

# GO EASY

CS5551

Instructor: Dr. Yuggyung Lee

TA: Malathy Krishnan

Team Members:

Pavan Kumar Bollaram (class id-7)

Vara Prasad Reddy (class id-22)

Preetham Kumar (class id – 11)

Lakshmi Priyanka (class id- 49 )

## **Table of Contents**

<b><u>S.no</u></b>	<b><u>Description</u></b>	<b><u>Page No</u></b>
<b>1.</b>	<b>Project Deployment</b>	<b>3</b>
<b>2.</b>	<b>User Manual</b>	<b>4</b>
<b>3.</b>	<b>Project Management</b>	<b>8</b>
<b>4.</b>	<b>Project Proposal</b>	<b>9</b>
<b>5.</b>	<b>Project Plan</b>	<b>9</b>
<b>6.</b>	<b>Project Increment 1</b>	<b>21</b>
<b>7.</b>	<b>Project Increment 2</b>	<b>36</b>
<b>8.</b>	<b>Project Increment 3</b>	<b>51</b>
<b>9.</b>	<b>Project Increment 4</b>	<b>72</b>
<b>10</b>	<b>Presentation Materials and Project Evaluation</b>	<b>102</b>

## PROJECT DEPLOYMENT

The project description and methodologies are described briefly and are captured as a video. The video has been uploaded to the YouTube which can be viewed using the following link:

<https://www.youtube.com/watch?v=DtsX-dtiAIE>

Our project has been planned using the Scrum do which is an agile tool and the link is given below as follows:

<https://www.scrumdo.com/projects/project/ase-project8/summary>

Source code is deployed at this location in GIT:

<https://github.com/pavankumar-b/ASEspringSem/Increment4>

Final report is published to GIT location mentioned below

<https://github.com/pavankumar-b/ASEspringSem/FinalReport>

# USER MANUAL

## System Requirement:

Operating System: Android

Android Version Support: froyo to kitkat

Memory Required: 256MB

Internet Connection: Yes

## Devices: Android devices

- Bluetooth 4.0 compatible android device
- Android 4.3 or above.

## Goal:

The main objective of the Application is to be handy to the new students of the UMKC. It provides the entire details of the university. It gives the details of the events going in the university during the current month with the location and the timing details. Coming to the additional feature. It provides all the live events going at a particular City.

## Guide Lines:

1. Please download our application “GO EASY” and install in your mobile device.
2. By clicking or tapping the icon of our application on user screen, the application launches.
3. You will be directed to screen for login and for registration.
4. After entering the details in the form submit it by clicking the login button to enter the application.
5. You will notice a button go to enter to the tab view screen
6. In tab view screen you can find three tabs to select, one for finding events in across city, one for accessing google maps for transportation and one for the enter into the university screen.
7. If you want details of the university click university tab, it segregates university info into different sections - about University, Academics, Sports, Map and Events
8. You can click or tap the desired section and get the details accordingly.

## Description:

Better Answers for Bigger questions. Experience to the immediate and one stop access to all things about a university.

Navigation and Transportation:

Included in University Map and Transportation section.

- Maps
- Bus timings and routes.
- Navigates to event location.

Academics Related:

- List of courses for under graduate and graduate.
- Contact details of different schools.
- Quick access to information regarding university.

Sports:

- Quick info of the sports held.
- Transportation to the sports event.

About University:

- Contact information
- University details

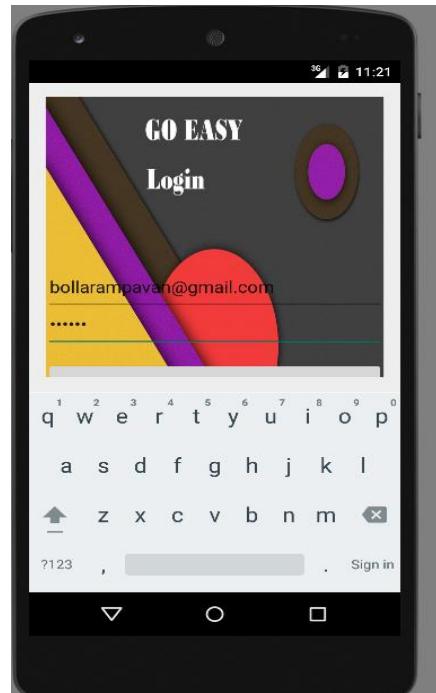
What's New:

- Voice guided navigation
- Direct Access to events
- Quick access to Campus Details.

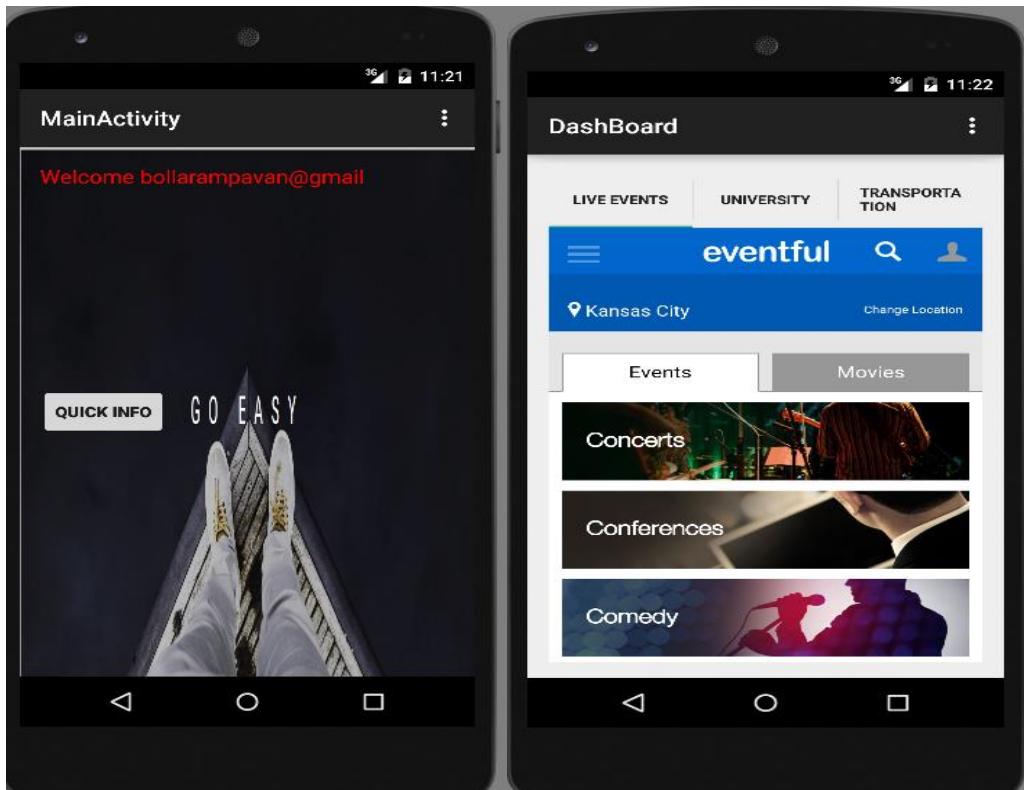
Sample Execution:

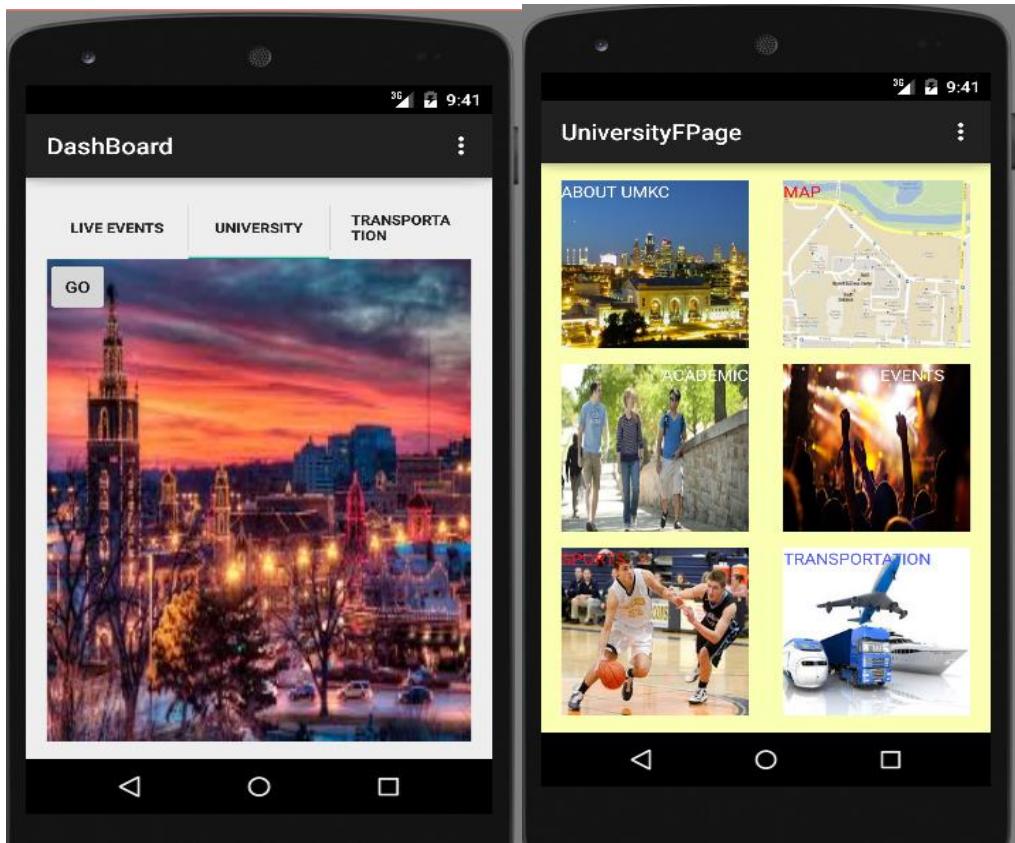
Here included the screen shots of our application.

On run of the application it pops up for user login if new user needs a registration process to carry over.



Below are the screen shots that shows the main features of the application.





#### TroubleShooting:

- If you have any issues while installing, please remove the apk and reload it.

## PROJECT MANAGEMENT

As part of our Project Management, We had used Scrum Do Website for planning our tasks among our team mates.

1. Pavan Kumar,Bollaram
2. Lakshmi Priyanka
3. Preetham kumar ,Danaboina
4. Vara Prasad Reddy.Jaggu

<https://www.scrumdo.com/projects/project/ase-project8/summary>

# PROJECT PROPOSAL

## **Project Goal and Objectives:**

### **Overall Goal:**

Main objective and goal of this project proposal is to create a web application which comes in handy to help the new students joining in universities which are located far from their home towns. This application should provide location based services to end user and should provide options to the end user so that he can opt for services of his interest. Once the user selects the services then application should present the detailed list of services around him and end user can get alerts or voice based interactions about the live events, concerts which are scheduled at that time.

### **Specific Objectives**

User notifications on mobiles and voice based alerting system are the specific objectives we are planning at this juncture and all of this implementation is completely based on location services or user can manually enter the location and get the services too.

### **Significance**

The significance of this application is to provide a comprehensive user friendly details about the events which end user subscribe to and providing timing of events, directions to reach the location where the events are going to be. On the final note we can say it is simple application where one new comer to a place can find the basic required details.

## **Project Background and Related Work**

College guide, NearbyMe and google's attractions are few applications which generalizes useful information at single place by allowing user to save the time. All these applications shows the user nearby tourist attractions, theater lists, food restaurants info based on the location information provided by end user. Our application does the same kind of operations which are cited earlier in these application but the specific services provided are event notifications to user or voice based alerts which are not implemented any of these related applications.

### **Proposed system**

### **Requirement Specification**

Functional requirements:

- User login screen
- Dashboard with available services

- Options to provide the location manually
- Directions to reach the location or event(Navigation using Google API)
- Integrating all the web services at once place(mashup with Tabs on GUI)
- Map end user locations using google API

Non-functional requirements:

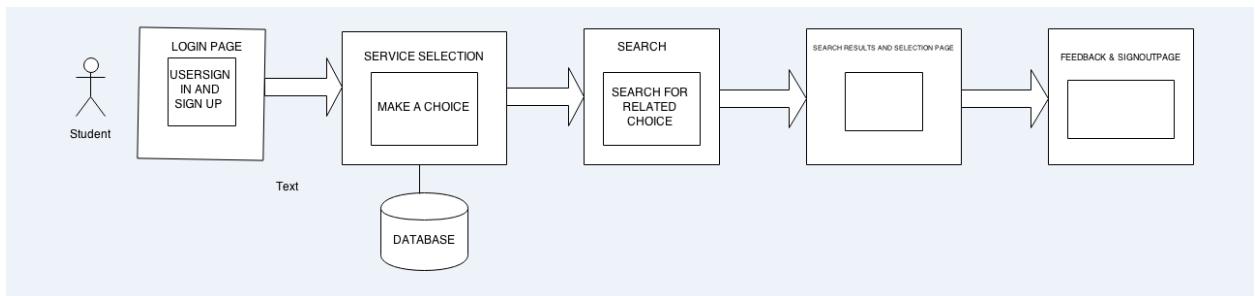
- Feel good GUI
- Quick response for the subscribed services by end user

## Framework Specification

- **Assumptions and Principles**

Assumption is that always a novice student faces complications as he/ she attend a new school like having no idea of initial enrollment procedure, finding lecture building, housing facilities or any recreational events around him would be difficult. Also, we assume that there would a communication overhead when he attends a school in foreign land.

- **System Architecture Diagram**



## System Specification

- (i) **Existing Services:**

Map services from Google Maps

[https://www.google.com/maps?q=google+maps+umkc&rlz=1C1CHWA\\_enUS626US626&ion=1&espv=2&es\\_th=1&bav=on.2,or.r\\_cp.r\\_qf.&bvm=bv.85761416,d.aWw&biw=1366&bih=620&dpr=1&um=1&ie=UTF-8&sa=X&ei=lIzbVNKA8b-yQTz4oDAAg&ved=0CAYQ\\_AUoAQ](https://www.google.com/maps?q=google+maps+umkc&rlz=1C1CHWA_enUS626US626&ion=1&espv=2&es_th=1&bav=on.2,or.r_cp.r_qf.&bvm=bv.85761416,d.aWw&biw=1366&bih=620&dpr=1&um=1&ie=UTF-8&sa=X&ei=lIzbVNKA8b-yQTz4oDAAg&ved=0CAYQ_AUoAQ).

## RSS Feeds

<http://feeds.finance.yahoo.com/rss/2.0/headline?s=>

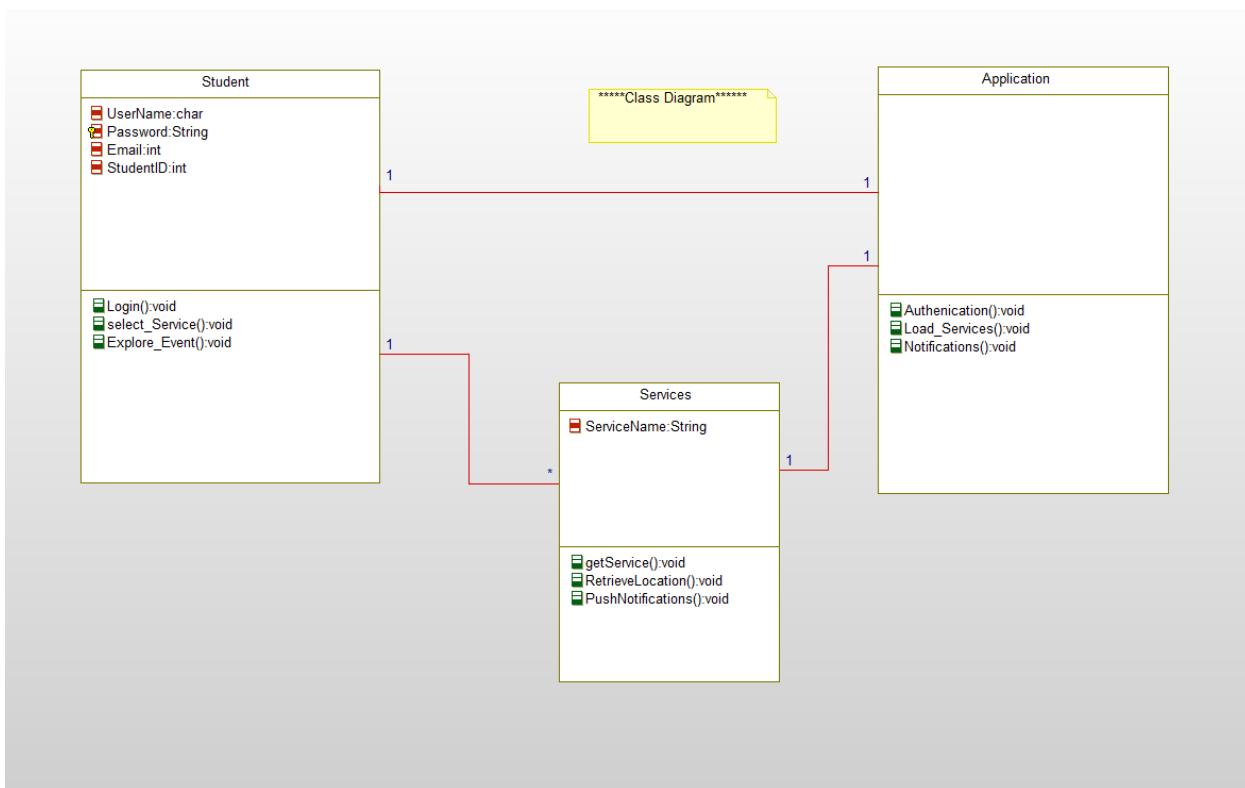
<http://www.kcata.org/>

## New Services to be included:

Mash Up application where one can access at a single time university lecture building activities pop up held at that particular building.

