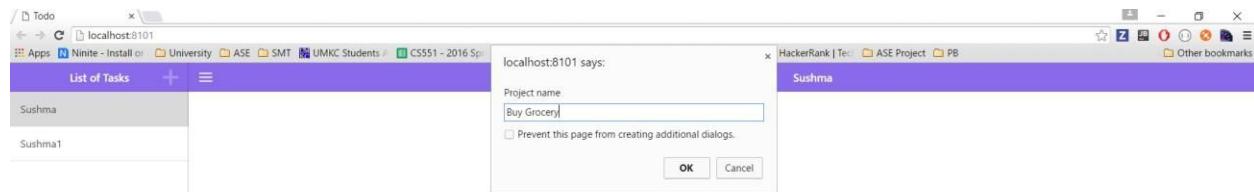


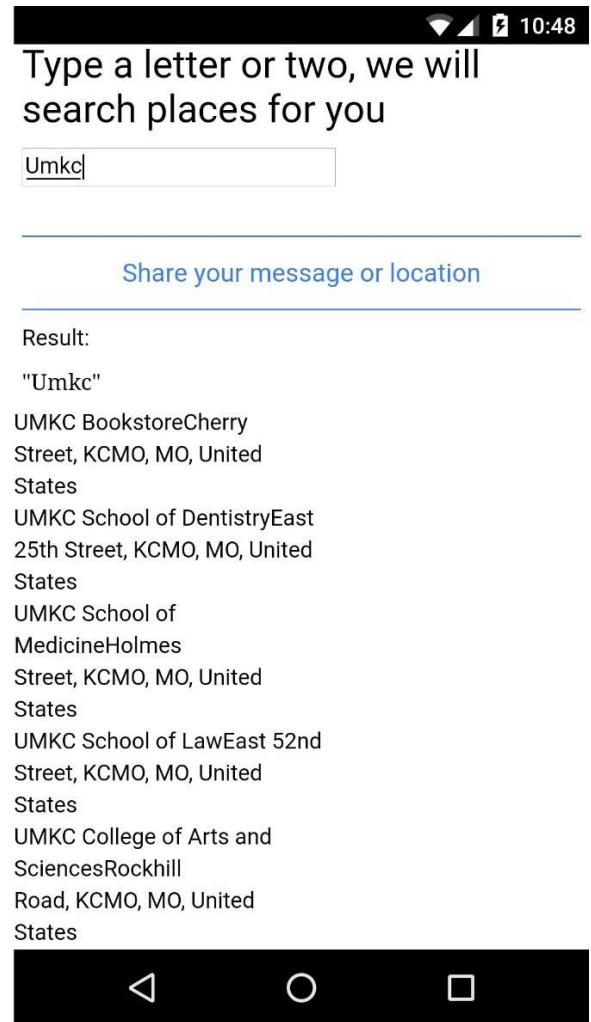
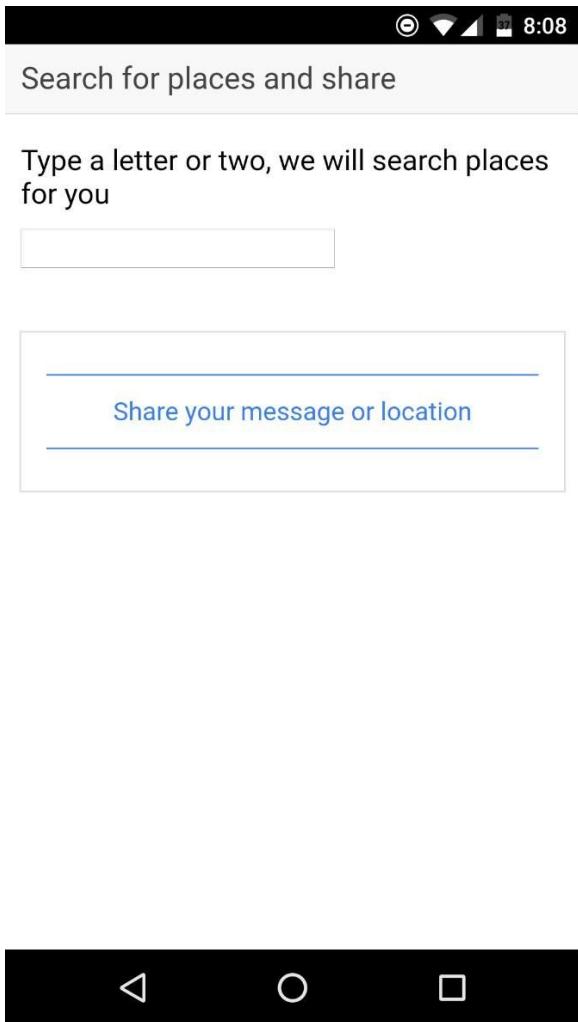
Architecture Diagram:

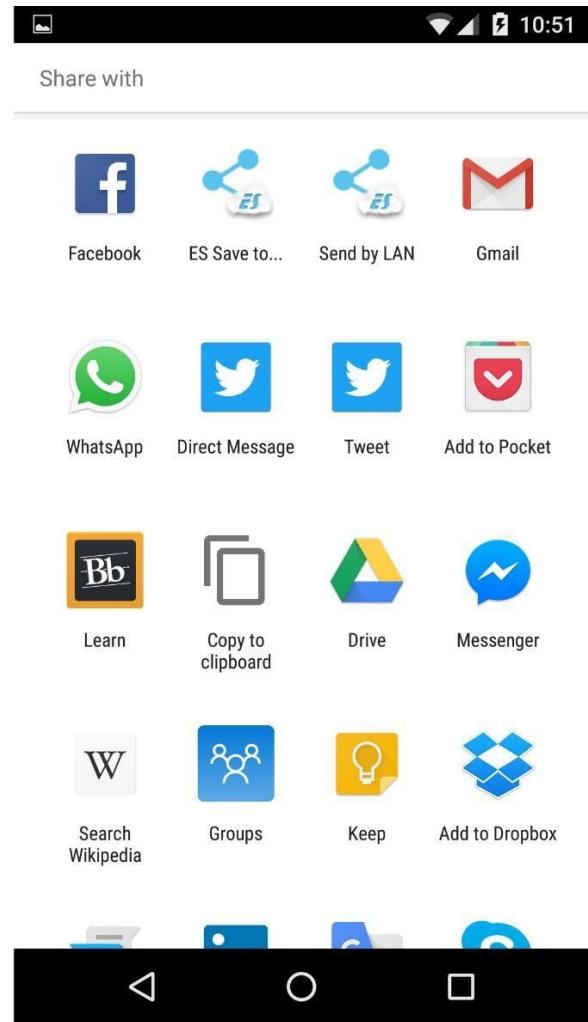
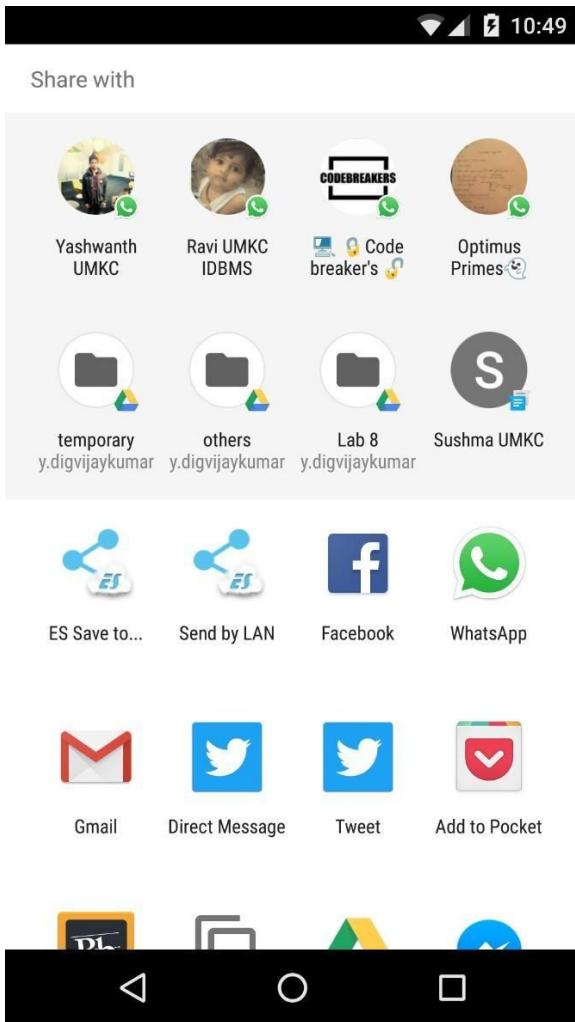


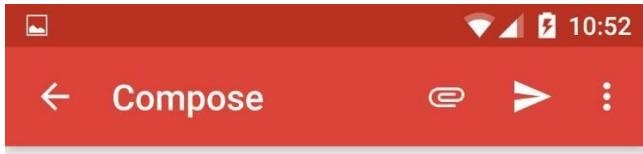
## Results:











From y.digvijaykumar@gmail.com

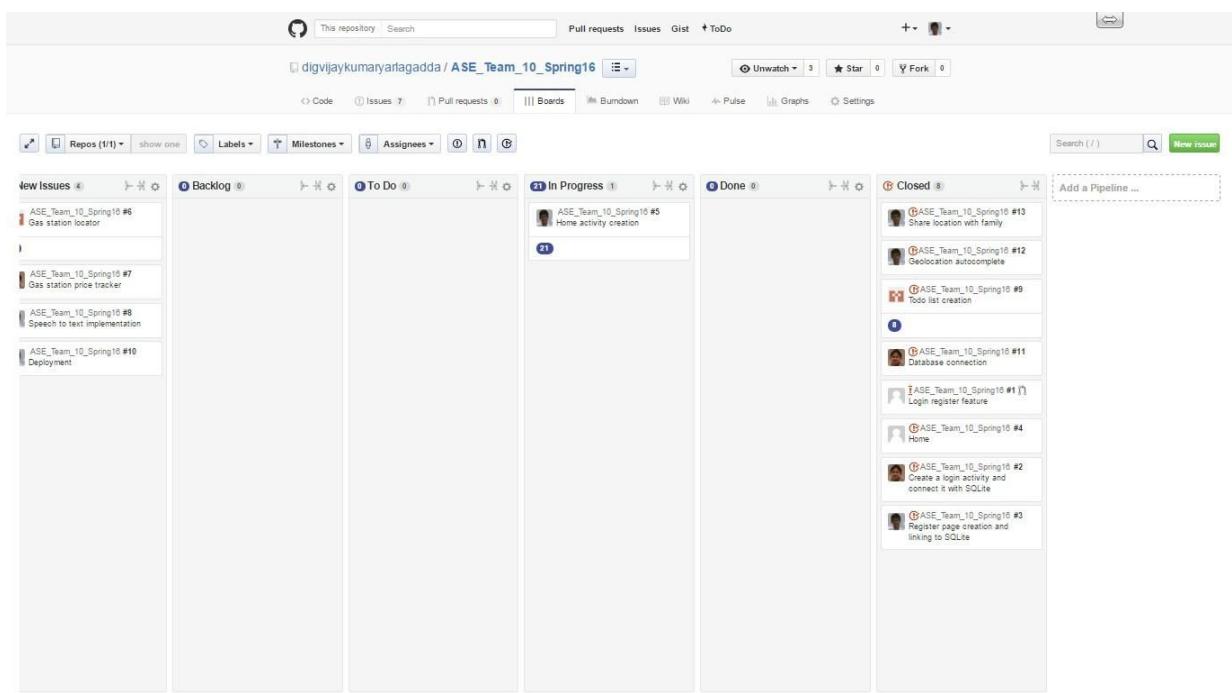
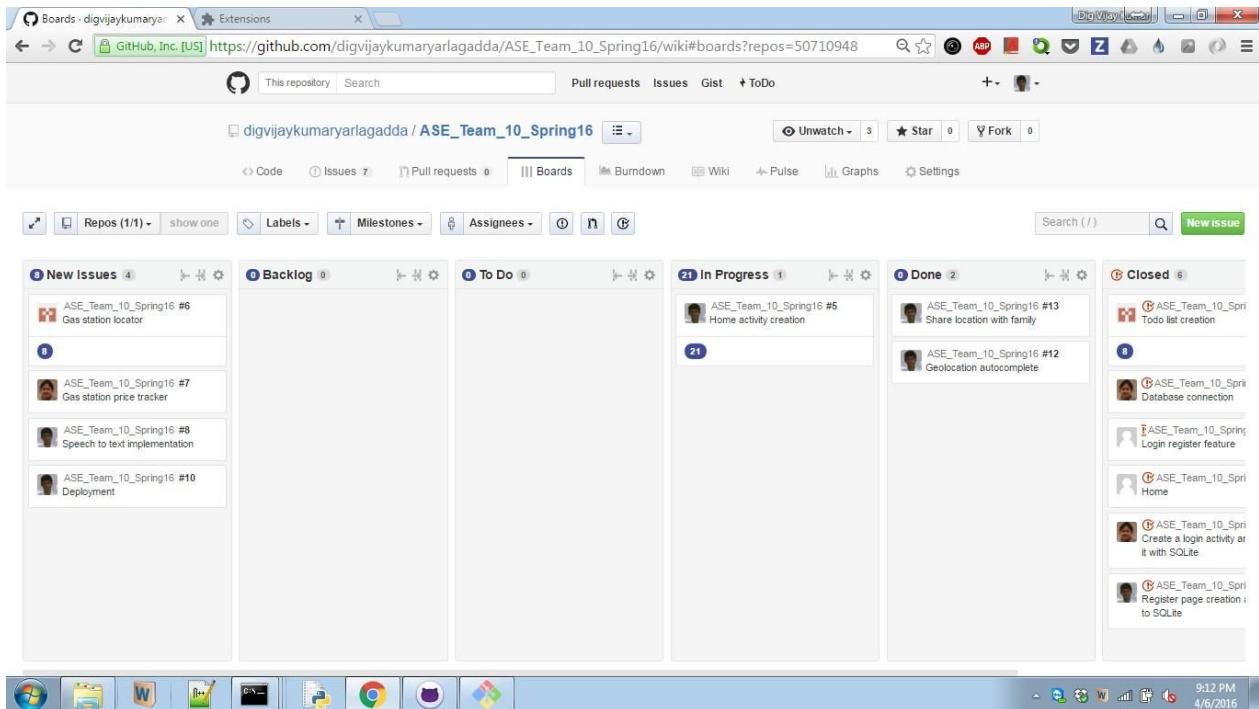
To  Digvijay kumar yarlagadda ▾

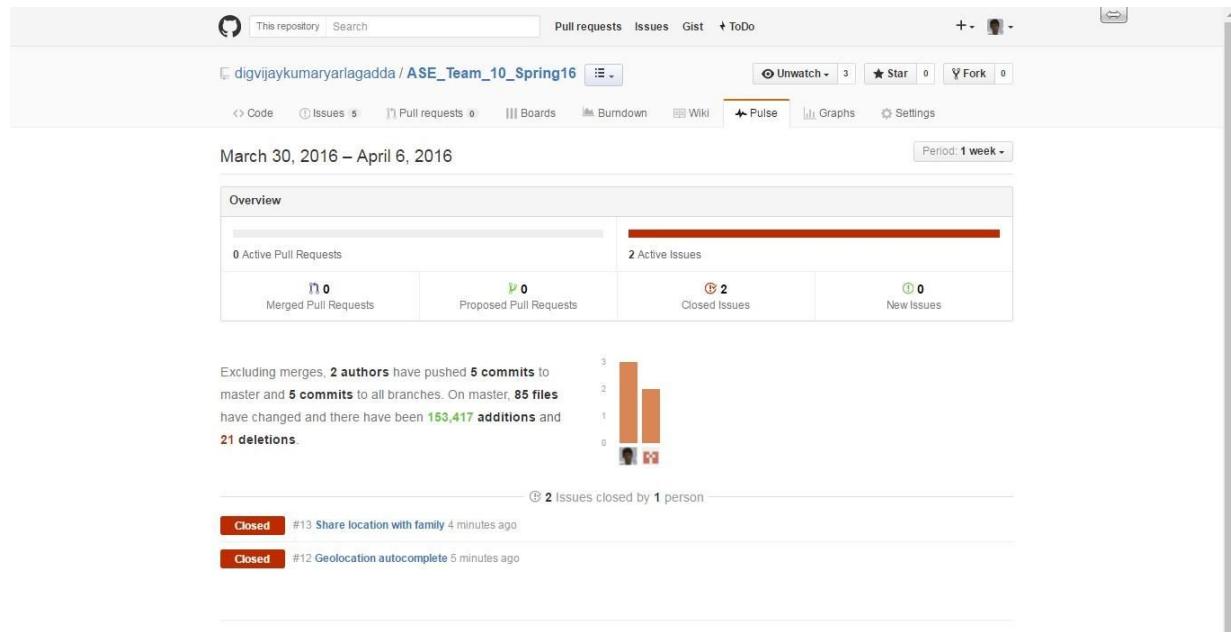
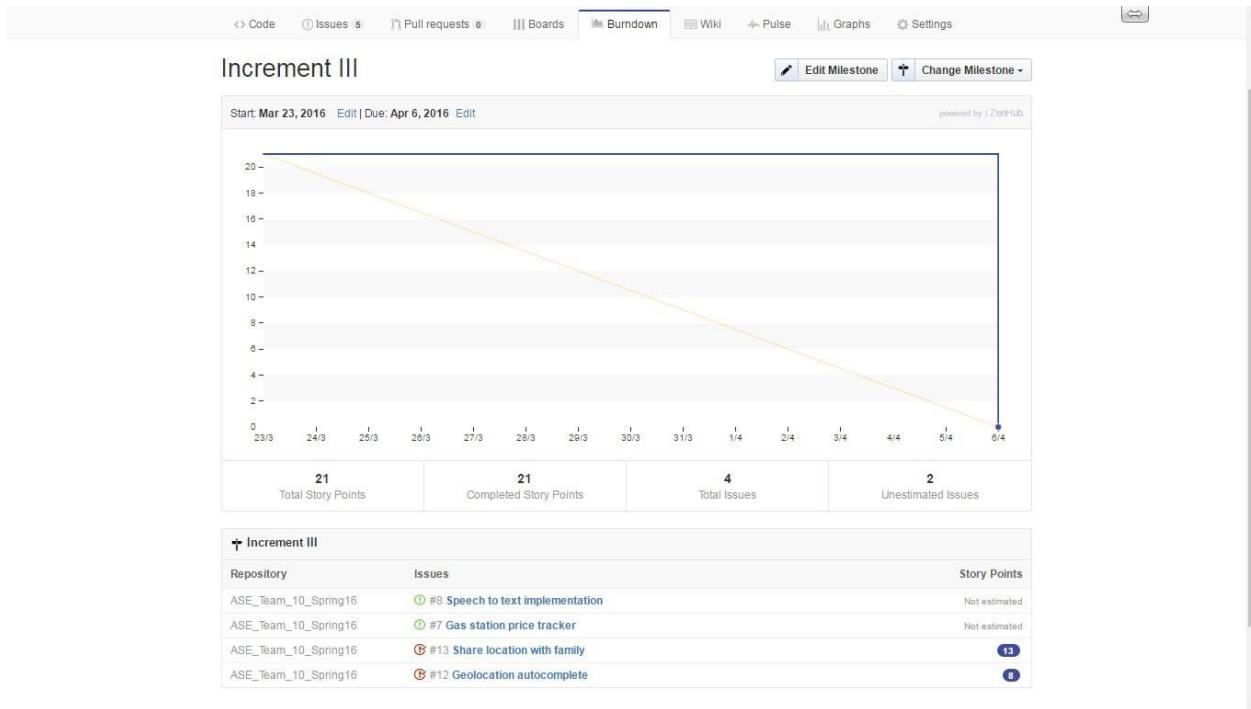
A Subject