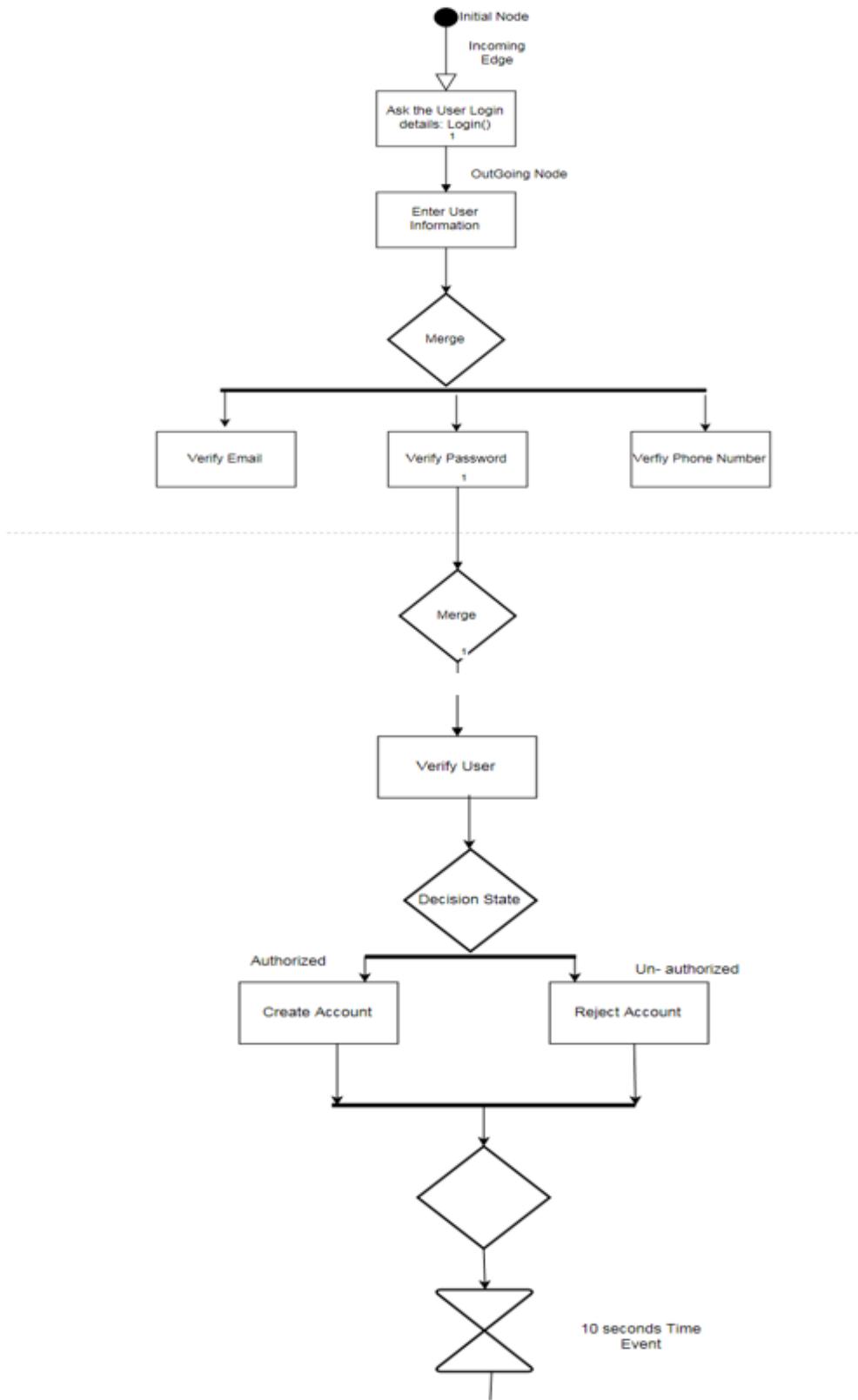
Movie theatres and show timing API's to be included.

## Class Diagram

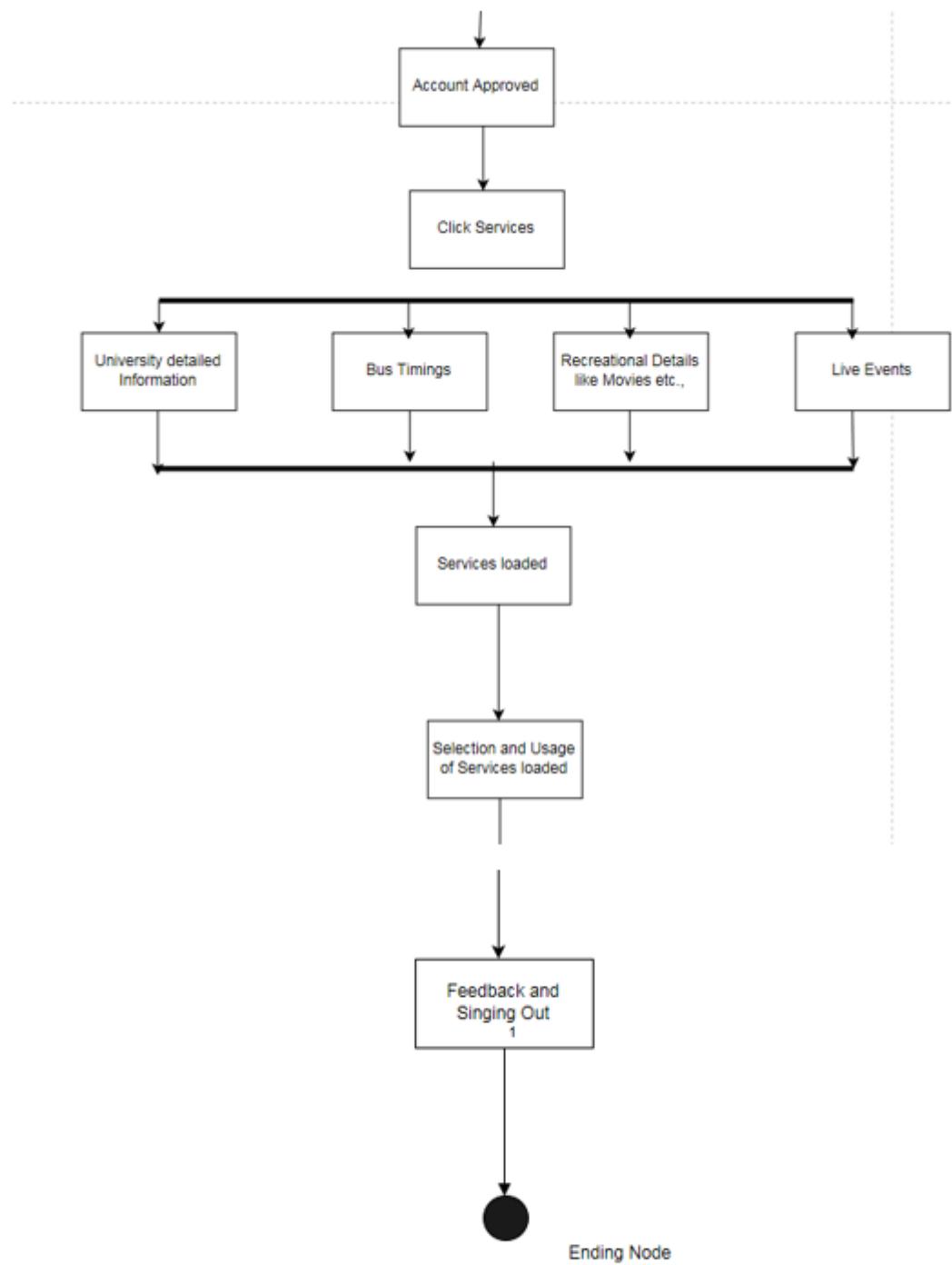


## Activity

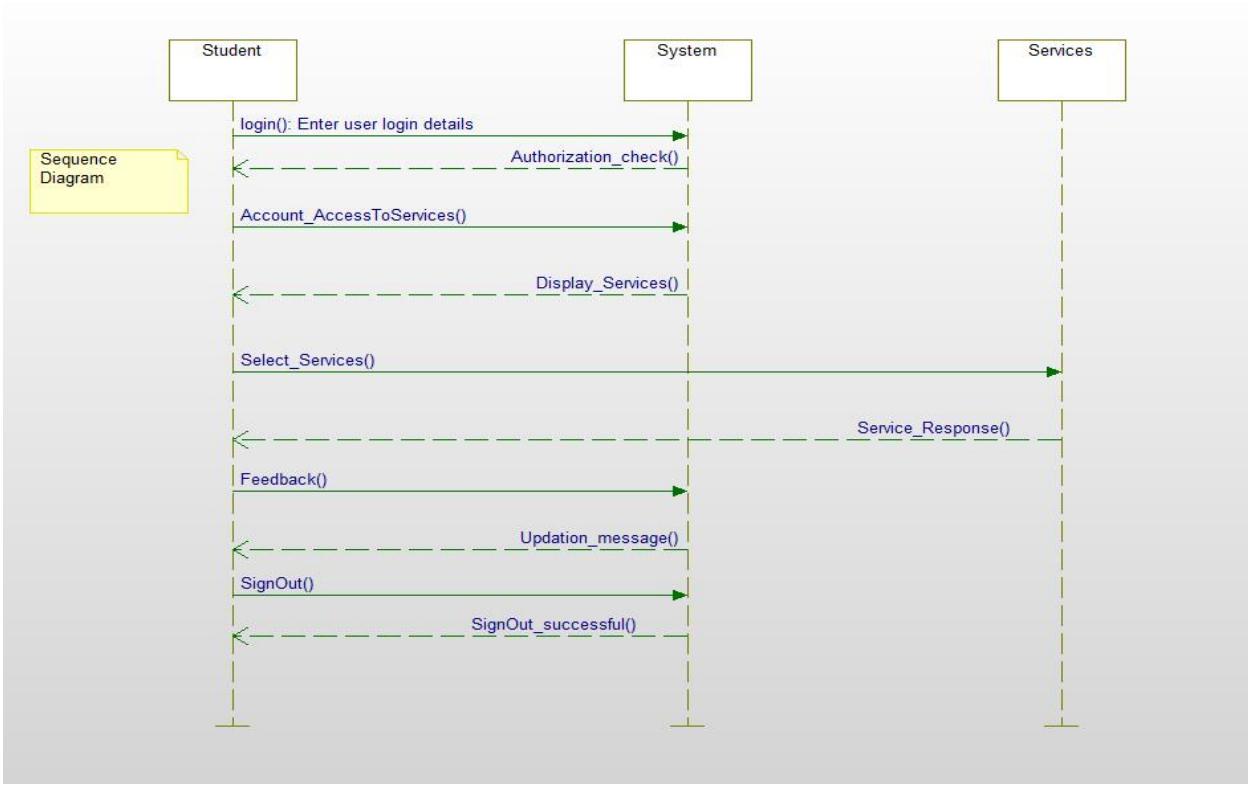
## Diagram

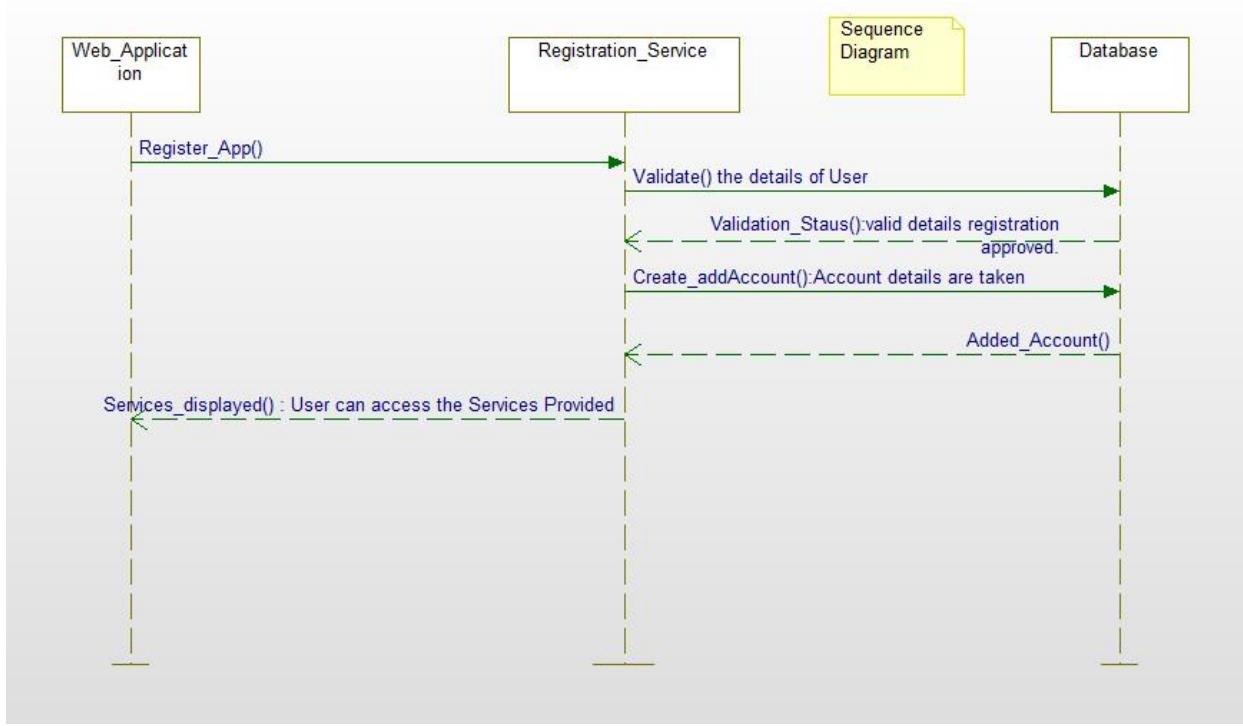


\



## Sequence Diagram





### **Service Specifications:**

- The application guides the student regarding each and every activity around him without the intervention of third person.
- The User is free to access any details about the University through the service columns provided.
- The User is always assisted with google maps guiding the routes in and around University.

### **Operational description:**

- The system checks for authorization while signing in for already registered user and also allows to create a account for a new user.
- The system displays featured page with every information regarding the university.

By selecting location in university end user can track the details of selected place, events around it and bus timings and navigation from that location.

### **Plan by Services (using ScrumDo)**

## Iterations

	Name	Stories	Start	End
	 Backlog	0 stories		
	 Database Creation	1 stories	Feb 12, 2015	Feb 25, 2015
	 User Interface Design	1 stories	Feb 26, 2015	Mar 18, 2015
	 Unit Testing & Integration	1 stories	Mar 19, 2015	Apr 08, 2015
	 Testing & Implementation	1 stories	Apr 09, 2015	Apr 29, 2015

 New Iteration

ScrumDo Search Project

## Database Creation Feb 12, 2015 - Feb 25, 2015

Scrum Board

Stories	Total Points	Points In Progress	Points Completed	Days Left
1	8	0	0	13

Burnup | Burndown | Stacked | Time

Add Story

### Stories

Filter Board

#1 As a member of the team, I want to create a database

[Todo](#) [Tasks](#) | 0 Comments 8

ScrumDo Search Project

## User Interface Design Feb 26, 2015 - Mar 18, 2015

Scrum Board

Stories	Total Points	Points In Progress	Points Completed
1	5	0	0

**Not Enough Data**

We don't have enough data to draw this burn up chart right now.

Some Tips:

1. Size your stories
2. Set the iteration dates to include today
3. Burn-Up charts are generated nightly

Burnup | Burndown | Stacked | Time

Add Story

### Stories

Filter Board

#2 As a member of the team, I want you to design the User Interface.

[Todo](#) [Tasks](#) | 0 Comments Preetham123 5

ScrumDo Search Project

## Unit Testing & Integration Mar 19, 2015 - Apr 08, 2015

Scrum Board

Stories

**1**

Total Points

**0**

Points In Progress

**0**

Points Completed

**0**

### Not Enough Data

We don't have enough data to draw this burn up chart right now.

Some Tips:

1. Size your stories
2. Set the iteration dates to include today
3. Burn-Up charts are generated nightly

[Burnup](#) | [Burndown](#) | [Stacked](#) | [Time](#)

Add Story

### Stories



Filter Board



#3 I want to test the functionality of each module and want to integrate them.

[Todo](#) [Tasks](#) | 0 Comments Preetham123

ScrumDo Search Project

## Testing & Implementation Apr 09, 2015 - Apr 29, 2015

Scrum Board

Stories

**1**

Total Points

**0.5**

Points In Progress

**0**

Points Completed

**0**

### Not Enough Data

We don't have enough data to draw this burn up chart right now.

Some Tips:

1. Size your stories
2. Set the iteration dates to include today
3. Burn-Up charts are generated nightly

[Burnup](#) | [Burndown](#) | [Stacked](#) | [Time](#)

Add Story

### Stories



Filter Board



#4 I want you both to test the final output and to implement it.

[Todo](#) [Tasks](#) | 0 Comments

**0.5**

## Risk Management

1. Since we are showing all the live Events going nearby we have to be up to date with the Database.
2. Our application should be compatible with the handheld devices and web applications.
3. We should be careful while uploading the events into the Database.
4. Response time of the application shouldn't be too slow and it cannot be off grid due to overload.

**Posting**                  **the**                  **class**                  **google**                  **site:**

[https://docs.google.com/spreadsheets/d/1QtbhKeCep4SvzP5gCYs1iGPyKQxhooCoGqLR5uYl8/  
edit#gid=2115159985](https://docs.google.com/spreadsheets/d/1QtbhKeCep4SvzP5gCYs1iGPyKQxhooCoGqLR5uYl8/edit#gid=2115159985)

## Bibliography

Google Maps API for Business <https://developers.google.com/maps/documentation/business/>

My-Spare Time [https://www.youtube.com/watch?v=Bf0\\_ZsJQxD8](https://www.youtube.com/watch?v=Bf0_ZsJQxD8)

[http://www.kcata.org/maps\\_schedules/](http://www.kcata.org/maps_schedules/)

<http://www.youtube.com/watch?v=w7TIS6NYdtk&feature=youtu.be>

<https://www.zomato.com/mobile>

<https://play.google.com/store/apps/details?id=com.tweakersoft.aroundme&hl=en>



# PROJECT INCREMENT 1

## Existing services/API:

- 1) <http://api.eventful.com/rest/events/search?...&keywords=books&location=San+Diego&date=Future>

This API is used to get the live events happening across a specific location where the location can be retrieved from location and geo sensor of android mobile device

- 2) <https://data.cityofchicago.org/resource/alternative-fuel-locations.json?>

This API also used to get the details of the location or city

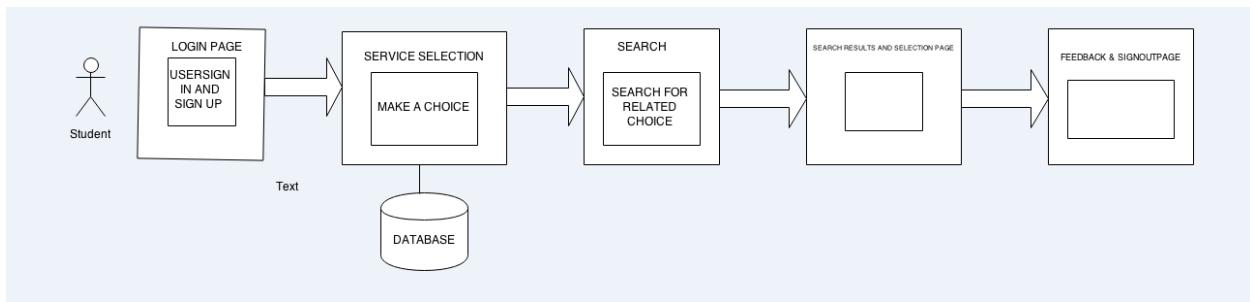
- 3) <https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true>

This API is used for location tagging and navigation purpose which can be used by end user in our application.

## Detail Design of Services:

Go Easy Web services:

We are developing a customized web service which includes all the required data at one place which going to ease the way of searching for the events by end user.



This architecture explains flow and behavioral description of each layer starting with Login page of our application to the data layer. The application mainly consists of three section which includes entertainment related events, university related events and transportation related events. We are using publically available API to connect to their end point and get the response (usually JSON).

## ScrumDO and stories:

**ScrumDo**  Search Organization

ASE SPRING PG3 Preetham123

**Upgrade your plan**

**Grace Period**

This organization is in a grace period. You have quick links for 2 more days.

Please feel free to email us at [support@scrumdo.com](mailto:support@scrumdo.com) for help.

**Projects**

- Dashboard
- Subscription
- Track Time
- Newsfeed
- Projects
- Files
- Time Sheets
- Members
- Teams
- Export
- Releases
- Organization Admin

**PG3**

**Switch Organization**

<https://www.scrumdo.com/projects/project/pg31/summary>

**ScrumDo**  Search Project

ASE SPRING PG3 PG3 Preetham123

## PG3

Velocity: 20 Total Stories: 13 Stories Completed: 8 Stories In Progress: 2 Iterations Left: 6

Add Story

Scrum Log Newsfeed

Hi There, It looks like you don't have a verified email set up. It's important...

[https://www.scrumdo.com/projects/project/pg31/all\\_iterations](https://www.scrumdo.com/projects/project/pg31/all_iterations)

**ScrumDo**  Search Project

ASE SPRING PG3 PG3 Preetham123

## Iterations

Name	Stories	Start	End
Backlog	4 stories		
Iteration1	5 stories	Feb 10, 2015	Feb 25, 2015
Iteration2	2 stories	Mar 01, 2015	Mar 18, 2015
Iteration3	2 stories	Mar 20, 2015	Apr 08, 2015
Iteration4	0 stories	Apr 10, 2015	Apr 29, 2015
Project (Poster) Presentation and final report	0 stories	May 01, 2015	May 08, 2015

**New Iteration**

**Quick Links**

- Project Summary
- Epics
- Iteration Planning
- Chat
- History
- Predictions
- Planning Poker
- Iterations
- Backlog
- Project (Poster)
- Presentation and final report
- Iteration4
- Iteration3
- Iteration2
- Iteration1

ScrumDo  ASE SPRING PG3 PG3 Preetham123

Stories	Total Points	Points In Progress	Points Completed
5	100	20	80

**Stories**

Add Story

#19 linking the database to android project find the best methods to connect the database to our android project <b>Doing</b> Tasks   0 Comments Preetham123	20
#18 creating the database install the oracle 11g software and create tables related to our project <b>Done</b> Tasks   0 Comments prasadreddy2348	20
#17 designing the user interface design the mobile web view of our project using android studio <b>Done</b> Tasks   0 Comments PKBOLLARAM	20
#15 create github a/c of our project create github a/c of our project <b>Done</b> Tasks   0 Comments Preetham123, prasadreddy2349	20
#16 Search for api's related to our project search api's such as google maps.api's live events in a city based on location <b>Done</b> Tasks   0 Comments LakshmiPriyanka	20

Hi There. It looks like you don't have a verified email set up. It's important...

ScrumDo  ASE SPRING PG3 PG3 Preetham123

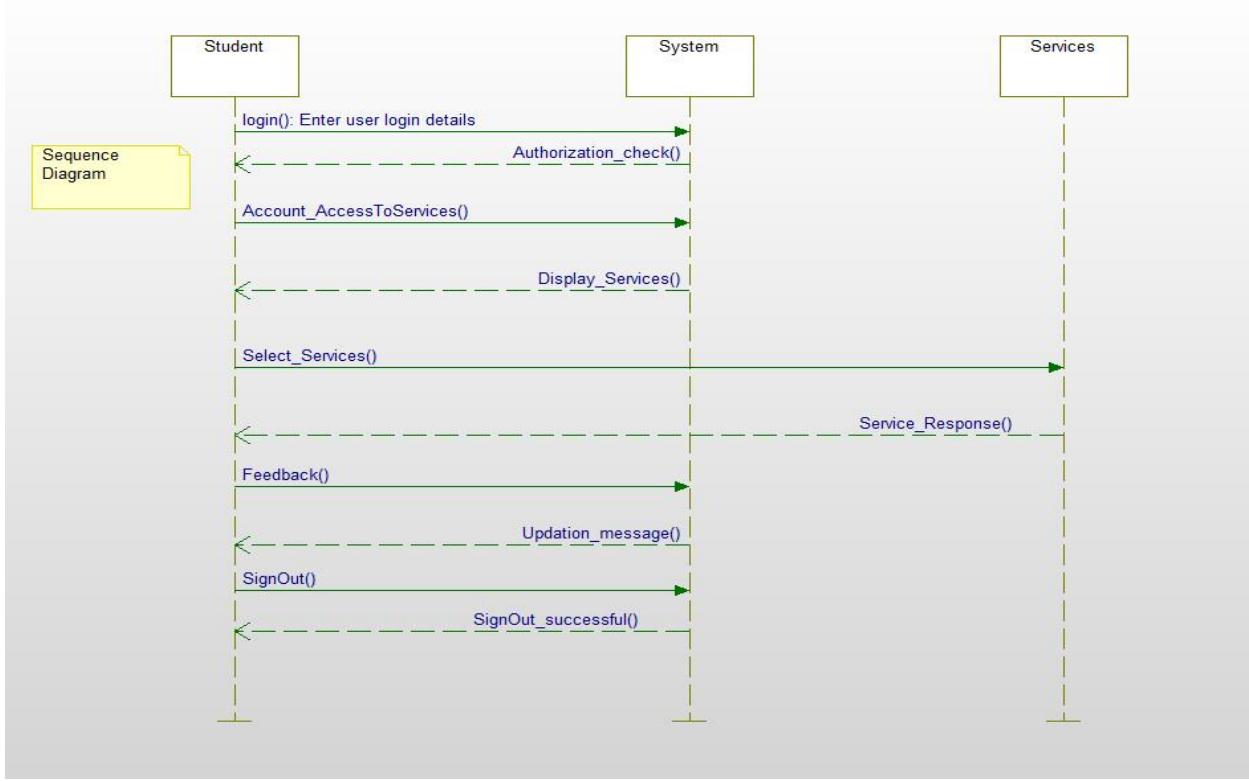
## PG3

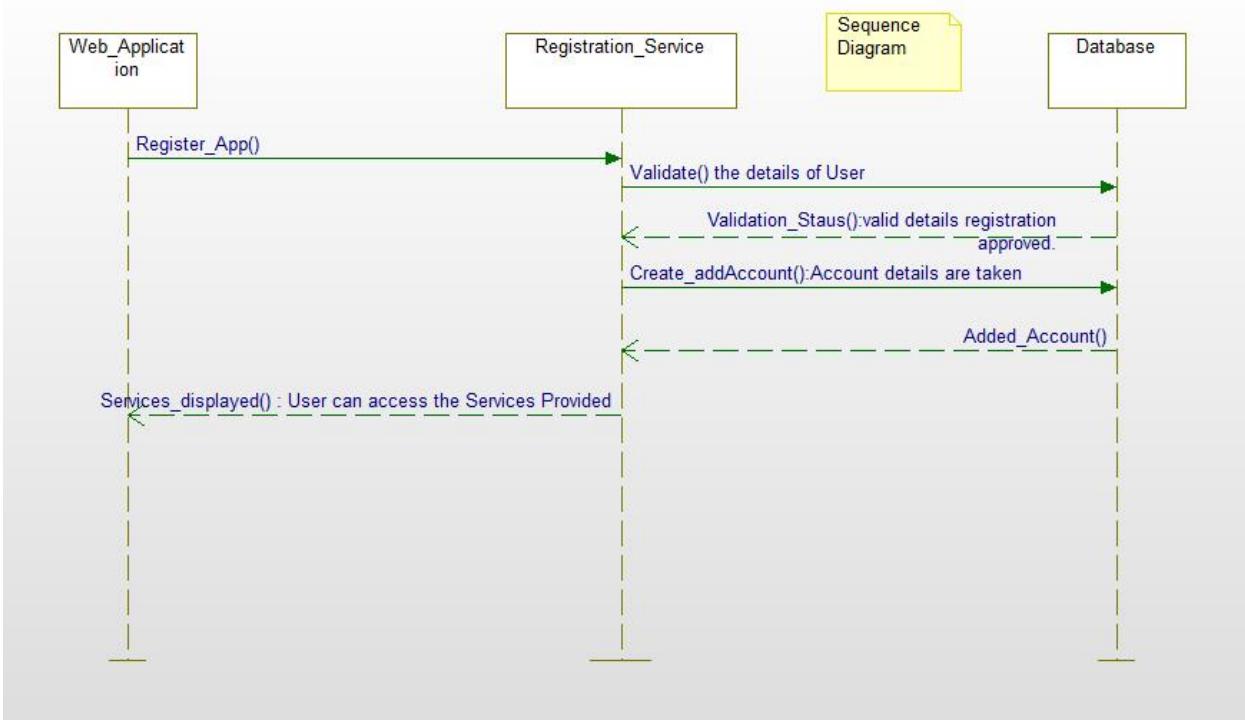
Project Velocity 20 Points	Last Iteration 5 Stories 80 Points	Quick Links
Backlog	Iteration1	Project Summary Epics Iteration Planning Chat History Predictions Planning Poker Iterations Backlog Project (Poster) Presentation and final report 0 stories Iteration4 0 stories Iteration3 2 stories Iteration2 2 stories Iteration1 5 stories
#9 Add planning while creating iteration <b>Done</b> Tasks   0 Comments PKBOLLARAM, prasadreddy2349	#19 linking the database to android project find the best methods to connect the database to our android project <b>Doing</b> Tasks   0 Comments Preetham123	20
#8 Create Iteration Create an iteration <b>Done</b> Tasks   0 Comments	#18 creating the database install the oracle 11g software and create tables related to our project <b>Done</b> Tasks   0 Comments prasadreddy2348	20
#1 create github link and add to your scrumdo Please create git hub Links and map to your scrumdo accounts so that u can get project repositories any time <b>Done</b> Tasks   0 Comments Preetham123, prasadreddy2349	#17 designing the user interface design the mobile web view of our project using android studio <b>Done</b> Tasks   0 Comments PKBOLLARAM	20
#2 Phase 2 Create a Visio login <b>Done</b> Tasks   0 Comments None prasadreddy2349	#15 create github a/c of our project create github a/c of our project <b>Done</b> Tasks   0 Comments Preetham123, prasadreddy2349	20
	#16 Search for api's related to our project search api's such as google maps.api's live events in a city based on location <b>Done</b> Tasks   0 Comments LakshmiPriyanka	20

Hi There. It looks like you don't have a verified email set up. It's important...

<https://www.scrumdo.com> 5 Stories 100 Points 14:30 Hrs

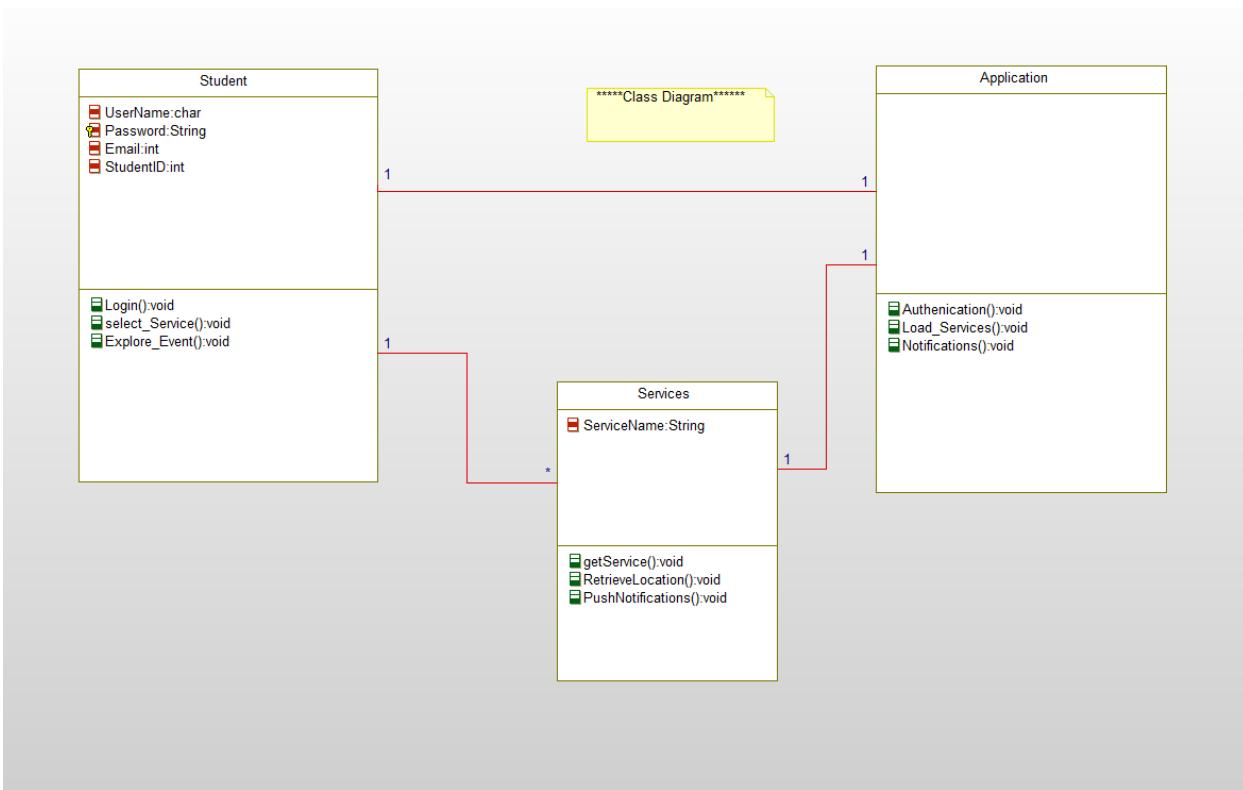
## Sequence Diagram:



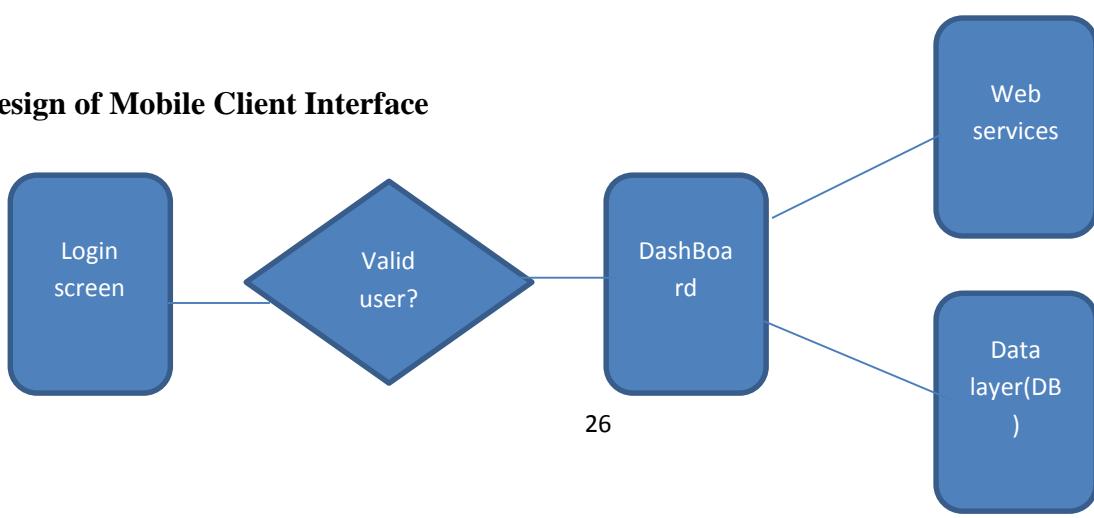


This sequence diagram explains an abstraction level of system implementation with required services and flow of each service. User can register to events using Registration services offered by application which gets the users data and persists to database. This very same data can be fetched from the data layer using display services call and dashboard is updated with these results. Along with the useful live events data a small level navigation part is also going to be implemented in this application.

## Class Diagram:



## Design of Mobile Client Interface



## **Design of Unit test cases:**

To be implemented in following increment.