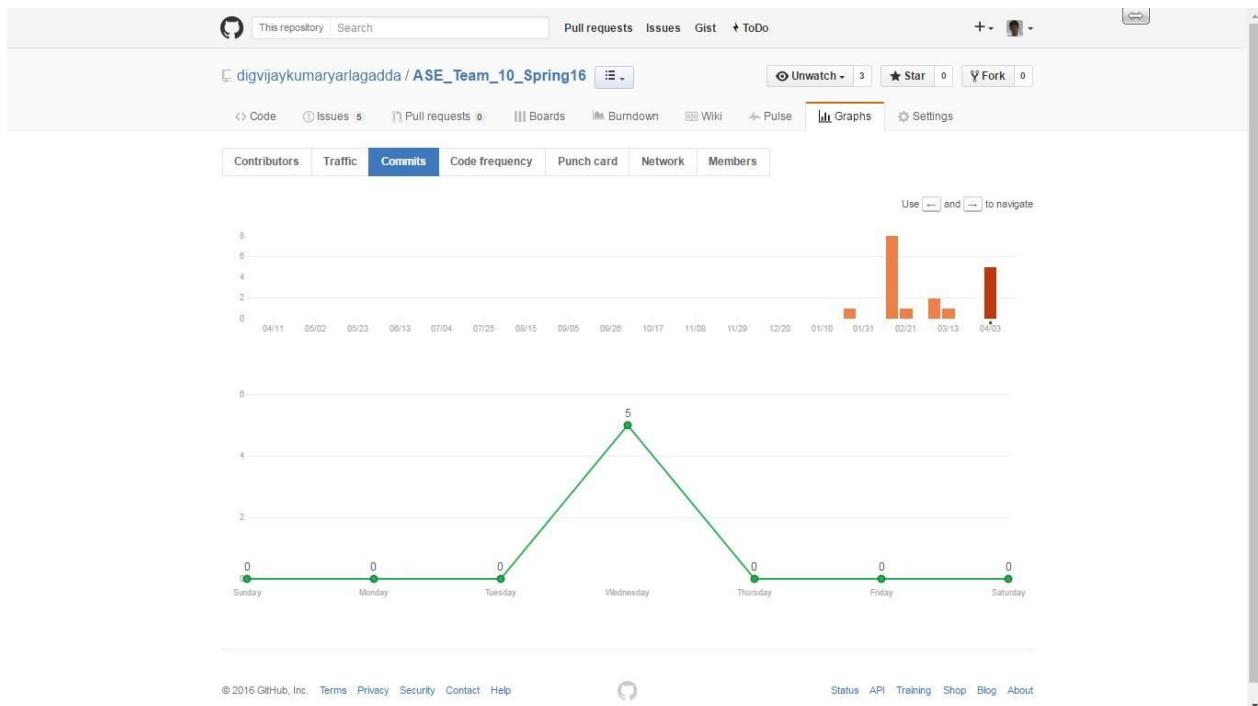
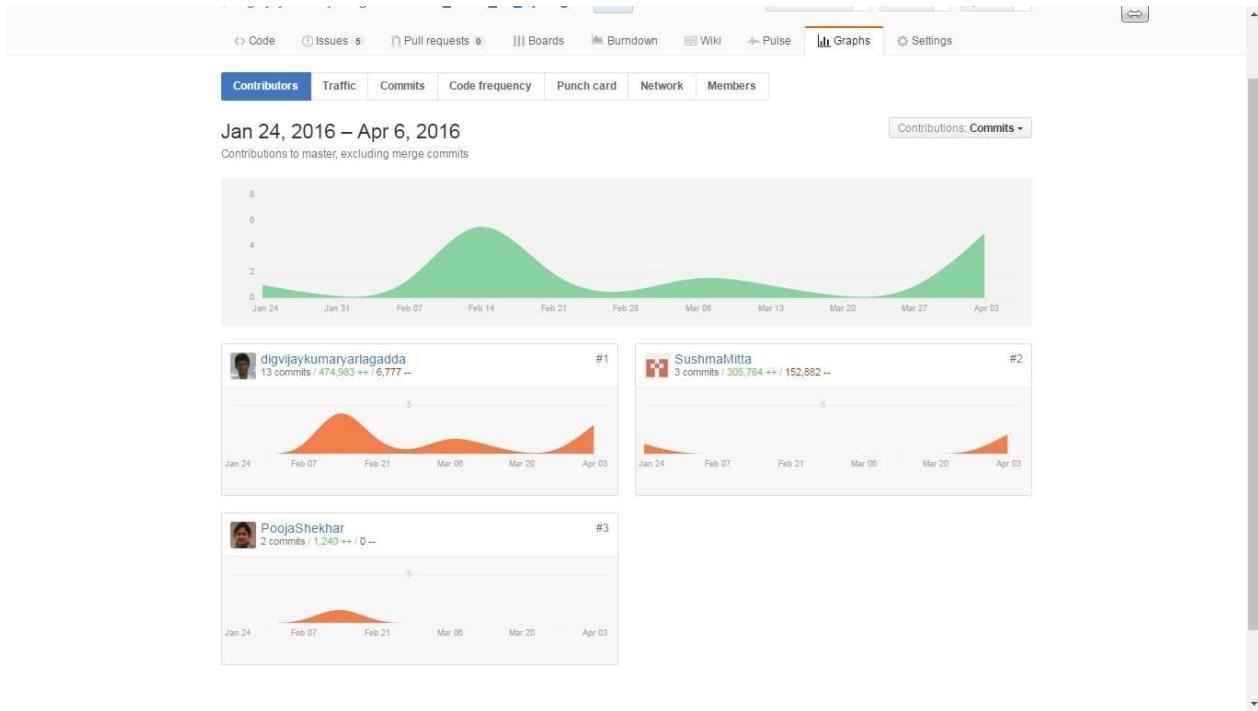
Here is my location. Pick me.



## Project management:







Project GitHub repository:

[https://github.com/digvijaykumaryarlagadda/ASE\\_Team\\_10\\_Spring16](https://github.com/digvijaykumaryarlagadda/ASE_Team_10_Spring16)

Team members contribution:

Pooja Shekhar (25%) – User interface design and supermarket API  
Dig Vijay Kumar Yarlagadda (25%) – Google Places API Autocomplete and search features, Sharing feature  
Sushma Mitta (25%) – Task creation, Push notifications and MongoDB storage  
Chandra Sekhar Janyavula (25%) – Project documentation

Bibliography:

- [1]. <http://ionicframework.com/>
- [2]. <https://github.com/driftyco/ionic-todo>
- [3]. <https://github.com/EddyVerbruggen/SocialSharing-PhoneGap-Plugin>
- [4]. <http://ngcordova.com/docs/plugins/socialSharing/>
- [5]. <https://github.com/petehouston/demo-ionic-social-sharing>
- [6]. <https://developers.google.com/places/>

# Fourth Increment Report

## Introduction:

Saver 'n Saviour is an android application which can manage everyday to do lists with time and location based reminders. After logging in to the application, users can add, edit, delete items in their to do lists and create reminders on them. Whenever the user reaches nearby a grocery store, they will be reminded of the items in their to do list.

In this increment, we have developed location based reminders: users are reminded whenever they reach nearby a grocery store.

## Objectives:

The primary objectives are:

- Create task list
- Establish MongoDB connection to store user data
- Create location based reminders

## Features:

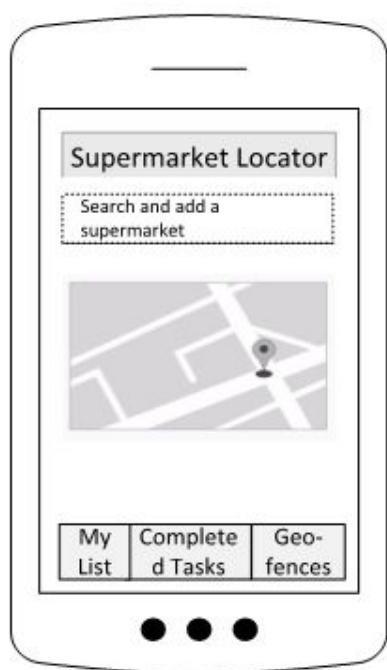
The salient features of the application are:

- Register and Login features
- Create to do lists for different categories and modify, delete or check mark tasks
- Store and synchronize all the user data with database
- Location based reminders

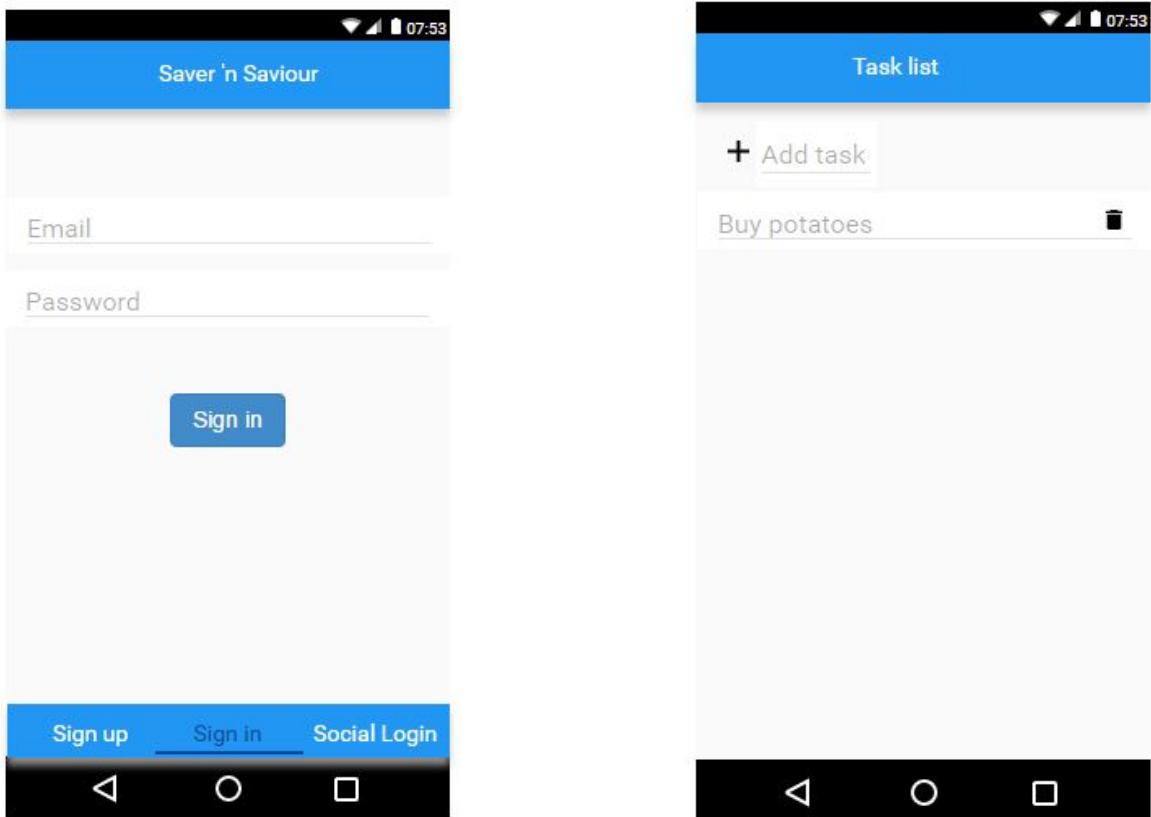
## Existing Services/API:

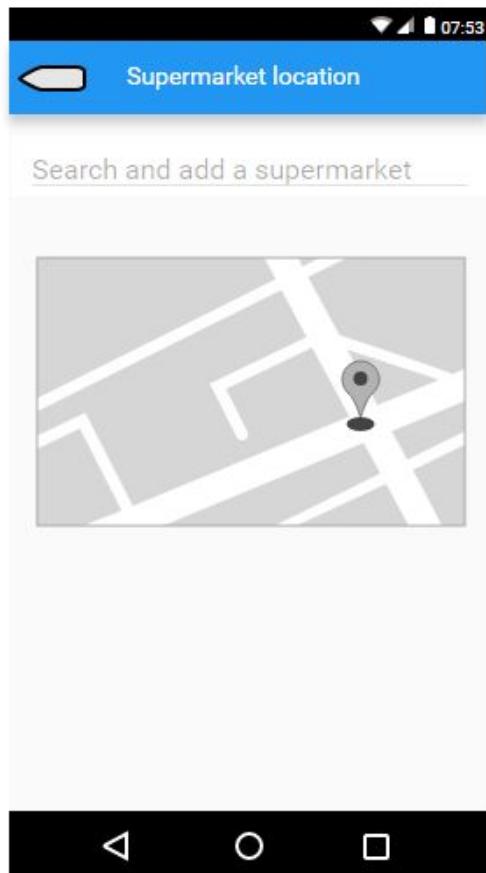
- Mongo Lab
- Google Places API
- Walmart API
- OpenMaps API
- Twilio API

Wireframes:

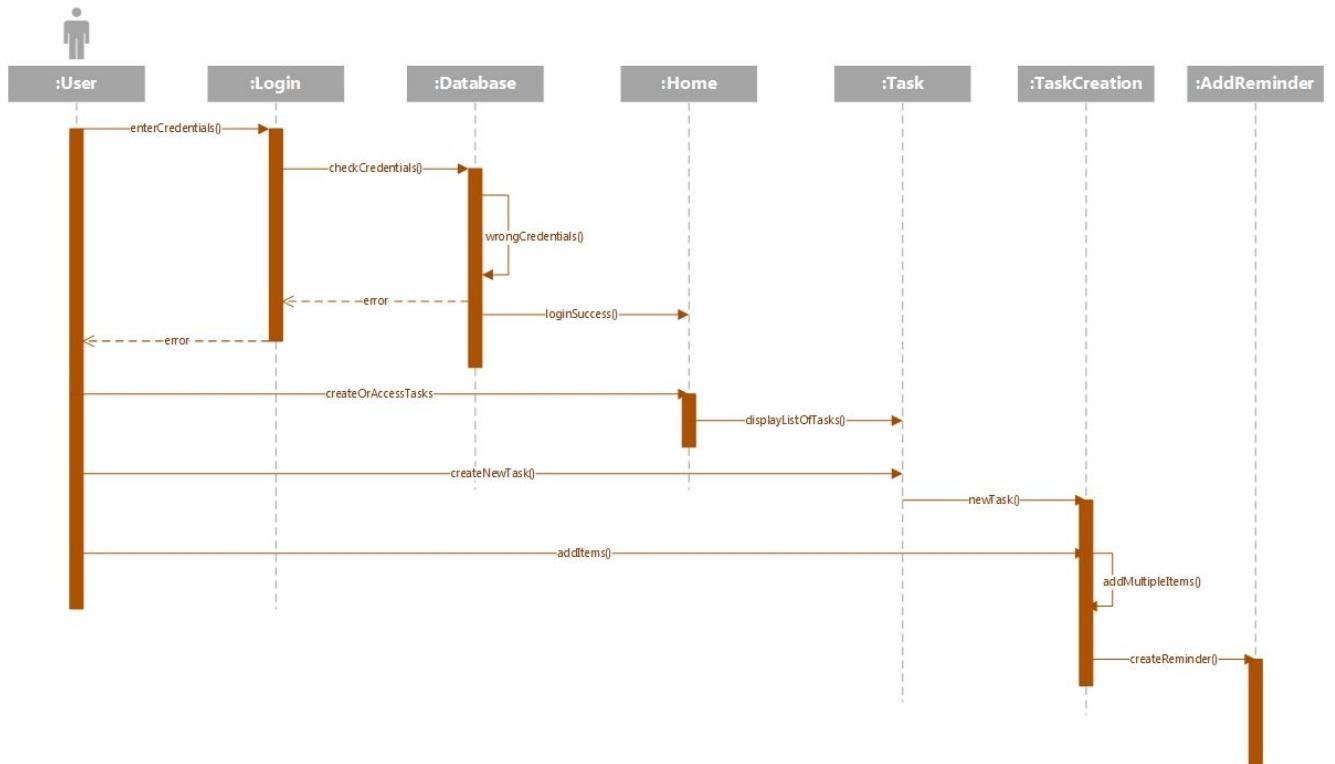


Mockups:

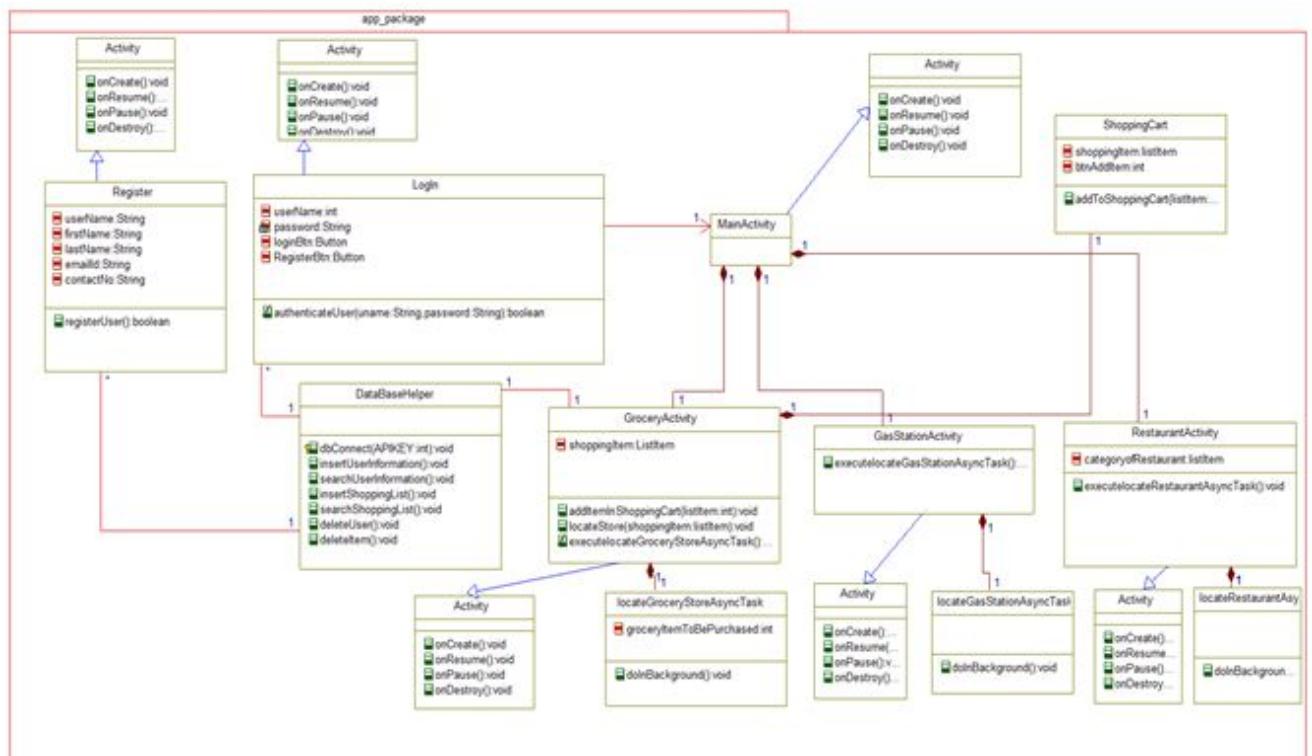




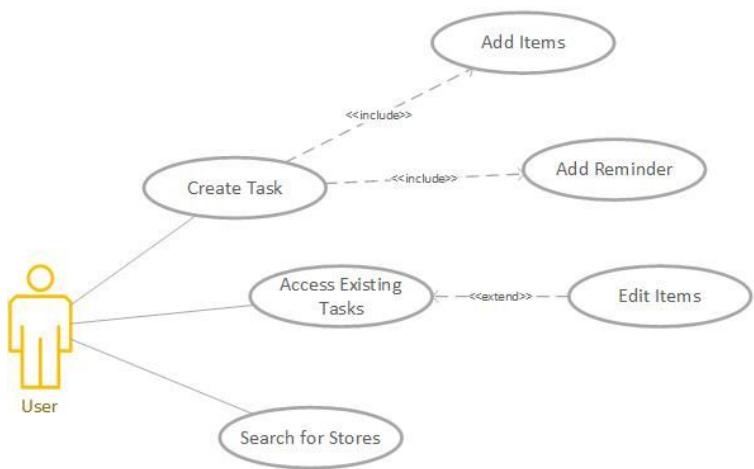
Sequence diagram:



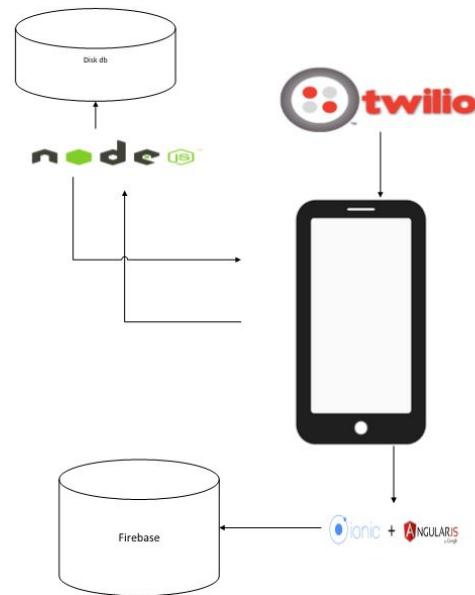
Class diagram:



Use case diagram:



Architecture diagram:

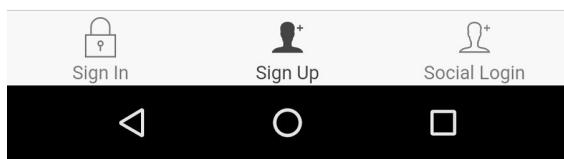
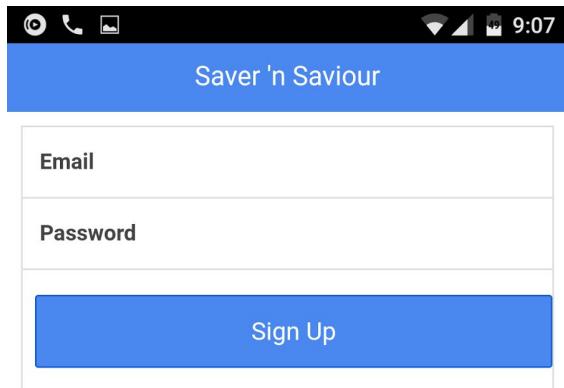


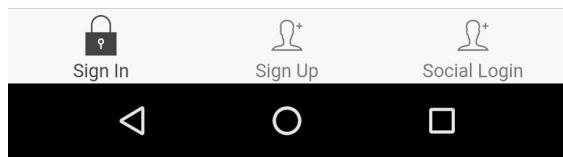
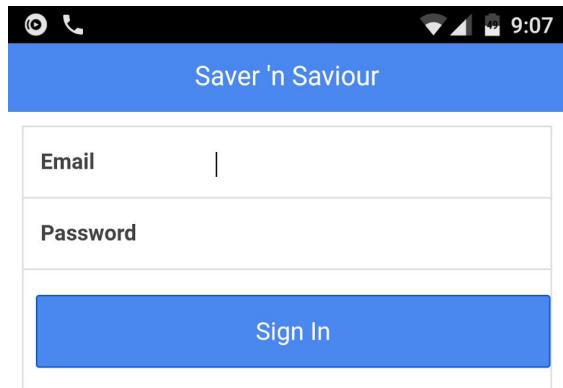
Implementation:

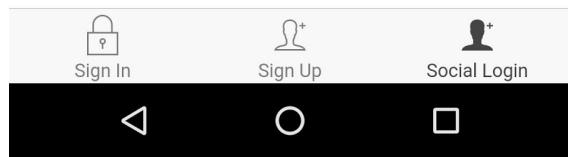
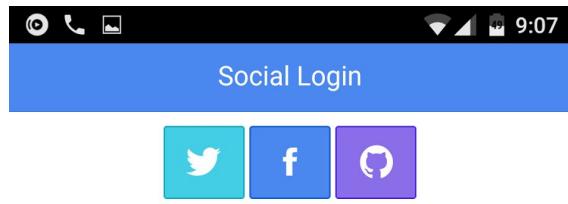
This android application has been implemented using ionic framework which is based on Angular.js. The app data has been hosted on a Firebase database.

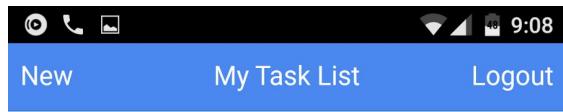
Deployment:

The application has been deployed to Nexus 5 running Android 6.0. The below screenshots shows the login and register features, task creation and location based reminders.

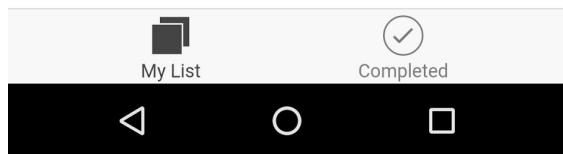


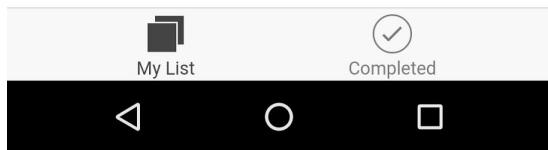
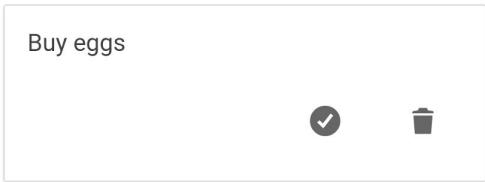
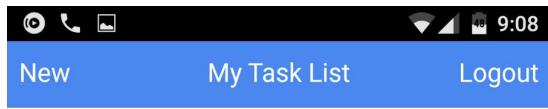


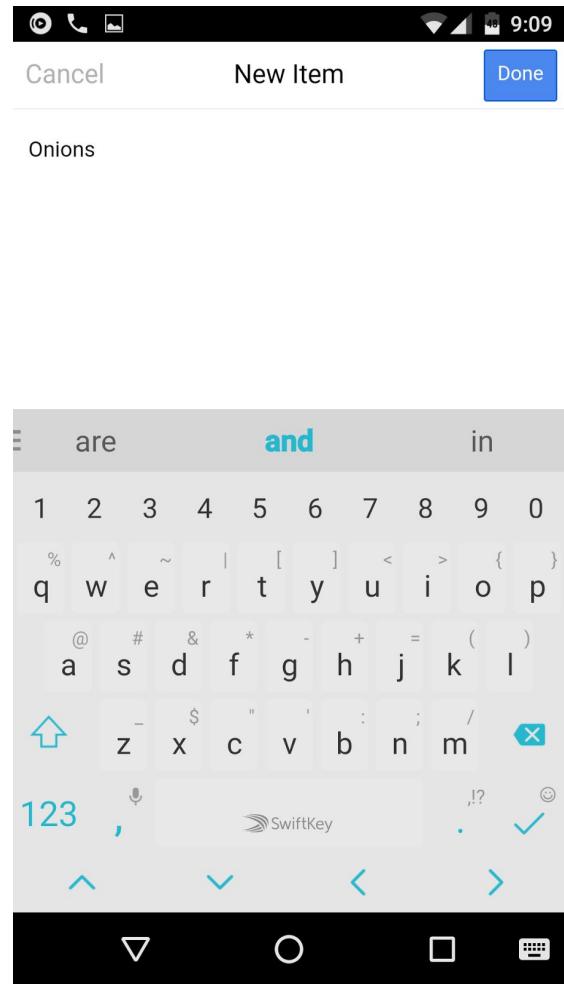


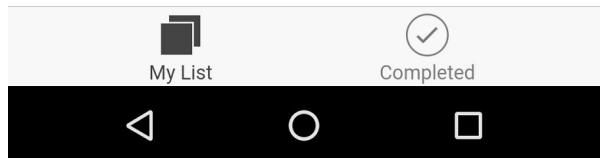
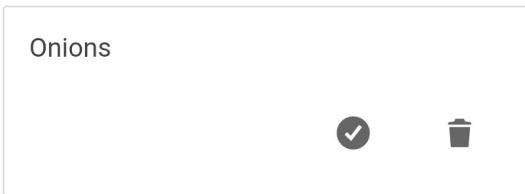
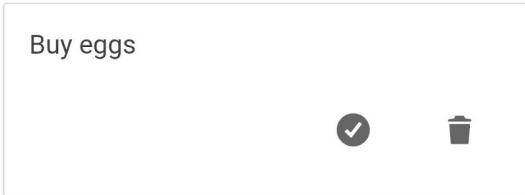
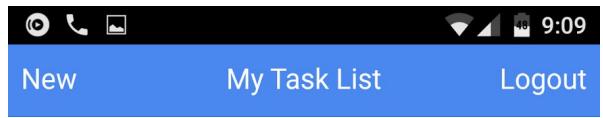


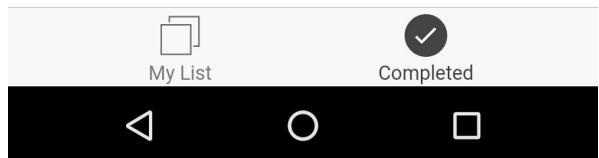
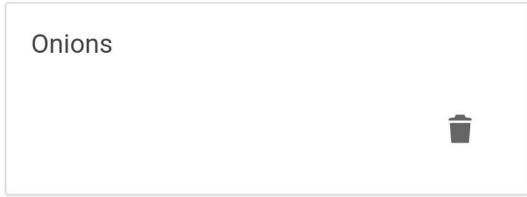
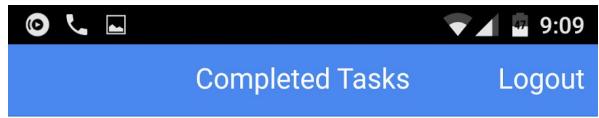
No Items in your bucket List. Click [Here](#) and create one