## **Implementation**

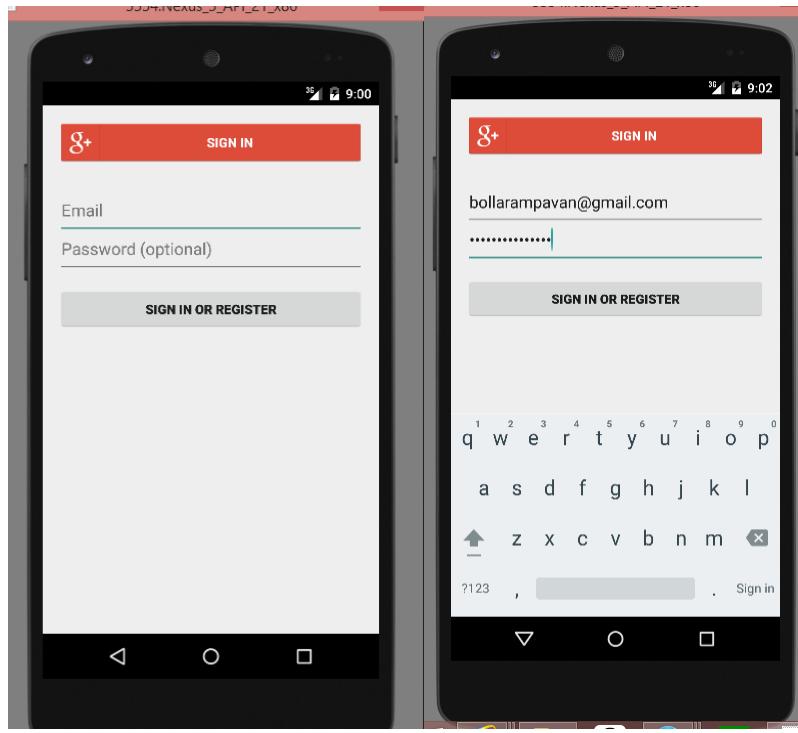
### **Implementation of REST services**

As of now we are using three of available web service API's to get the data

- <http://api.eventful.com/rest/events/search?...&keywords=books&location=San+Diego&date=Future>
- <https://data.cityofchicago.org/resource/alternative-fuel-locations.json?>
- <https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true>

Customized REST service needs to implemented for enhancement of our application which will be further detailed next increment.

## Implementation of User interface(Mobile Interface)

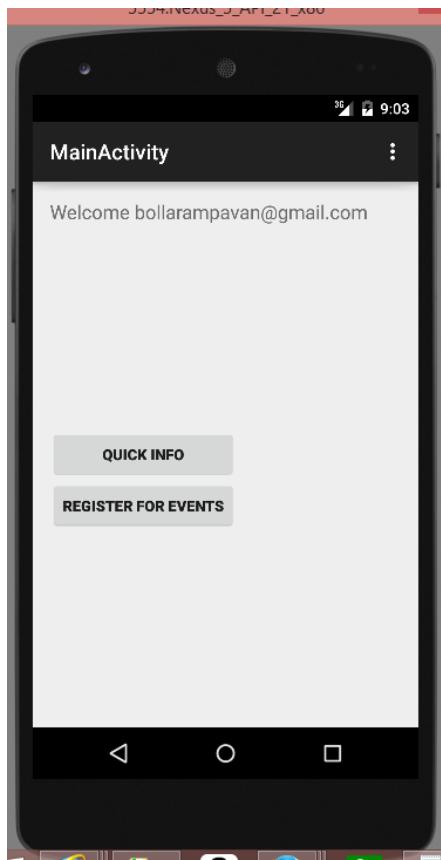


Login screen

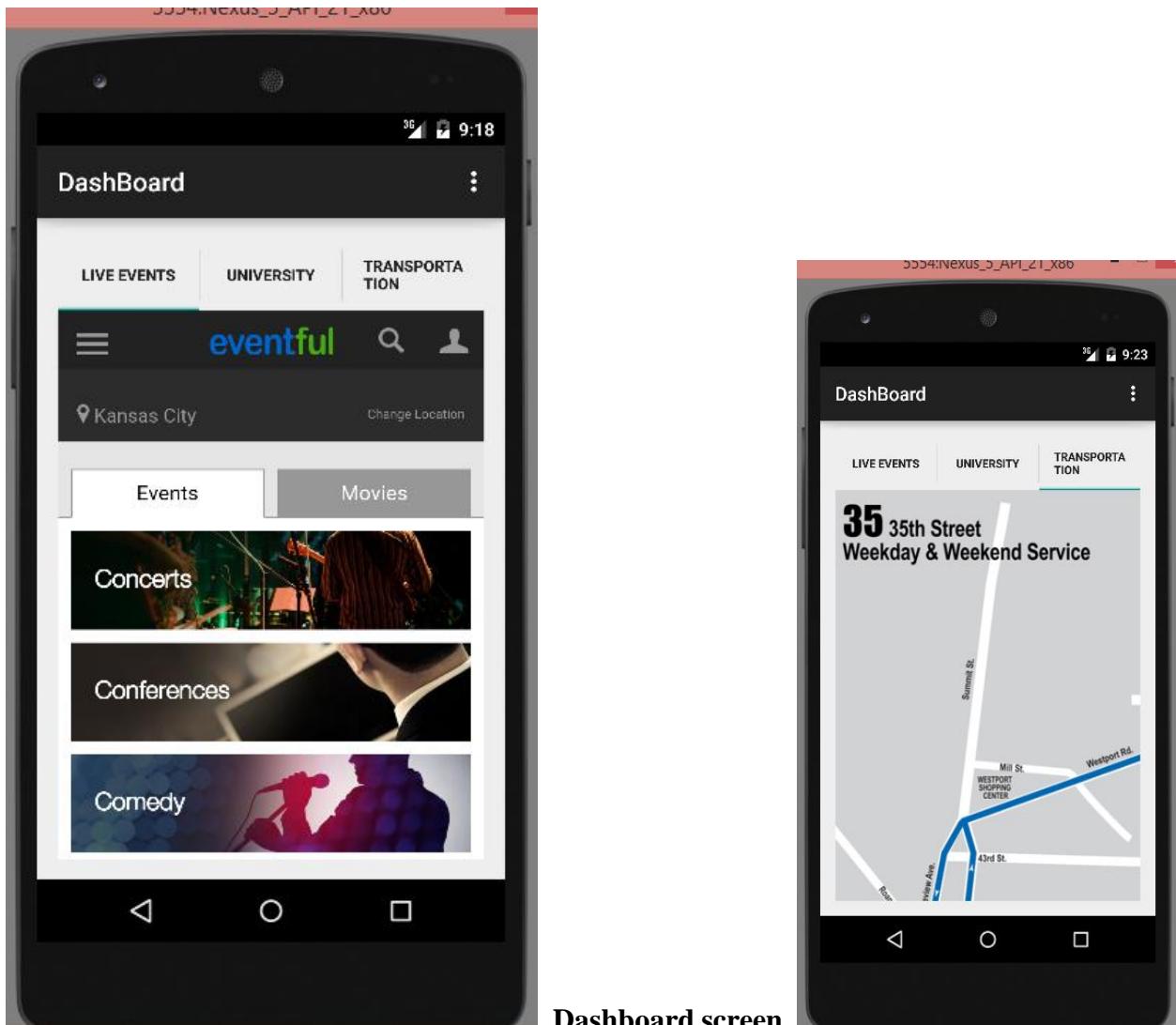
Login to system uses existing google+ sign in services so that end user can easily login with his/her gmail id's which reduces the use of separate database for our application .This login screen contains two separate parts which are

**SIGN IN :** This button used incase user is not registered with our application

**EMAIL** and **PASSWORD** fields which takes input from user and validates the user details. Once validated user can login to application and can check his dashboard



Welcome Page – This page displays the user with which email id he/she login to application and provides **QuickInfo** and **Register for Events** buttons. If we click on quick info user is navigated to dashboard screen with useful info about live events and all.



Dashboard screen

This is dashboard screen which will be displayed when user login to application. This is complete mashup kind of application where end user can easily check the required tab.

We simplified GUI with simple tab host concept so that end user can easily navigate between Live-events, University and transportation tabs. The content is displayed according to the selection of tab.

## Database Design implementation

```
SQL> create table registerevents(event_name varchar2(10) NOT NULL,event_type var  
char2(10),location varchar2(25),email_id varchar2(20) NOT NULL, FOREIGN KEY(email_id) REFERENCES login (email_id));  
Table created.  
SQL> desc registerevents  


| Name       | Null?    | Type         |
|------------|----------|--------------|
| EVENT_NAME | NOT NULL | VARCHAR2(10) |
| EVENT_TYPE |          | VARCHAR2(10) |
| LOCATION   |          | VARCHAR2(25) |
| EMAIL_ID   | NOT NULL | VARCHAR2(20) |

  
SQL>
```

```

SQL> create table university (
  2  univ_name varchar2<20>,
  3  location varchar2<20>,
  4  event_name varchar2<20>,
  5  event_type varchar2<20>);

Table created.

SQL> desc university
Name          Null?    Type
-----        -----
UNIV_NAME           VARCHAR2<20>
LOCATION            VARCHAR2<20>
EVENT_NAME          VARCHAR2<20>
EVENT_TYPE          VARCHAR2<20>

SQL>

```

Run SQL Command Line

```

Copyright <c> 1982, 2014, Oracle. All rights reserved.

SQL> conn system/Preetham123$  

Connected.  

SQL> create table login (
  2  email_id varchar2<20> NOT NULL PRIMARY KEY,  

  3  conn system/Preetham123$  

  4  

SQL> drop table login;  

Table dropped.  

SQL> create table login (
  2  email_id varchar2<20> NOT NULL PRIMARY KEY,  

  3  password varchar2<20> NOT NULL);  

Table created.  

SQL> desc login
Name          Null?    Type
-----        -----
EMAIL_ID          NOT NULL VARCHAR2<20>
PASSWORD          NOT NULL VARCHAR2<20>

SQL>

```

## Implementation of test cases:

Yet to be implemented.

## Testing

Will be detailed in next increment plan.

## Deployment

### ScrumDo

<https://www.scrumdo.com/projects/project/pg31/summary>

### GitHub

**Source code-**[\*\*https://github.com/pavankumar-b/ASEspringSem/Increment1\*\*](https://github.com/pavankumar-b/ASEspringSem/Increment1)

**Documentation-**[\*\*https://github.com/pavankumar-b/ASEspringSem/IncrementDoc\*\*](https://github.com/pavankumar-b/ASEspringSem/IncrementDoc)

## **Project Management**

### **Implementation status Report**

#### **Work completed:**

- **Description**  
Android Login screen is implemented  
Successfully implemented Dashboard screen with tabhost interface  
Login screen with google sign in option  
Separate tab screen for each functionality  
Database table creation
- **Responsibility**  
Task1: Android Login screen and implementation of tab interface / Pavankumar Bollaram  
Task2: Web API's(Google SignIn) implementation/Lakshmi Priyanka  
Task3: Database creation/ Varaprasad  
Task4:Dashboard screen design and navigation/ Preetham kumar
- **Time Taken**  
200 Man hours
- **Contribution**  
PavanKumar Bollaram (25%)  
Preetham kumar(25%)  
Lakshmi Priyanka(25%)  
Varaprasad(25%)

## **Work to be completed**

Task1: Custom Web API design and Implementation (pavankumar Bollaram)

Task2: Database integration with android interface (varaprasad/Preetham kumar)

Task3: University API implementation and integration (Lakshmi Priyanka)

Time to be taken (400 Man hours)

### **Issues/Concerns**

Facing issue while establishing connection with database

Incomplete information about web API's and their documentation

Android studio performance is very poor and time taking

# PROJECT INCREMENT 2

## IMPORT EXISTING SERVICES/API:

The following services are likely to be consumed in our project:

### Existing services/API:

- <http://api.eventful.com/rest/events/search?...&keywords=books&location=San+Diego&date=Future>  
The above API acknowledges a user about the live events happening across a specified location where the location can be retrieved from location and geo sensor of android mobile device
- <https://data.cityofchicago.org/resource/alternative-fuel-locations.json?>  
This API also used to get the details of the location or city
- <https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true>
  - This API is used for location tagging and navigation purpose which can be used by user in our application.

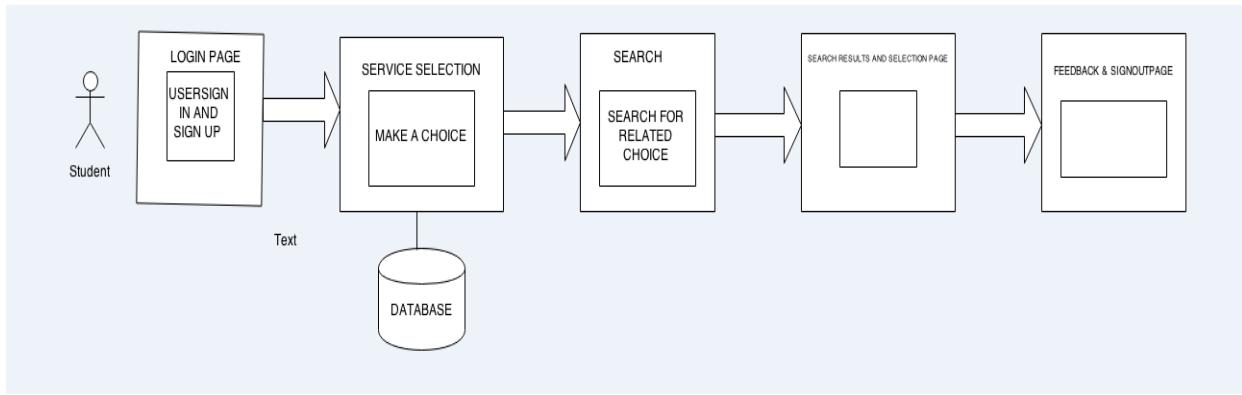
**Registration API:** developed custom registration Api which stores user login details into database table.

**Login API:** This custom api Deals with user login to application. It does validation with given user inputs

## DETAIL DESIGN OF SERVICES:

Go Easy Web services:

We are intended to develop customized web service which includes all the required data at one place which going to ease the way of searching for the events by end user. The below figure depicts the architecture which explains flow and behavioral description of each layer starting with login page of our application to the data layer.



The application mainly consists of three section which includes entertainment related events, university related events and transportation related events. We are using publically available API to connect to their end point and get the response (usually JSON).

### **Scrum Do and Stories:**

The screenshot shows the ScrumDo application interface. At the top, there's a navigation bar with links like 'Project Summary', 'Epic', 'Iteration Planning', 'Chat', 'History', 'Predictions', 'Planning Poker', and 'Iterations'. On the right, there's a sidebar titled 'Quick Links' with links to 'Project Summary', 'Epic', 'Iteration Planning', 'Chat', 'History', 'Predictions', 'Planning Poker', and 'Iterations'. The main area has two sections: 'Backlog' and 'Stories'.

**Backlog:** Shows summary statistics: Stories (4), Total Points (50), Points In Progress (0), and Points Completed (50). There's also an 'Add Story' button.

**Stories:** A list of stories with their details and progress. The stories are:

- #19 linking the database to android project: Done, Tasks: 0, Comments: 0, Preetham123, prasadreddy2349, Points: 20
- #6 Create iteration: Done, Tasks: 0, Comments: 0, Preetham123, prasadreddy2349, Points: 5
- #1 create github link and add to your scrumdo: Done, Tasks: 0, Comments: 0, Preetham123, prasadreddy2349, Points: 5
- #2 Phase 2: Done, Tasks: 0, Comments: 0, None, prasadreddy2349, Points: 20

A tip at the bottom says: 'ScrumDo Tip: Use tags to group stories into features.'

ScrumDo  Search Project

Iteration2 Mar 01, 2015 - Mar 18, 2015

Scrum Board

Stories 5 Total Points 100 Points In Progress 0 Points Completed 100

Stories

Add Story

Filter Board

Stories

#422 search for bus routes data go through kclata.org website and search for bus routes connecting the univ	20
#421 create documentation of iteration2 create documentation of iteration2	20
#420 search for university events go through university website and collect all the events	20
#411 work on database to integrate it with our app watch videos related to database and try to implement them	20
#412 improve the login screen and dashboard UI make use of rich resources to improve the UI	20

ScrumDo Tip: Burndown charts are generated nightly, subscribe to a pad plan to update charts on the fly.

ScrumDo  Search Project

ASE SPRING PG3 PG3 prasadreddy2349

Iterations

Name	Stories	Start	End
Backlog	4 stories		
Iteration1	4 stories	Feb 10, 2015	Feb 25, 2015
Iteration2	5 stories	Mar 01, 2015	Mar 18, 2015
Iteration 3	2 stories	Mar 20, 2015	Apr 06, 2015
Iteration4	0 stories	Apr 10, 2015	Apr 29, 2015
Project (Poster) Presentation and final report	0 stories	May 01, 2015	May 08, 2015

New Iteration

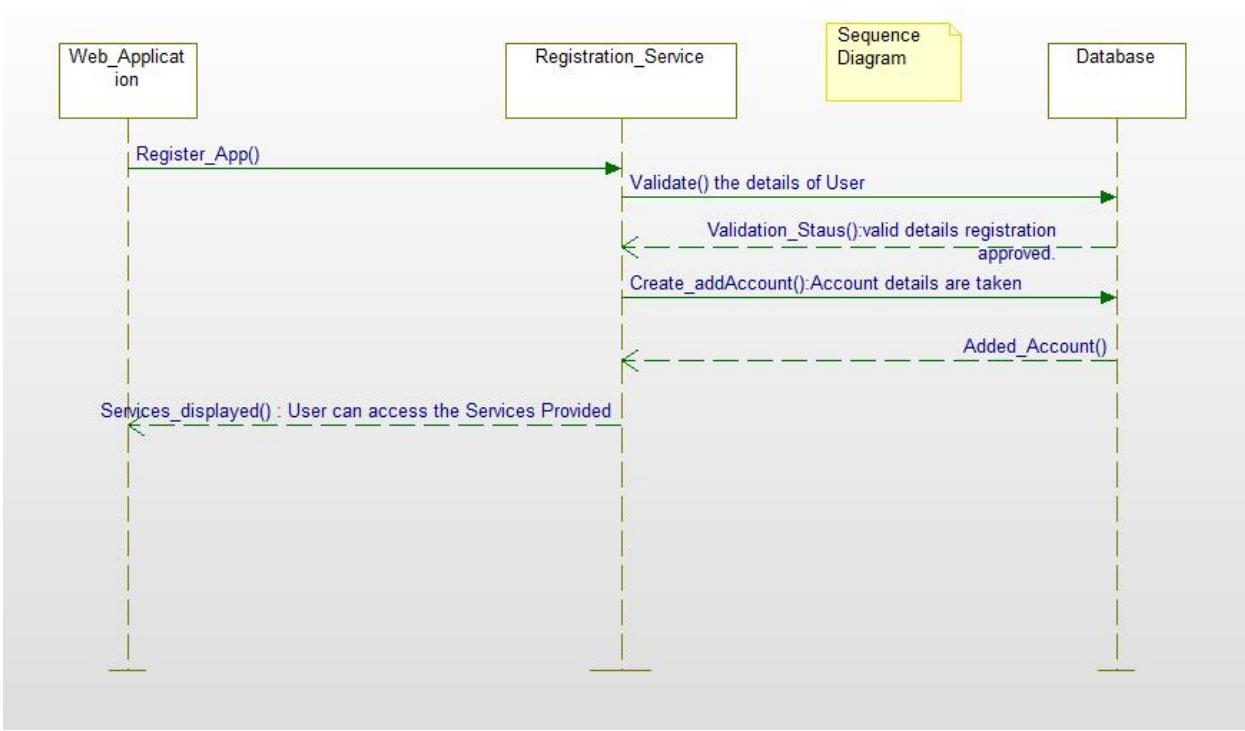
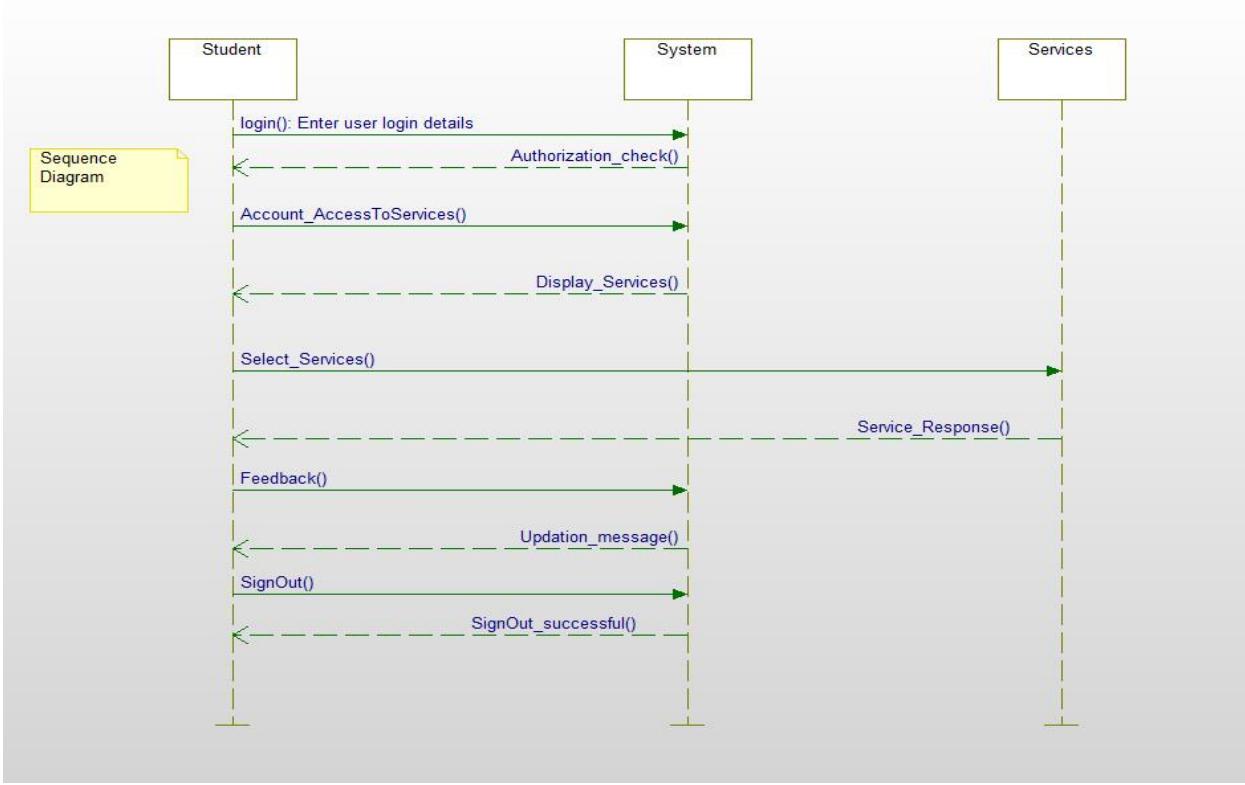
ScrumDo Tip: We have an ATDD addon to help create Cucumber tests

The screenshot shows the ScrumDo interface for Project PG3. On the left, the 'Backlog' board lists several user stories with their details and status (Done or In Progress). On the right, the 'Iteration2' board shows the same stories mapped to specific tasks and team members. The top navigation bar includes links for 'Project Summary', 'Epics', 'Iteration Planning', 'Chat', 'History', 'Predictions', 'Planning Poker', and 'All Iterations'. A sidebar on the right provides quick links to various project sections.

Story	Description	Owner	Priority
#19 linking the database to android project	find the best methods to connect the database to our android project	Preetham123, prasadreddy2349	20
#20 Create iteration	Create an iteration	Preetham123, prasadreddy2349	5
#21 create documentation of iteration	Please create git hub Links and map to your scrumdo accounts so that u can get project repositories any time	Preetham123, prasadreddy2349	5
#22 search for bus routes data	go through kclata.org website and search for bus routes connecting the univ	Preetham123, prasadreddy2349	20
#23 work on database to integrate it with our app	watch videos related to databases and try to implement them	PKBOLLARAM, Preetham123, prasadreddy2349, LakshmiPriyanka	20
#24 improve the login screen and dashboard UI	make use of rich resources to improve the UI	PKBOLLARAM	20

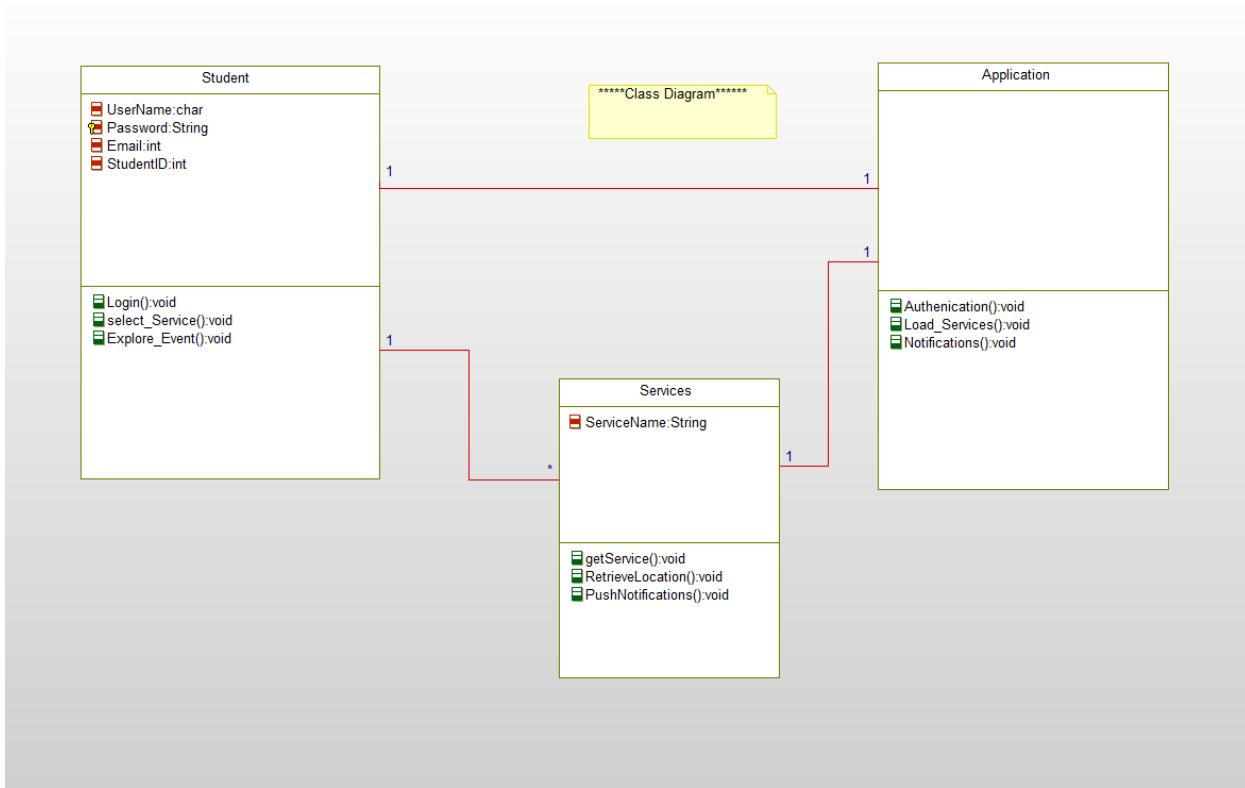
## IMPLEMENTATION:

Sequence Diagrams:



Sequence diagram is message interaction chart diagram that depicts how processes order with one another and what process they do. It depicts the objects and classes interactions in time sequence. The above sequence diagram explains an abstraction level of system implementation with required services and flow of each service. User can register to events using Registration services offered by application which gets the users data and persists to database. This very same data can be fetched from the data layer using display services call and dashboard is updated with these results. Along with the useful live events data a small level navigation part is also going to be implemented in this application.

### Class Diagram:



A class diagram is a type of structure diagram which describes the attributes, classes and methods (or operations) among the elements involved.

### TESTING:

Tested web service availability through android

Tested user login validations with different probable conditions

Tested registration process with few specific conditions

## **IMPLEMENTATION:**

Implementation of REST Services:

Two API's are implemented as part of this increment. They are i) Login API  
ii) Registration API

### **Login API:**

Login API deals with login procedure which does validation on user inputs. Checks whether user is registered to application or not.

Separate database table is created which stores the registered details and does validations by fetching the details from this database table

dbo.Table4: Tabl...\\APP_DATA\\DB.MDF		
	username	password
	pavan	pavan
	pk	pk
▶	bollarampavan...	bpavan
	pbk	pbk
	pvn	pvn
	pbc6d@umkc	pavankumar
	preetham@gmail	preetham
	sm8xd@umkc	srikanroanoke
	google@gmail	google

Table created to store Login details and Login API connects to this table and fetches data.

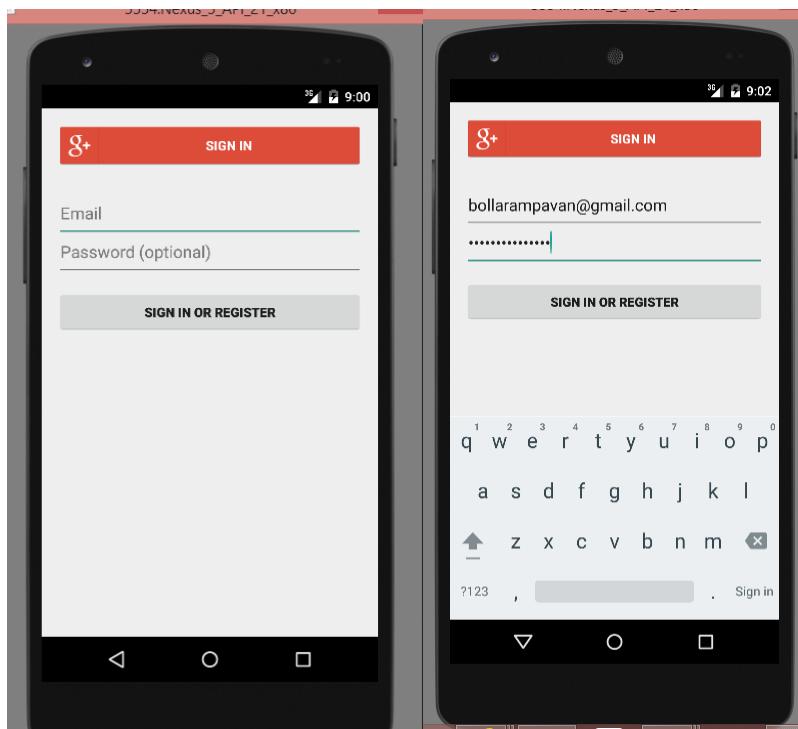
### **Registration API:**

Registration API deals with registration of end user into application as of now we are accepting email id as primary attribute and password as second.

End user with email id and password which were given at the moment of registration can login into application.

## **Implementation of User Interface (Mobile Interface):**

### **Screen I: Login Screen**



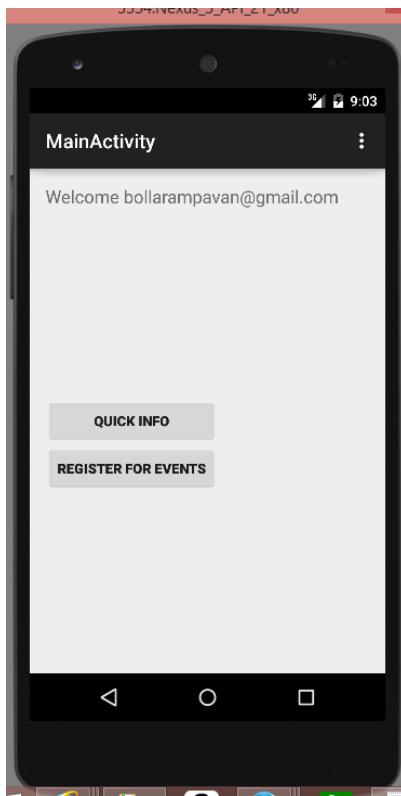
**Login Screen**

Login to system uses existing google+ sign in services so that end user can easily login with his/her Gmail id's which reduces the use of separate database for our application .This login screen contains two separate parts which are

**Sign in and Register:** This button used incase user is already registered with our application. He can directly go into access of further screens. Or else if he is not registered already it generates into new screen showing you are not registered / wrong password; login again.

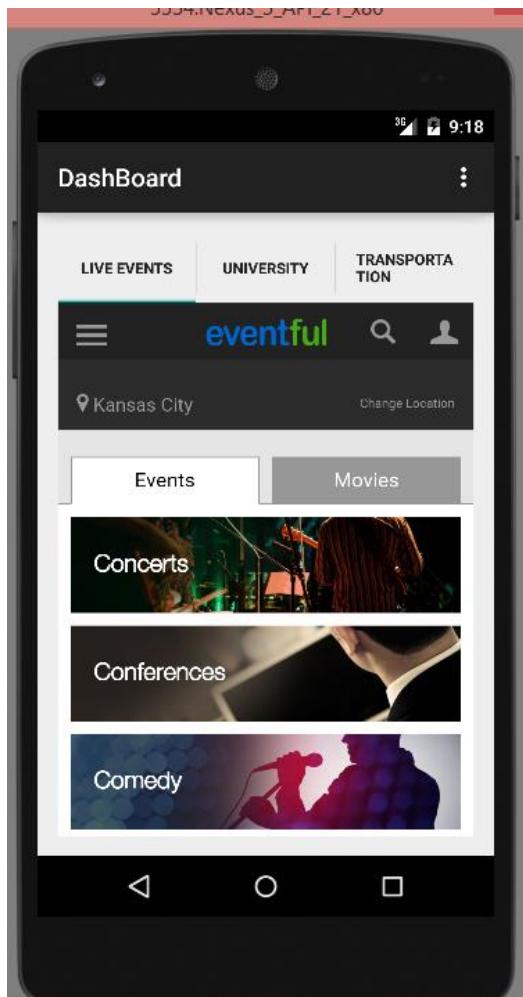
**Email and Password:** These fields which takes input from user and validates the user details which are already into the database when registered. Once validated user can login to application it pops into his dashboard.

## Welcome Page (Screen II)



Welcome Page: The above figure depicts the welcome page displayed to the user when he / she login into the application. We included two Events buttons **Quick-Info** and **Register for Events**. On click of Quick Info, the button navigates into dashboard screen which is categorized into Live-Events, University and Transportation.

### Dashboard (Screen III)



Live Event Tab.



Transportation Page

## DASHBOARD

Dashboard: This is dashboard screen which will be displayed when user press the Quick info button in Screen II. It is completely mashup application where user can easily check the required tab.