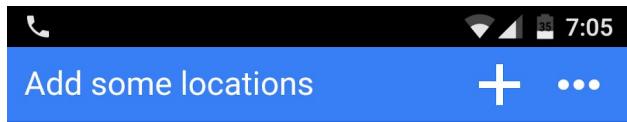


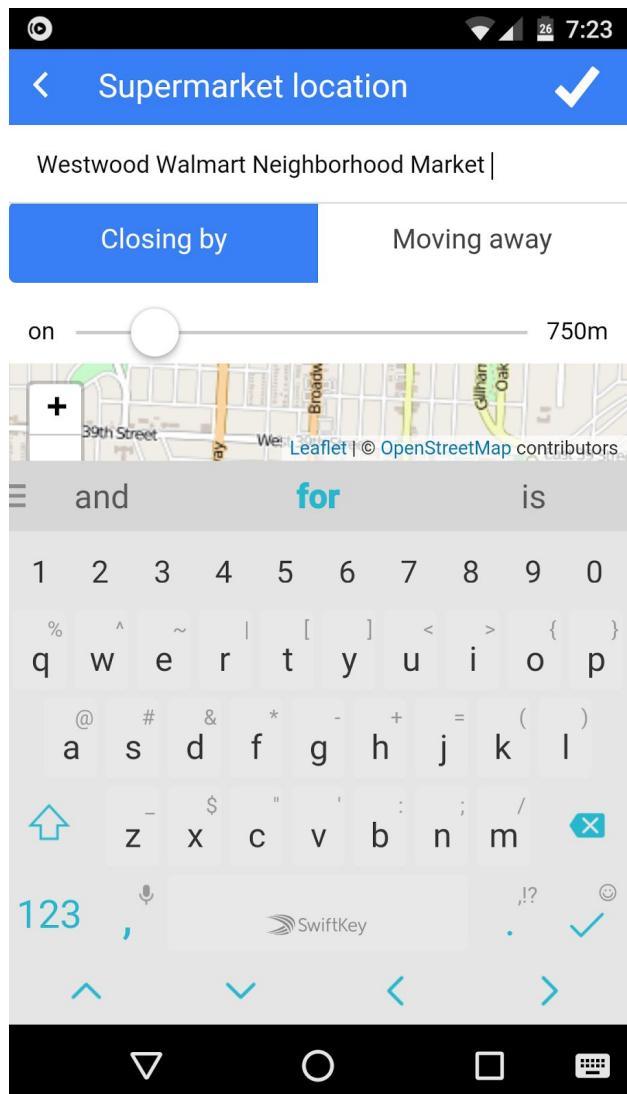


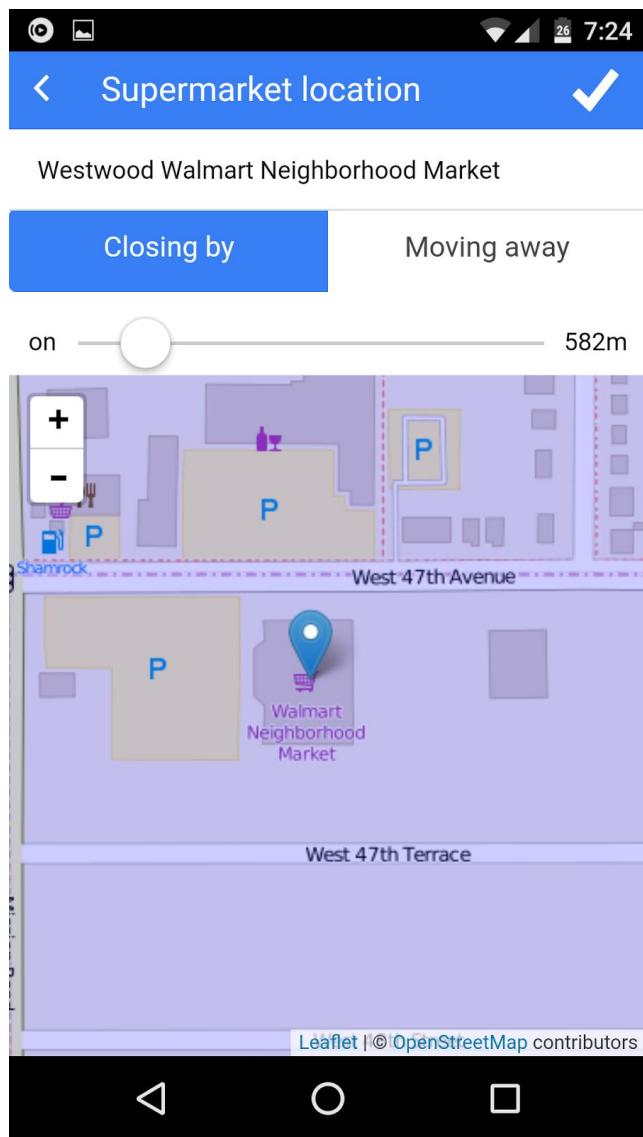


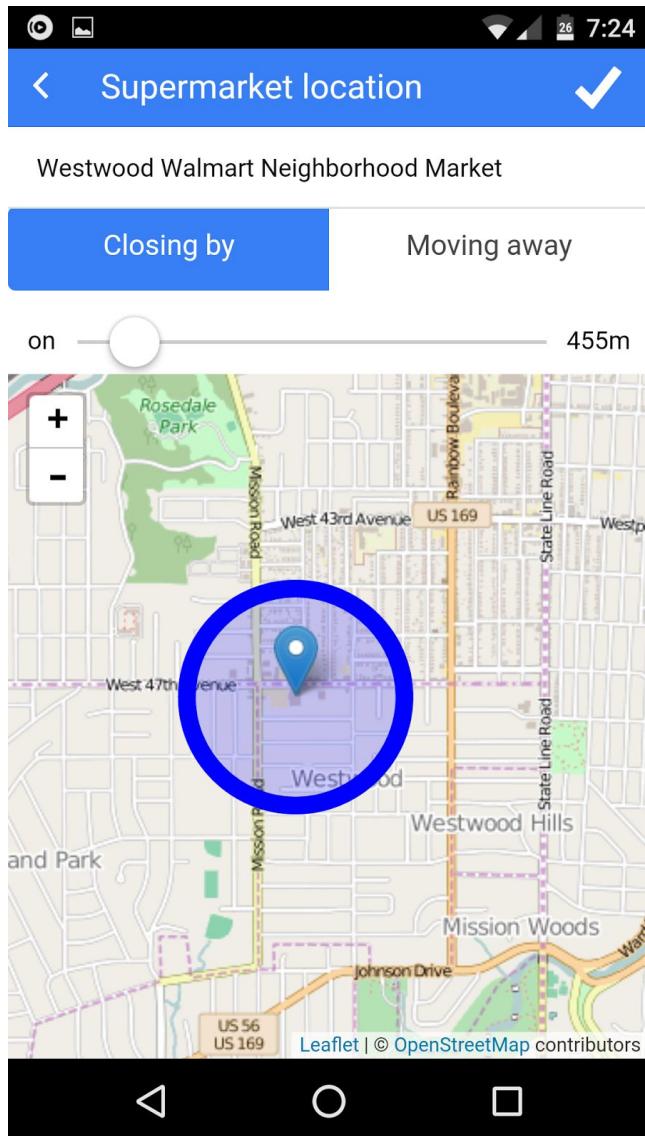


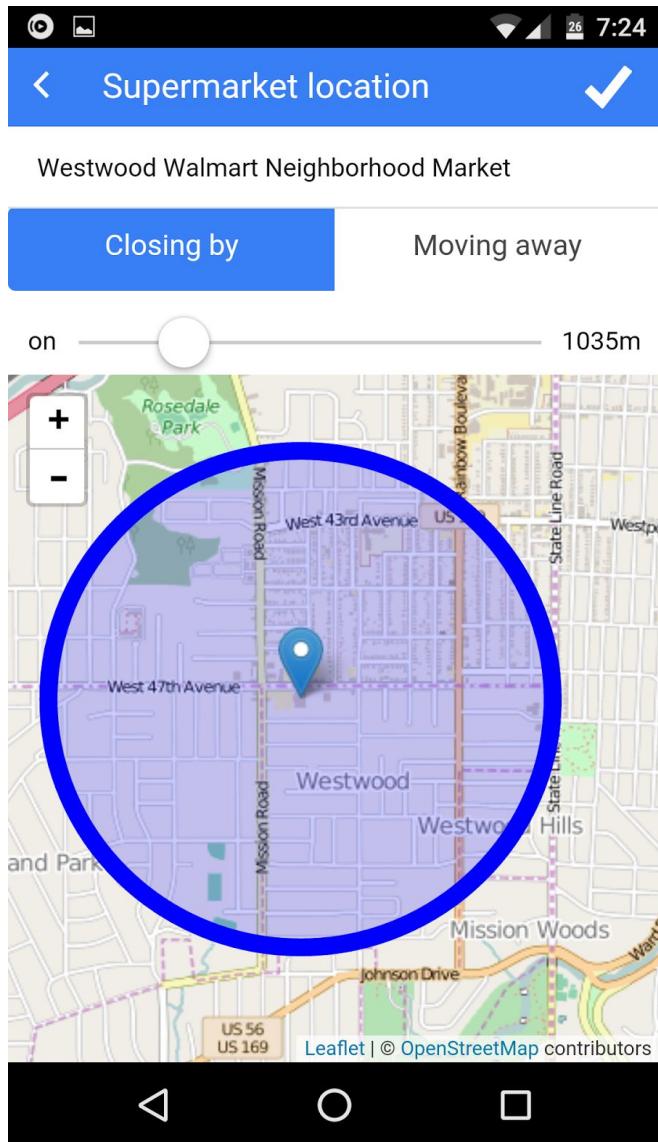


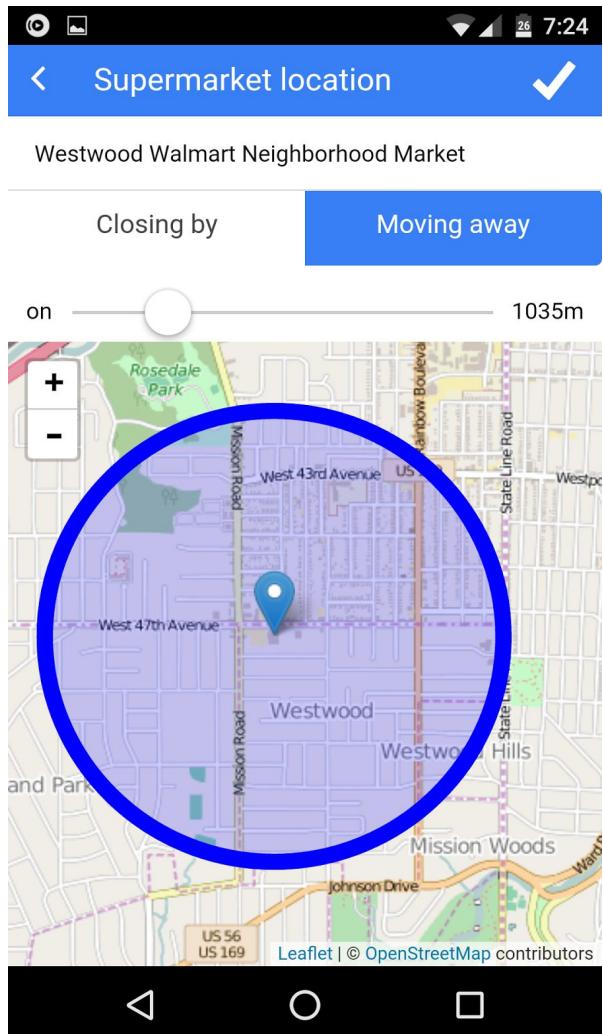


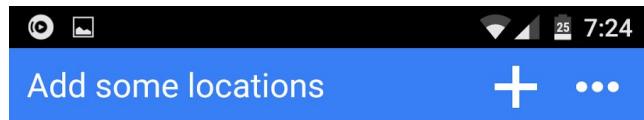


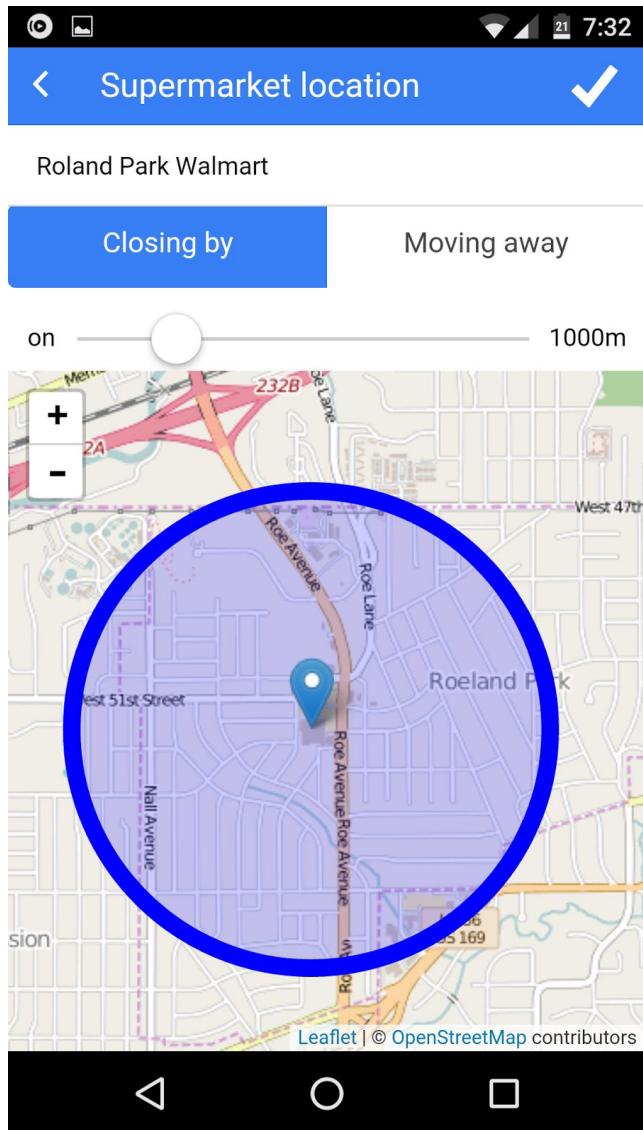


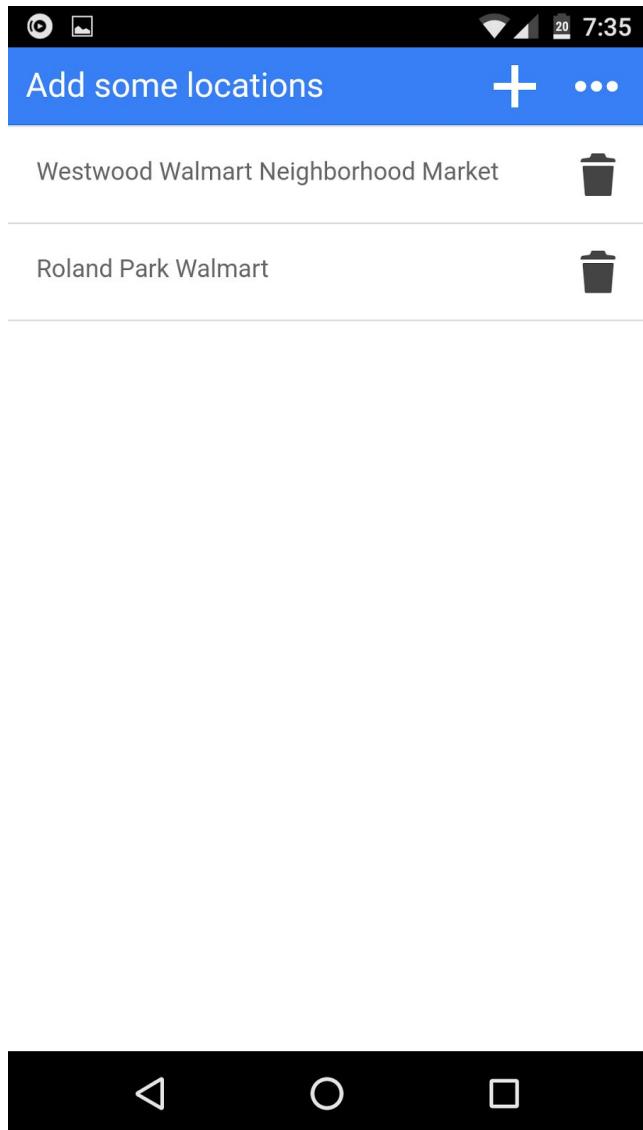








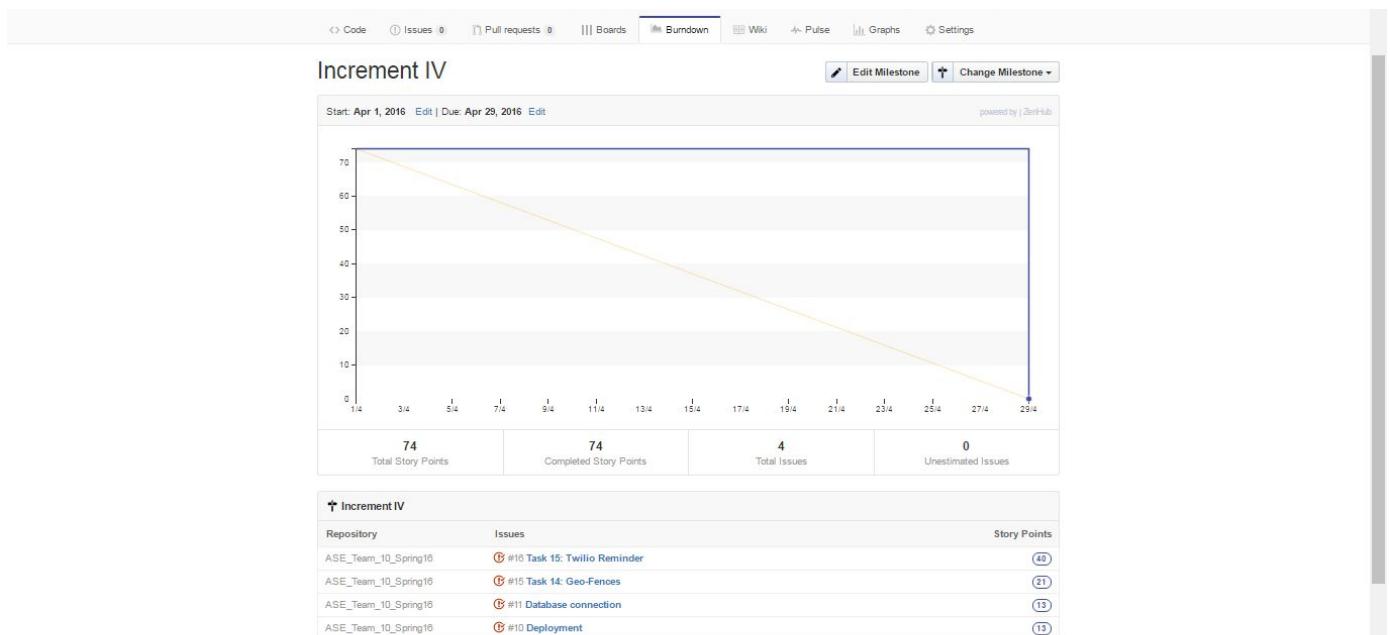




## Project management:

This screenshot shows a GitHub project board for the repository `digvijaykumaryalagadda / ASE_Team_10_Spring16`. The board has five columns: **New Issues**, **Backlog**, **To Do**, **In Progress**, and **Done**. The **Done** column contains 145 closed issues. A sidebar on the right provides a detailed view of these closed issues, including their titles and descriptions.

Issue	Description
ASE_Team_10_Spring16 #16	Task 15: Twilio Reminder
ASE_Team_10_Spring16 #15	Task 14: Geo-Fences
ASE_Team_10_Spring16 #14	geo-fence
ASE_Team_10_Spring16 #10	Deployment
ASE_Team_10_Spring16 #5	Home activity creation
ASE_Team_10_Spring16 #6	Gas station locator
ASE_Team_10_Spring16 #8	Speech to text implementation