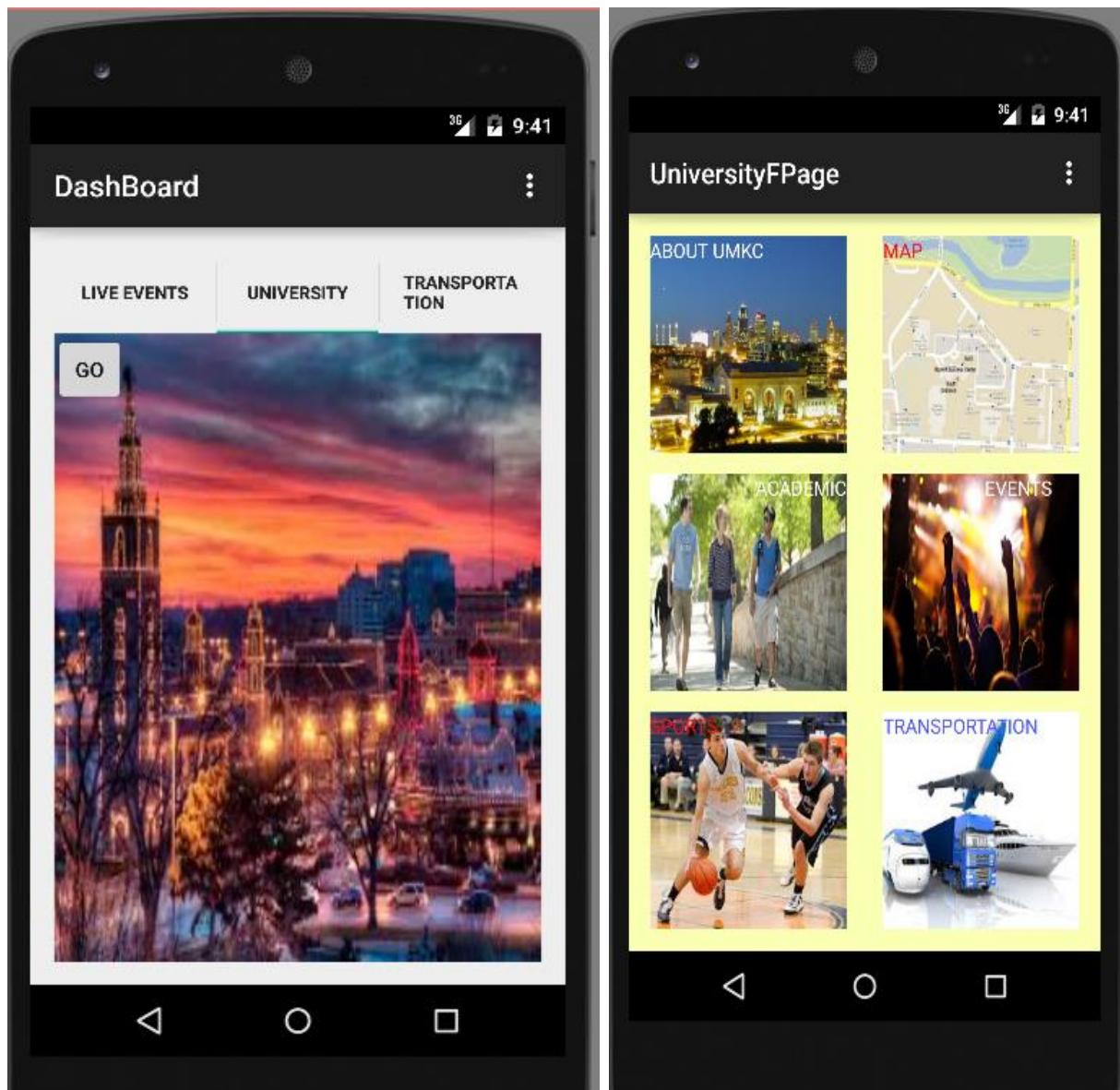
We simplified GUI with simple tab host concept so that end user can easily navigate between Live-events, University and Transportation tabs. The content is displayed according to the selection of tabs.

## University (Screen IV)

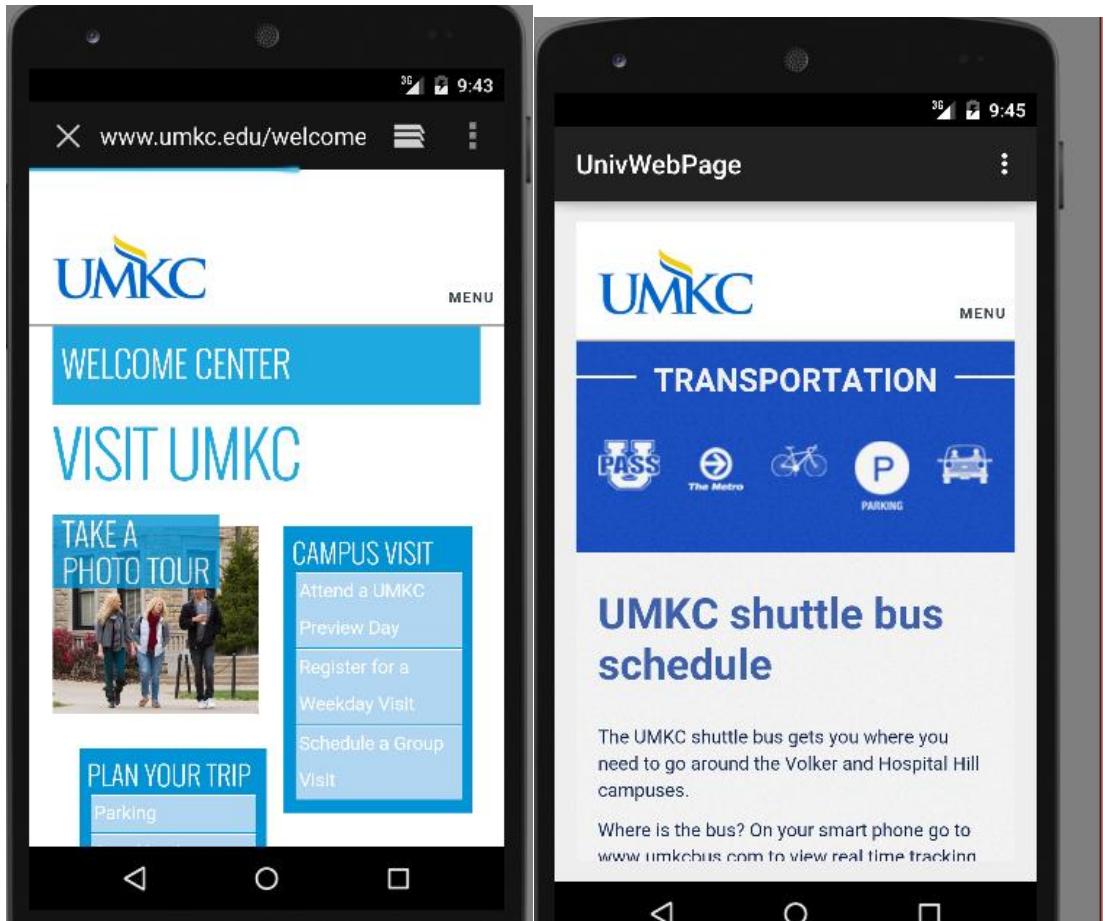
University details are presented to the user with tab in the Dashboard and when clicked navigated to different screen.

This screen consists of all data regarding university and it can be further improvised using deeper user interface

As of now we included few of the basic required data sections across screen they are as shown in the following images



The main dashboard screen consists of data sections like About Umkc, Map, Transportation, Academic details and sports sections.



## Database Design implementation

dbo.Table4: Tabl...\\APP_DATA\\DB.MDF)		Table4: Query(pv...	
	date	event	location
►	3/18/2015	An exhibition b...	umkc gallery of...
	3/18/2015	wind down we...	student union ,...
	3/19/2015	Civic Connecti...	Student Union, ...
	3/20/2015	Spring Opera	White Recital H...
	4/1/2015	Denim Day Drive	105 Haag Hall
	4/1/2015	Sarah Frisof, flu...	Grant Recital Hall
	4/2/2015	UMKC Health S...	Pierson Auditor...
	4/3/2015	New Historical ...	Cockefair Hall 2...
	4/3/2015	Clean Commut...	Swinney Recrea...

	bus number	route	time
	155	plaza-blueridge	07:40:00
	MAX	downtown-51st	08:00:00
	155	blueridge-plaza	08:20:00
	MAX	51st-downtown	08:40:00
	155	plaza-blueridge	08:40:00
	MAX	downtown-51st	09:00:00
	MAX	51st-downtown	09:10:00
	155	blueridge-plaza	09:20:00
	MAX	downtown-51st	09:30:00
	MAX	51st-downtown	09:40:00
	155	plaza-blueridge	10:00:00

Created database to store live events happening around umkc and web service is implemented to fetch these events

## DEPLOYMENT:

ScrumDo link:

<https://www.scrumdo.com/projects/project/pg31/summary>

Git Hub Link:

**Source code-** <https://github.com/pavankumar-b/ASEspringSem/Increment2>

**Documentation-**<https://github.com/pavankumar-b/ASEspringSem/IncrementDoc2>

## PROJECT MANAGEMENT:

Implementation Status Report

### Work Completed:

#### Description

- Design of Android Login Screen.
- Successfully implemented Login Functionality using web API and database. Validation of login users with already registered email and password or even when registered with Gmail.

- We are Including Separate Simple tab host interface.
- Creating, validating and the database tables of user registered and while signing in.
- University, Transportation and Live Events tabs (On click) are implemented successfully.
- Tables for university live events and transportation tables were created.

## **Responsibility**

Task 1: Implementation of web service / Login API

Pavankumar Bollaram/ Varaprasad jagg

Task 2: Implementation of web service / Registration API

Preetham kumar / Lakshmi priyanka

Task 3: Database table creation

Pavan kumar Bollaram, varaprasad,preetham kumar, Priyanka

Task 4: Extended User interface design including dashboard screens

Pavan Kumar, Preetham kumar, Varaprasad jagg, Priyanka

## **Time Taken**

250 Man Hours

## **Contribution**

Pavan Kumar Bollaram (25%)

Preetham Kumar Danaboina (25%)

Lakshmi Priyanka (25%)

Vara Prasad (25%)

## **Work to be completed:**

**Task1:** User interface design for custom API events fetch

**Task2:** Improvise UI

**Task3:** Google maps integration and GPS location finding

**Risk Management:**

Very slow processing speed of android

Web services API needs to be in Run state always otherwise just throws 404 error.

# PROJECT INCREMENT 3

## Introduction

Go Easy, is a mobile based application which is helpful for novice students who are attending a school or university. Every Student when attending a school at new location explicitly needs someone to guide them regarding school details and information about transportation in and around his school. Using this app a student can find out University Events, Transportation details, Live Events happening around his place. This app uses a friendly user interface which enables simple easy access with existing services. The aim of the project is to guide a student effectively without the need of a third party. The project is implemented on android platform.

## Significance

Even though there are many apps specific to a particular school what we have noticed is all these apps do not cover all the topics such as transportation details, events happening around the university. So we decided to develop an app which covers all these topics and would be helpful to the new students attending a school. We need to create various web applications capable of managing a user's interface collection to provide a relevant information for the student attending a school.

## API's and web services used

The below services are consumed in our project:

- <http://api.eventful.com/rest/events/search?...&keywords=books&location=San+Diego&date=Future>

This Eventful API is collection of events going around in particular city. This API acknowledges about live concerts, exhibitions and sport activity held in the city. Eventful.com open unique platform that enables web applications to maximum take advantage of Eventful's data and features using Eventful API.

Eventful's data is categorized using geo-graphical area and some other event parameters.

- <https://data.cityofchicago.org/resource/alternative-fuel-locations.json?>  
This API is also used to retrieve data from particular location using geographical location basis.
- <https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true>
  - [This API is provides help to explore a location and monitor maps for a particular place. Navigation purposes are also met through this API in our application. . This is usually helpful for people programming with java scripts and object orientated programming. The user should be familiar with the how to explore the google maps and how to navigate in point of view.](#)

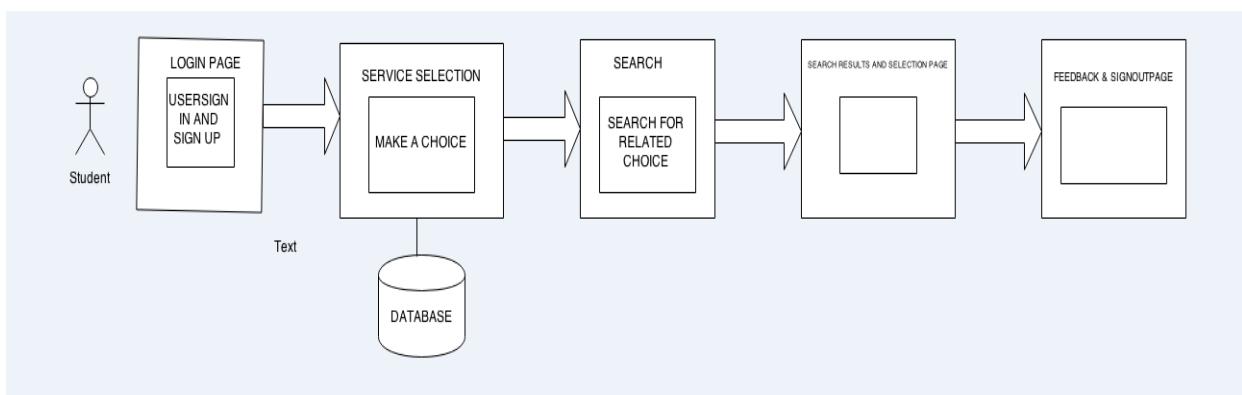
### Custom API 's used in project

- [http://kc-sce-cs551.kc.umkc.edu/aspnet\\_client/Group3/Transportation/Service1.svc/transportation/details](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details)
- <http://localhost:56120/Service1.svc/GetEventsDetails>
- <http://localhost:56120/Service1.svc/GetSportsDetails>

## **DETAIL DESIGN OF SERVICES:**

Go Easy Web services:

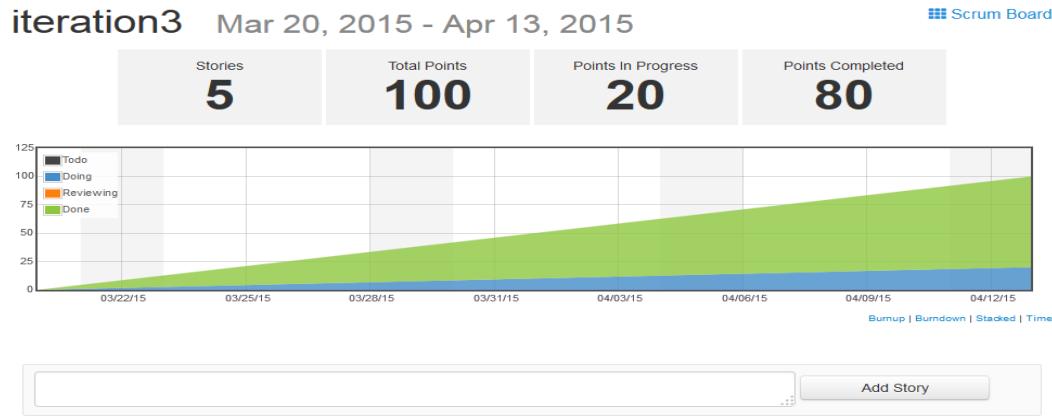
Developing customized web services all at one place which provides feasible access to the events by user. The architecture explains each layer functionality starting from Login page to data layer.



By using the openly available API to connect to end point and get the JSON response all the events; transportation events, university events and Live events can be viewed.

### **Scrum Do and Stories:**

For effective product development, an iterative and incremental methodology is needed. Agile product development methodology is present flexible product development strategy involving many various users as team workers working on unique and checking their progress simultaneously. Self-organizing makes agile process an encouraging toward a good product development irrespective of location and physical presence of the team member. Scrum do uses this agile methodology by adding stories and priority is set in the backlog. Backlog changes the course of the product development and it is not complete initially. In each iteration consists of stories to be done. After the stories are resolved then the iteration is closed and it get directed to next iteration. If any unfinished stories are present then they are directed to next iteration and further resolved. For an iteration the stories which are having priority are added from the backlog. So during this process the total progress is measured and overall project is accessed. The contribution of each team member is also visible which makes a clarity review of a team. The below screen shots depicts the iterations and dashboard of our Scrum Do. The dynamic flow represents the flexibility of the project.

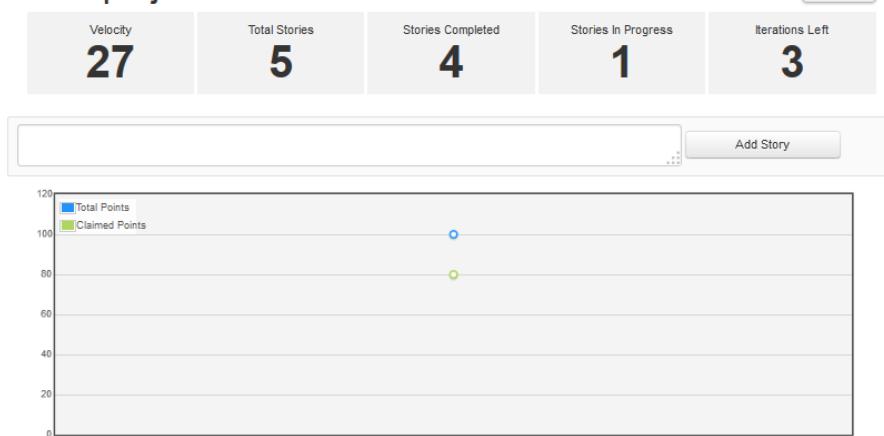


## Iterations

	Name	Stories	Start	End
☰ ⚙	Backlog	0 stories		
☰ ⚙	Iteration1	0 stories	Feb 10, 2015	Feb 25, 2015
☰ ⚙	iteration2	0 stories	Mar 01, 2015	Mar 18, 2015
☰ ⚙	iteration3	5 stories	Mar 20, 2015	Apr 13, 2015
☰ ⚙	iteration4	0 stories	Apr 15, 2015	Apr 29, 2015
☰ ⚙	Project Poster Presentation and final report	0 stories	May 01, 2015	May 08, 2015

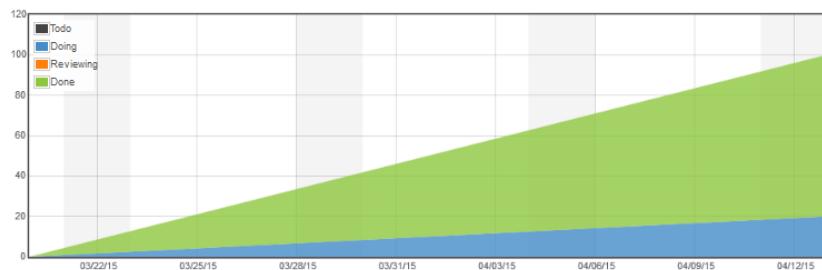
+ New iteration

## ASE project



## iteration3 Mar 20, 2015 - Apr 13, 2015

[View Board](#) | [View Story List](#)



Scrum Log

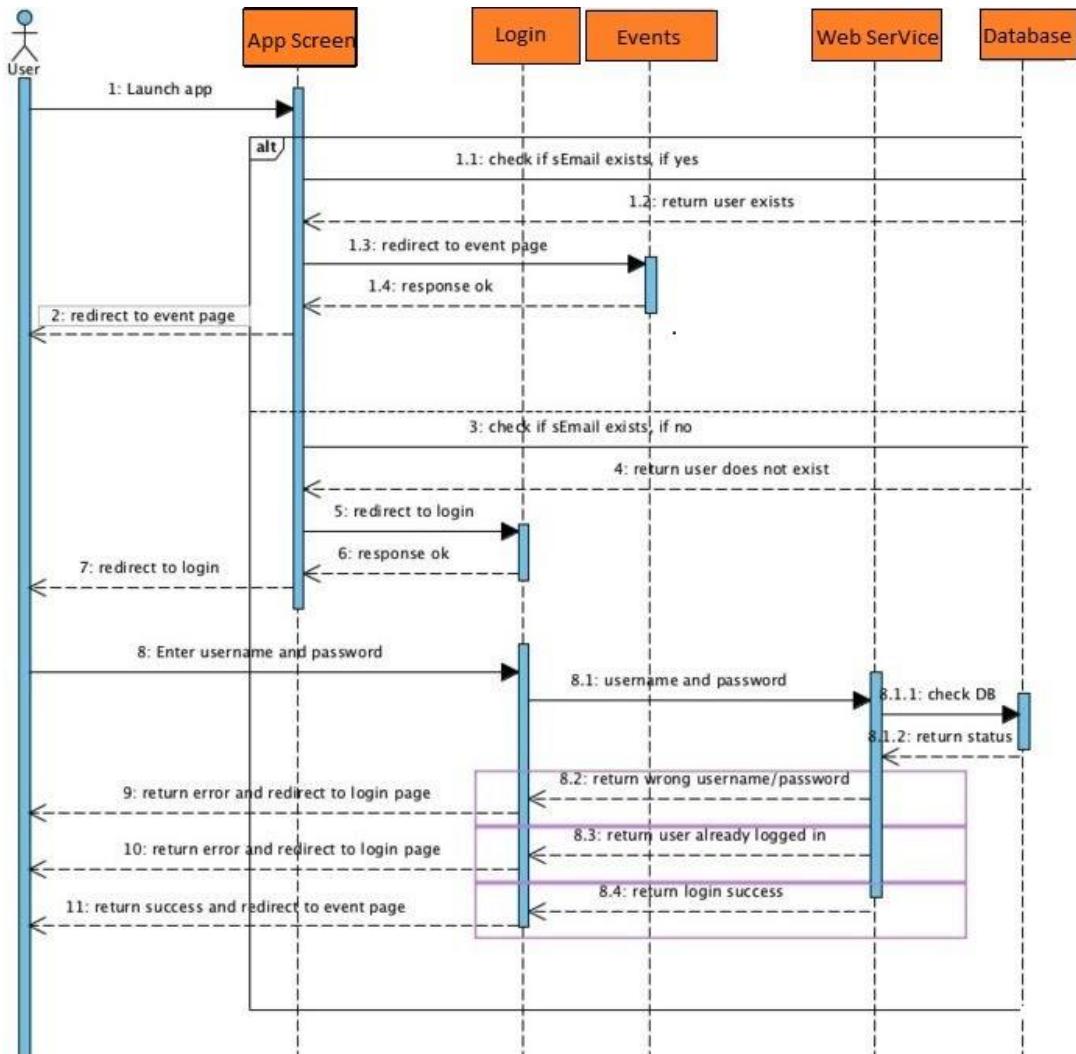
Newsfeed

Filter

21 minutes ago varaprasad92 modified story #5 perform performance testing in ASE project / iteration3  
o Status: Todo -> Doing

22 minutes ago varaprasad92 created story #5 in ASE project / iteration3.

## Sequence Diagrams:

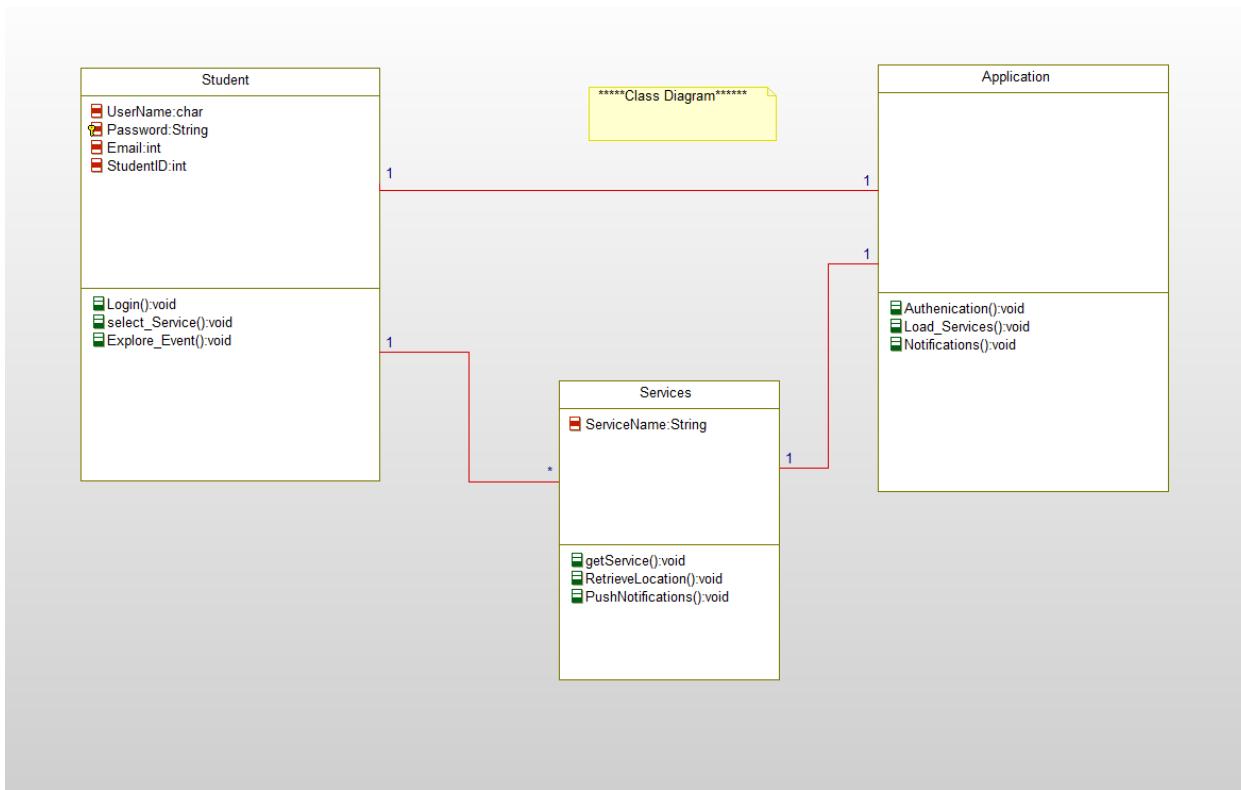


Message interaction chart diagrams which are sometimes called event diagrams are sequence diagrams explaining how the ordering of processes is done and process they do. In the diagram we consider objects and classes that interact in time sequence. The vertical lines are life lines that objects live on time sequence. Horizontal lines depicts the message interactions between the objects that carried in the process. The above sequence diagram is abstract level representation of

services and flow of them. In above diagrams, the message interaction between objects exhibiting the role of student, services and System is shown. First the student registers the app and validation of details mentioned by the user is made by checking out the database present already regarding the user enrolled. At the database a query is run to check whether the student is registered beforehand or need to create a new account. If old user, displaces the services to the user window. If new user, an account created and then displaces the services to the user window. This is abstract level of communication between the objects Student, system and Services.

### **Class Diagram:**

Structured representation of classes, attributes and methods (or Operations) which involved in a process are implemented using class diagram.



Here are the classes which we implemented in our project.

#### **Student Class:**

This class is primarily associated with students to register with the app by providing his/her login credentials. If the user is already registered then the application prompts the user to login in to the app or else a new login id is created. After logging the user can select the services that are in the app.

#### **Service Class:**

This class consists of all the services that we implemented in the application. The primary attribute of this class is a service name and the operations are to push and retrieve the data.

Application Class:

This class authenticates the users with the help of API's. Along with that this class is responsible for loading the services requested by the user.

## IMPLEMENTATION:

Implementation of REST Services:

Application authorization and registration is done using two API's

Login API and Registration API

Login

API:

This API does accept input from the end user and validates them for authentic login to avoid fraud attempts to enter into application

As part of storing the user details we created database table for registration purpose which can be used while user attempts login.

dbo.Table4: Tabl...\\APP_DATA\\DB.MDF		
	username	password
	pavan	pavan
	pk	pk
▶	bollarampavan...	bpavan
	pbk	pbk
	pvn	pvn
	pbc6d@umkc	pavankumar
	preetham@gmail	preetham
	sm8xd@umkc	srikanroanoke
	google@gmail	google

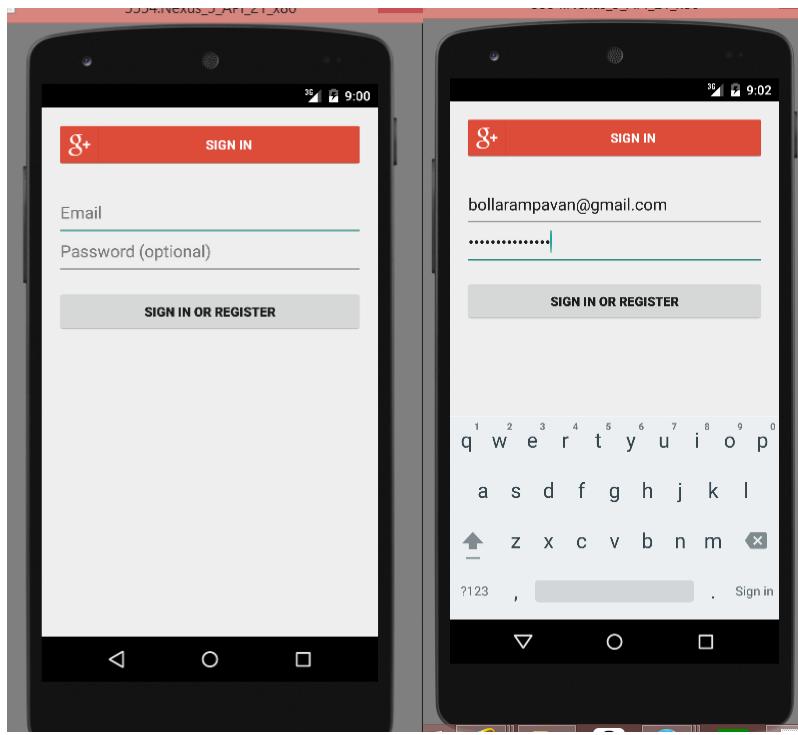
Registration API:

Registration API deals with registration of end user into application as of now we are accepting email id as primary attribute and password as second.

End user with email id and password which were given at the moment of registration can login into application as mentioned above.

## Implementation of User Interface (Mobile Interface):

### Screen I: Login Screen



Login Screen

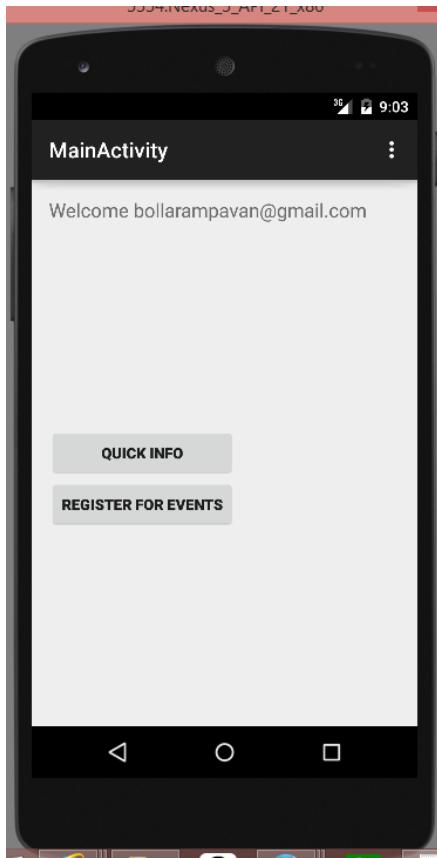
A user can login to system using existing google+ sign in services so that one can easily login with his/her Gmail id's which reduces the use of separate database for our application .This login screen contains two separate parts which are

**Sign in and Register:** This button used incase user is already registered with our application. He can directly go into access of further screens. Or else if he is not registered already it generates into new screen showing you are not registered / wrong password; login again.

### Email and Password:

These fields which takes input from user and validates the user details which are already into the database when registered. Once validated user can login to application it pops into his dashboard.

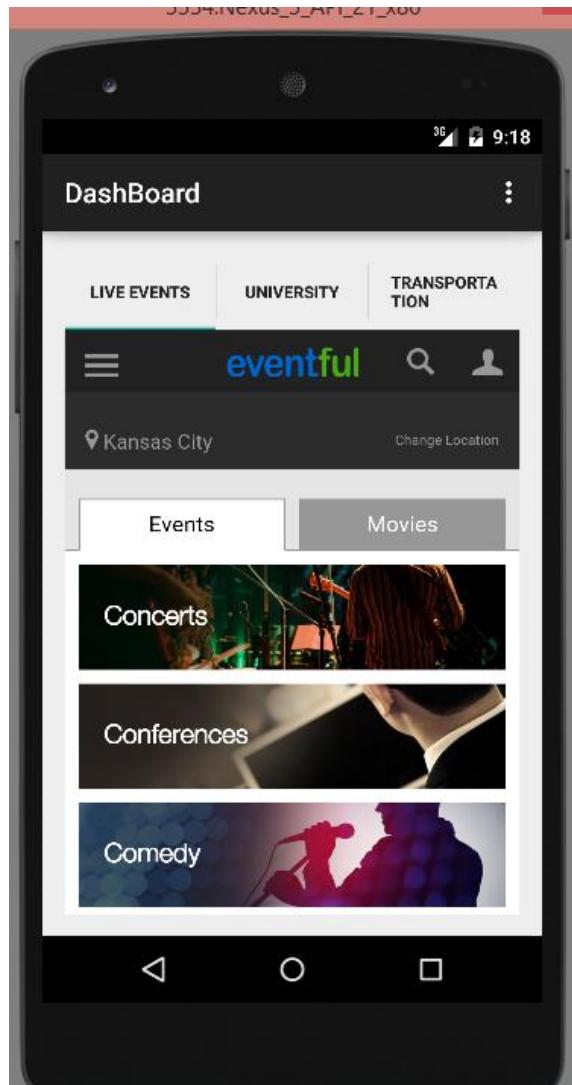
## Welcome Page (Screen II)



### Welcome Page:

The above figure depicts the welcome page displayed to the user when he / she login into the application. We included two Events buttons **Quick-Info** and **Register for Events**. On click of Quick Info, the button navigates into dashboard screen which is categorized into Live-Events, University and Transportation.

### Dashboard (Screen III)



Live Event Tab.



On Click of Transportation Tab.

## DASHBOARD

### Dashboard:

This is dashboard screen which will be displayed when user press the Quick info button in Screen II. It is completely mashup application where user can easily check the required tab.

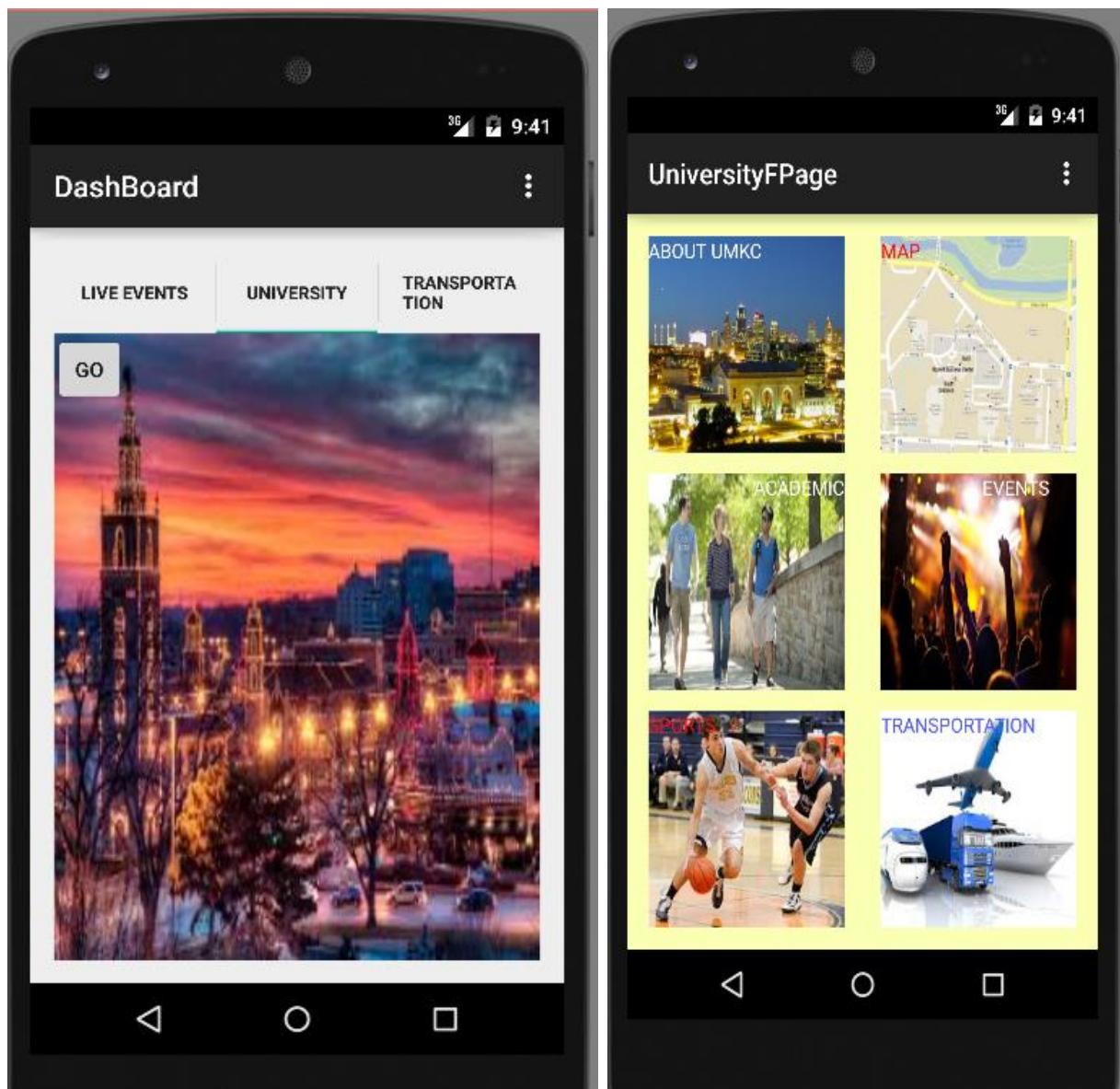
We simplified GUI with simple tab host concept so that end user can easily navigate between Live-events, University and Transportation tabs. The content is displayed according to the selection of tabs.