**Overview**

1 Active Pull Request		7 Active Issues	
Merged 1	Proposed 0	Closed 7	New 0

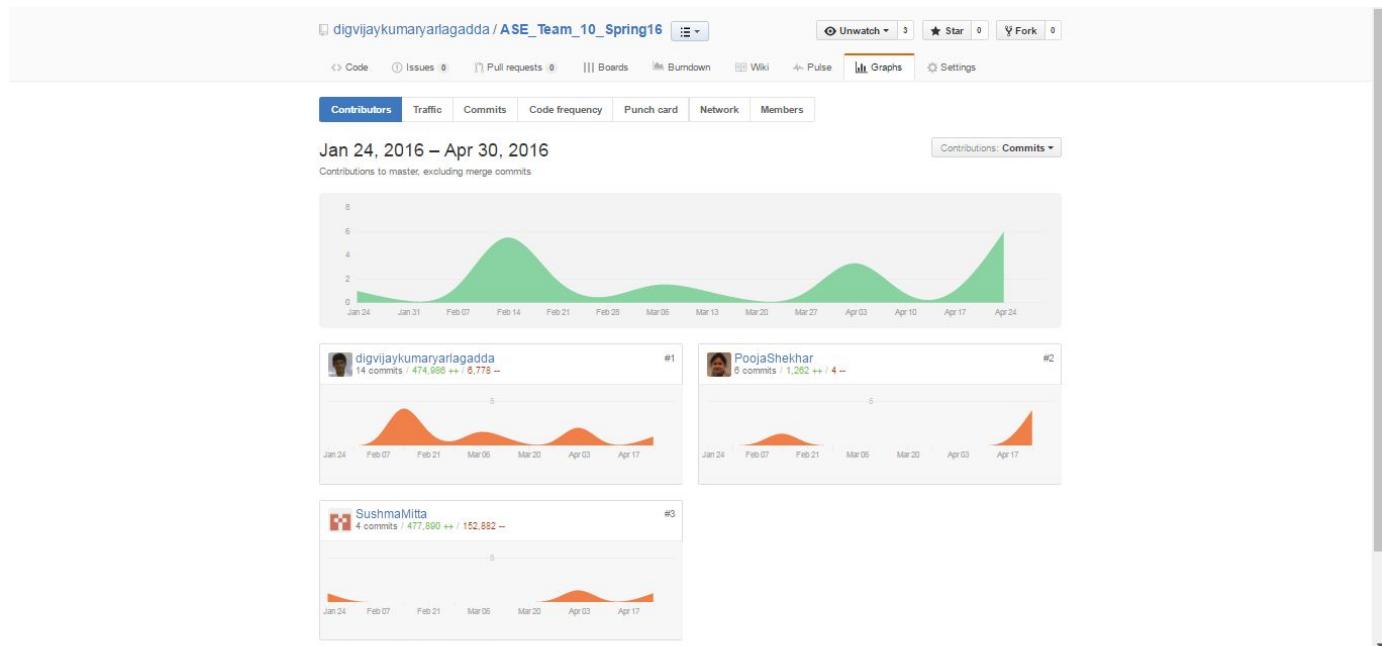
Excluding merges, 3 authors have pushed 6 commits to master and 7 commits to all branches. On master, 142 files have changed and there have been 172,147 additions and 1 deletions.

1 Pull request merged by 1 person

Merged #14 geo-fence 4 hours ago

7 Issues closed by 2 people

Closed #16 Task 15: Twilio Reminder 3 hours ago  
Closed #15 Task 14: Geo-Fences 3 hours ago  
Closed #10 Deployment 11 hours ago  
Closed #5 Home activity creation 11 hours ago  
Closed #6 Gas station locator 11 hours ago  
Closed #8 Speech to text implementation 11 hours ago  
Closed #7 Gas station price tracker 11 hours ago



Project GitHub repository:

[https://github.com/digvijaykumaryarlagadda/ASE\\_Team\\_10\\_Spring16](https://github.com/digvijaykumaryarlagadda/ASE_Team_10_Spring16)

Team members contribution:

Pooja Shekhar (25%) – Login and register page creation, DB connection

Dig Vijay Kumar Yarlagadda (25%) – Location based reminders (geofences), Task creation

Sushma Mitta (25%) – Task creation, DB storage

Chandra Sekhar Janyavula (25%) – Project documentation

Bibliography:

[1]. <http://ionicframework.com/>

[2]. <https://github.com/driftyco/ionic-todo>

[3]. <https://github.com/cowbell/cordova-plugin-geofence>

[4]. <https://github.com/cowbell/ionic-geofence>

[5]. <https://developer.walmartlabs.com/>

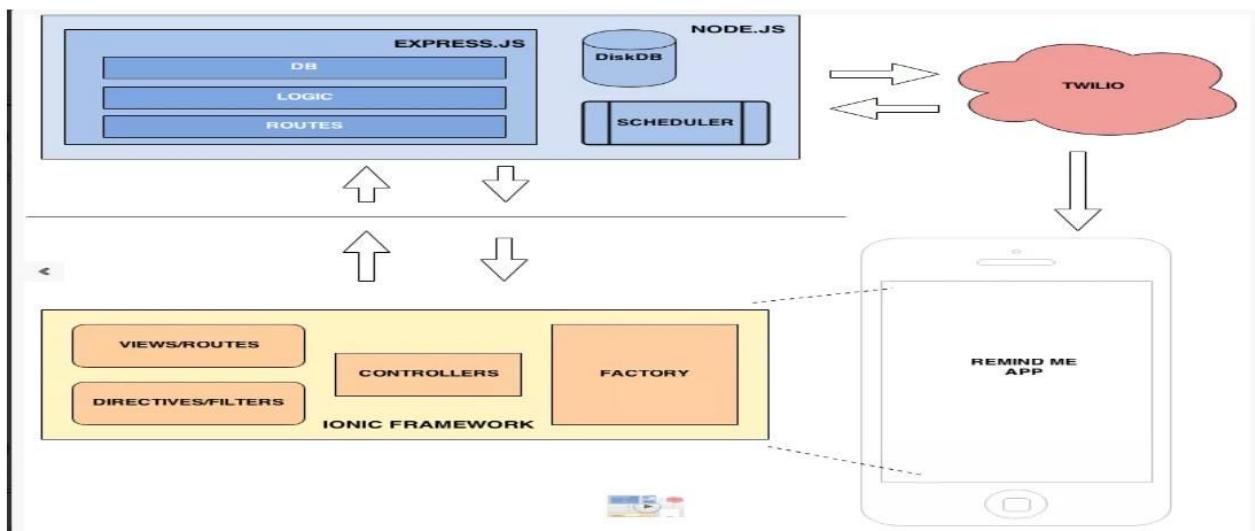
[6]. <https://www.firebaseio.com/>

[7]. <http://www.sitepoint.com/creating-firebase-powered-end-end-ionic-application/>

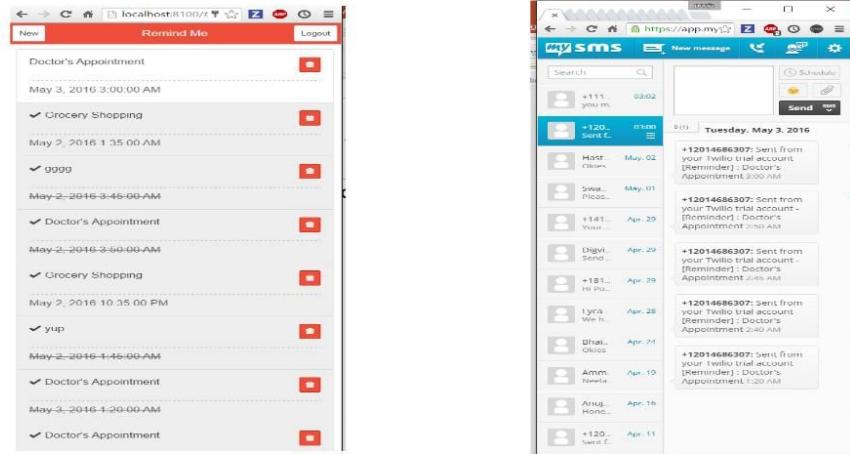
[8]. <http://thejackalofjavascript.com/ionic-twilio-and-node-scheduler-part-1/>

# Presentation Slides

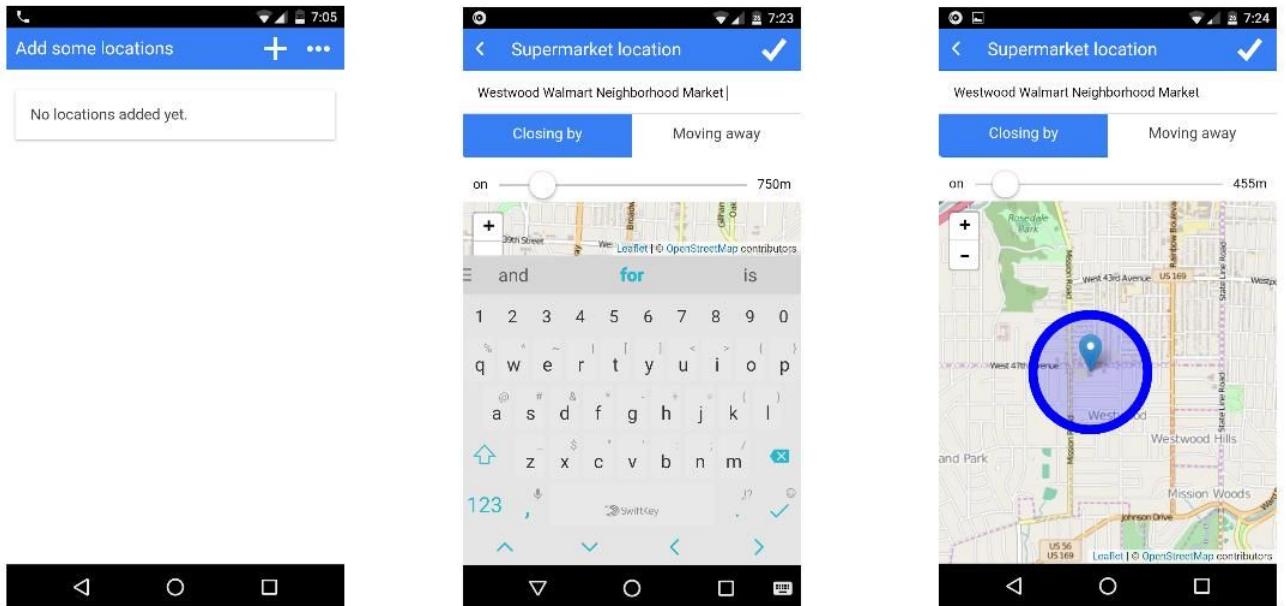
## Architecture



## Reminder using Twilio



## Push Notification using Geofence





## Features

- Saver 'n Savior is a hybrid application which can manage everyday to-do lists with time and location based reminders.
- By using this app, a user can register, login and then create a reminder. While creating the reminder, the user has an option to schedule a Call or a SMS to remind him/her of the task at the scheduled time. Here we used cloud based communication API, Twilio to make a call or send an SMS.
- There is also an option of creating geofences, using cordova geofence plugin, so that when the user reaches a geofence area, then they will be reminded of the items in their to-do list.

**Github URL:**

[https://github.com/SCE-UMKC/ASESP16\\_Saver\\_n\\_Savior\\_Team10](https://github.com/SCE-UMKC/ASESP16_Saver_n_Savior_Team10)

**Youtube URL:**

<https://www.youtube.com/watch?v=zQovKvqpsrk&feature=youtu.be>