## University Dash Board(Screen IV)

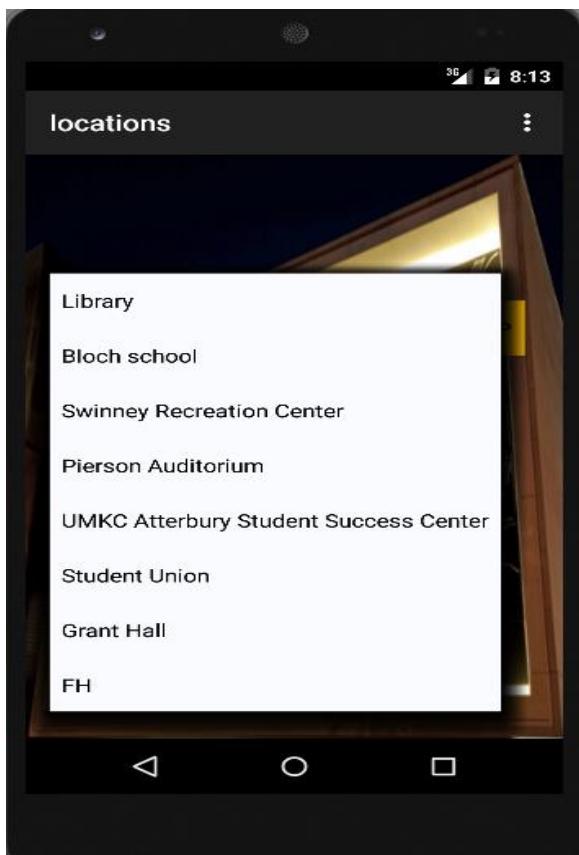
University details are presented to the user with tab in the Dashboard and when clicked navigated to different screen.

This screen consists of all data regarding university and it can be further improvised using deeper user interface

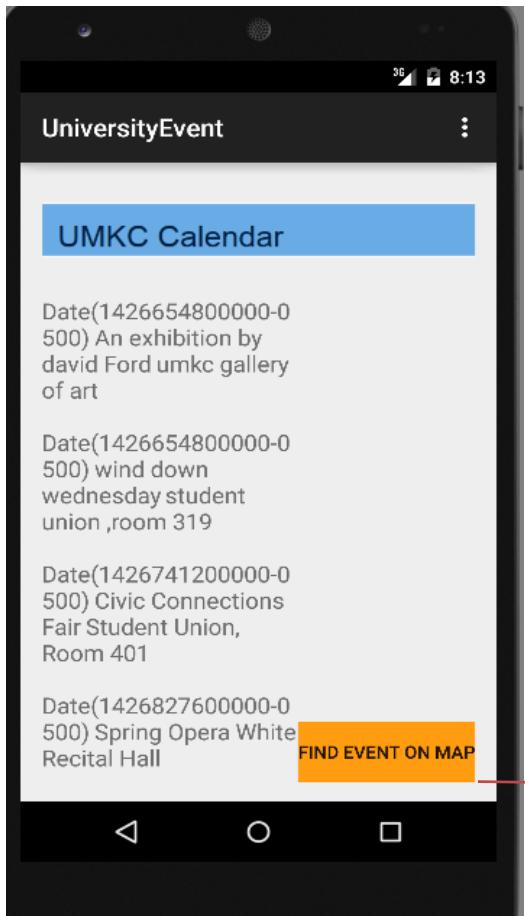
As of now we included few of the basic required data sections across screen they are as shown in the following images



The main dashboard screen consists of data sections like About UMKC, Map, Transportation, Academic details and sports sections.

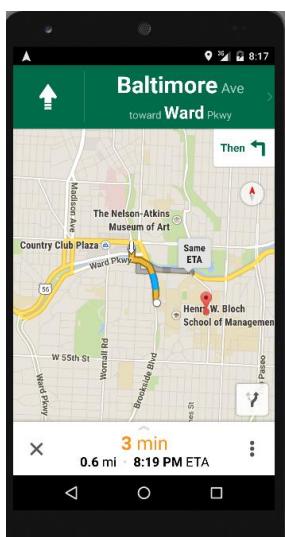
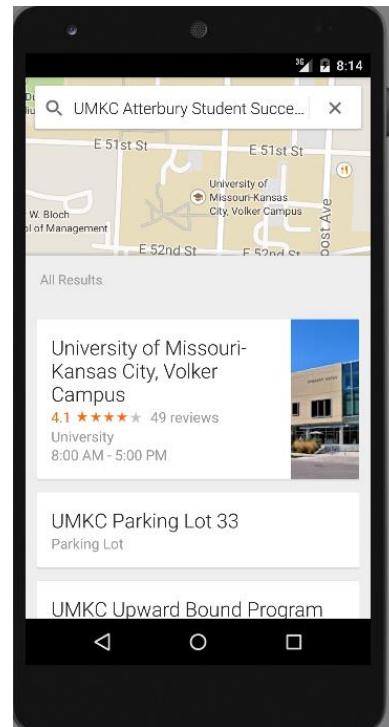


Locations page displays the event locations around university



### UMKC event calendar

Displays event names and their timings across all locations in UMKC



Google maps are integrated into application to display user the routes and available navigation options.

## Database Design implementation

Database tables are created to store all the data required by web API

dbo.Table4: Tabl...\\APP_DATA\\DB.MDF				Table4: Query(pv...)		
	date	event	location	bus number	route	time
▶	3/18/2015	An exhibition b...	umkc gallery of...	155	plaza-blueridge	07:40:00
	3/18/2015	wind down we...	student union ,...	MAX	downtown-51st	08:00:00
	3/19/2015	Civic Connecti...	Student Union, ...	155	blueridge-plaza	08:20:00
	3/20/2015	Spring Opera	White Recital H...	MAX	51st-downtown	08:40:00
	4/1/2015	Denim Day Drive	105 Haag Hall	155	plaza-blueridge	08:40:00
	4/1/2015	Sarah Frisof, flu...	Grant Recital Hall	MAX	downtown-51st	09:00:00
	4/2/2015	UMKC Health S...	Pierson Auditor...	MAX	51st-downtown	09:10:00
	4/3/2015	New Historical ...	Cockefair Hall 2...	155	blueridge-plaza	09:20:00
	4/3/2015	Clean Commut...	Swinney Recrea...	MAX	downtown-51st	09:30:00
				MAX	51st-downtown	09:40:00
				155	plaza-blueridge	10:00:00

Created database to store live events happening around UMKC and web service is implemented to fetch these events

sportstest: Query...\\APP_DATA\\DB.MDF		Sports: Query(pv...\\APP_DATA\\DB.M		
	Date	Time	Event	Location
▶	2015-04-14	05:30:00.0000000	Intramural Socc...	Durwood Socce...
	2015-04-14	17:30:00.0000000	Intramural Soft...	Durwood Socce...
	2015-04-15	17:30:00.0000000	Intramural Kick...	Durwood Socce...
	2015-04-17	17:30:00.0000000	Intramural Socc...	Durwood Socce...
	2015-04-18	09:00:00.0000000	Discover SCUBA	Swinney Recrea...
	2015-04-21	17:30:00.0000000	Intramural Socc...	Durwood Socce...
	2015-04-24	17:00:00.0000000	Agape Hoops Y...	Swinney Recrea...
	2015-04-24	17:30:00.0000000	Intramural Socc...	Durwood Socce...
	2015-04-28	17:30:00.0000000	Intramural Socc...	Durwood Socce...
	2015-05-01	17:00:00.0000000	Agape Hoops Y...	Swinney Recrea...
	2015-05-01	17:30:00.0000000	Intramural Socc...	Durwood Socce...
	2015-05-02	09:00:00.0000000	Pool Parties	Swinney Recrea...
	2015-05-07	17:30:00.0000000	Intramural Kick...	Durwood Socce...
	2015-05-12	09:00:00.0000000	Clean Commut...	Swinney Recrea...
	2015-05-27	16:00:00.0000000	'Lacrosse Assoc...	Swinney Recrea...

Table to store sports events data

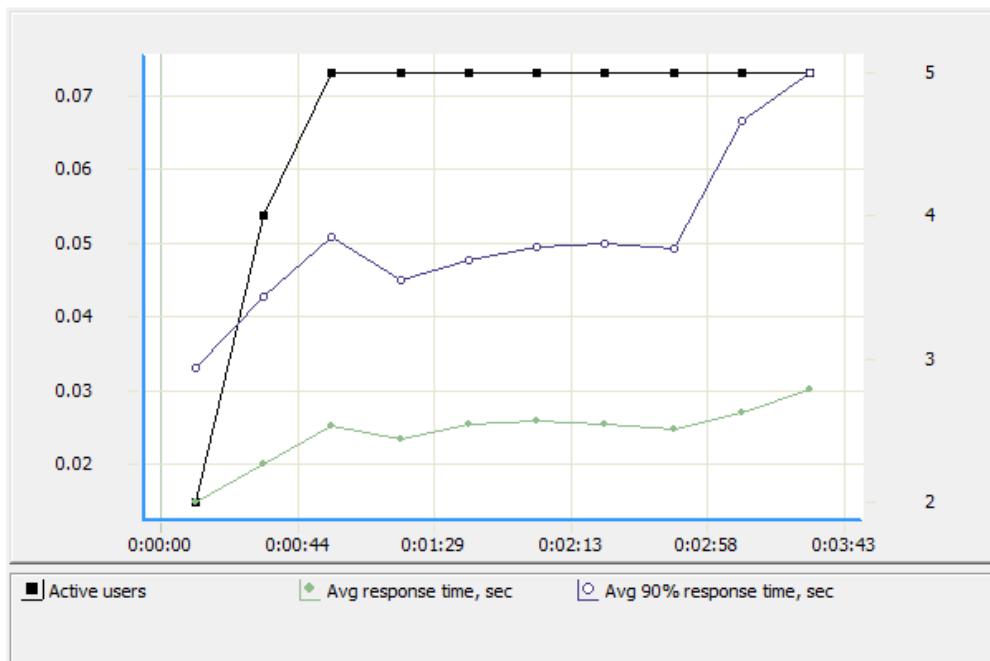
## TESTING:

WAPT is a load and stress testing tool that provides effective way to test a web page, web portal etc., We performed WAPT on web service on transportation. Here are the screenshots of testing using **WAPT** on link : [http://kc-sce-cs551.kc.umkc.edu/aspnet\\_client/Group3/Transportation/Service1.svc/transportation/details](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details)

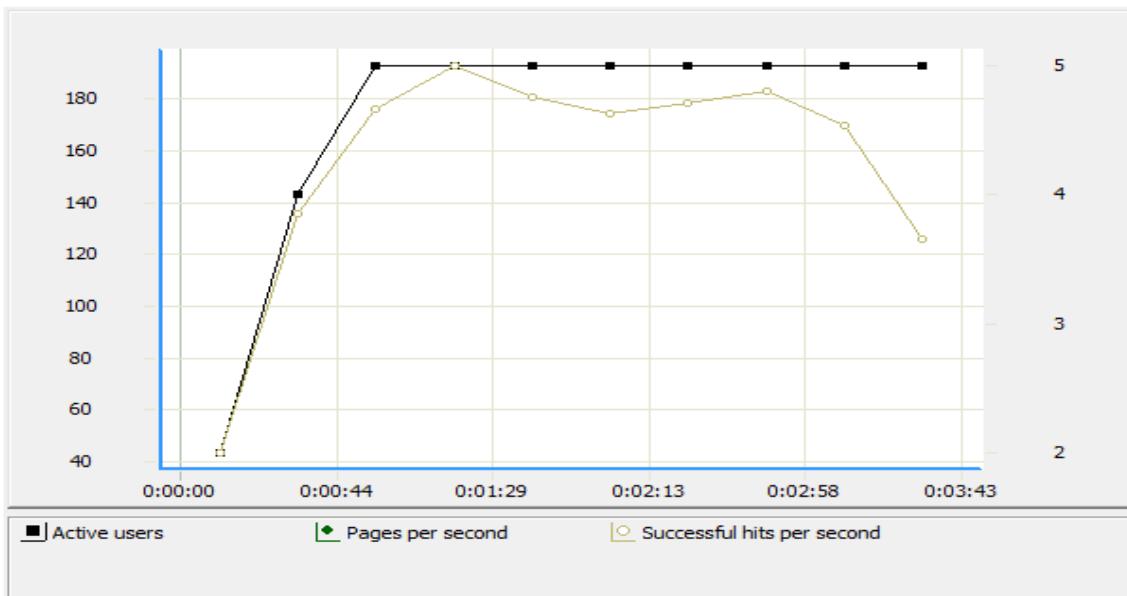
Response time, sec (with page elements)									
Name	Time	0:00:00-0:00:24	0:00:24-0:00:48	0:00:48-0:01:12	0:01:12-0:01:30	0:01:30-0:01:54	0:01:54-0:02:18	0:02:18-0:02:36	0:02:36-0:03:00
Profile1.page_1: http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details		1042	3259	4221	3470	4330	4189	3213	439
	Min	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
	Max	0.13(0.13)	0.25(0.25)	0.22(0.22)	0.14(0.14)	0.13(0.13)	0.14(0.14)	0.17(0.17)	0.17(0.17)
	Avg	0.01(0.01)	0.02(0.02)	0.03(0.03)	0.02(0.02)	0.03(0.03)	0.03(0.03)	0.03(0.03)	0.02(0.02)
	Avg90	0.03(0.03)	0.04(0.04)	0.05(0.05)	0.05(0.05)	0.05(0.05)	0.05(0.05)	0.05(0.05)	0.05(0.05)

### Average response time (without page elements)

Profile1.page\_1: [http://kc-sce-cs551.kc.umkc.edu/aspnet\\_client/Group3/Transportation/Service1.svc/transportation/details](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details)  
[http://kc-sce-cs551.kc.umkc.edu:80/aspnet\\_client/Group3/Transportation/Service1.svc/transportation/details](http://kc-sce-cs551.kc.umkc.edu:80/aspnet_client/Group3/Transportation/Service1.svc/transportation/details)



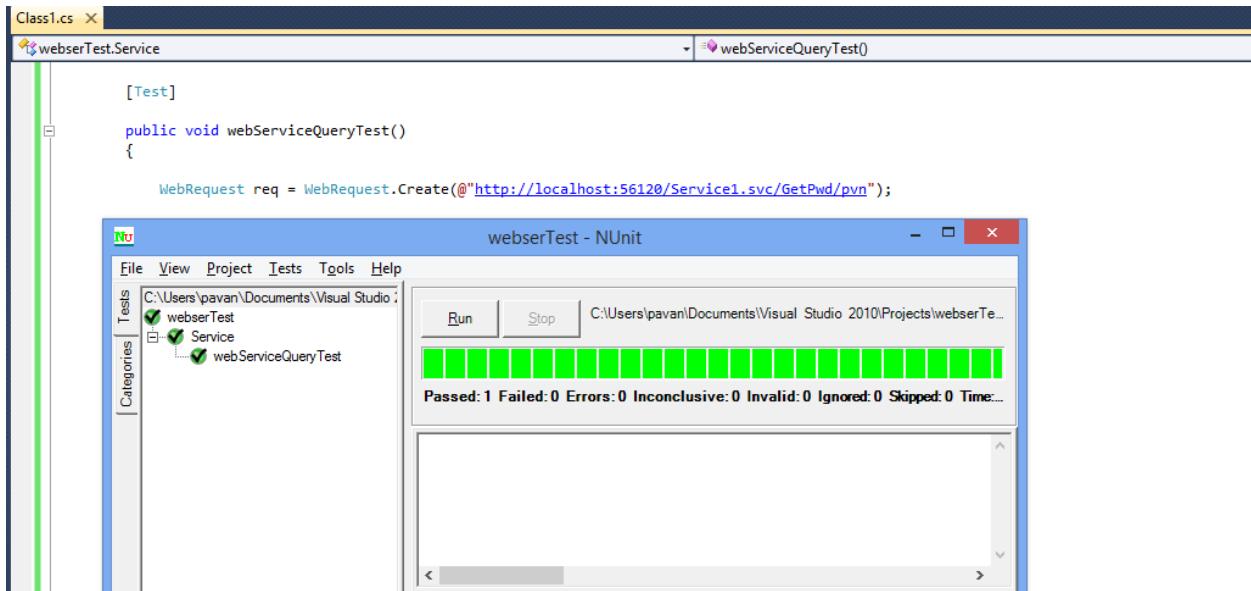
### Overall performance



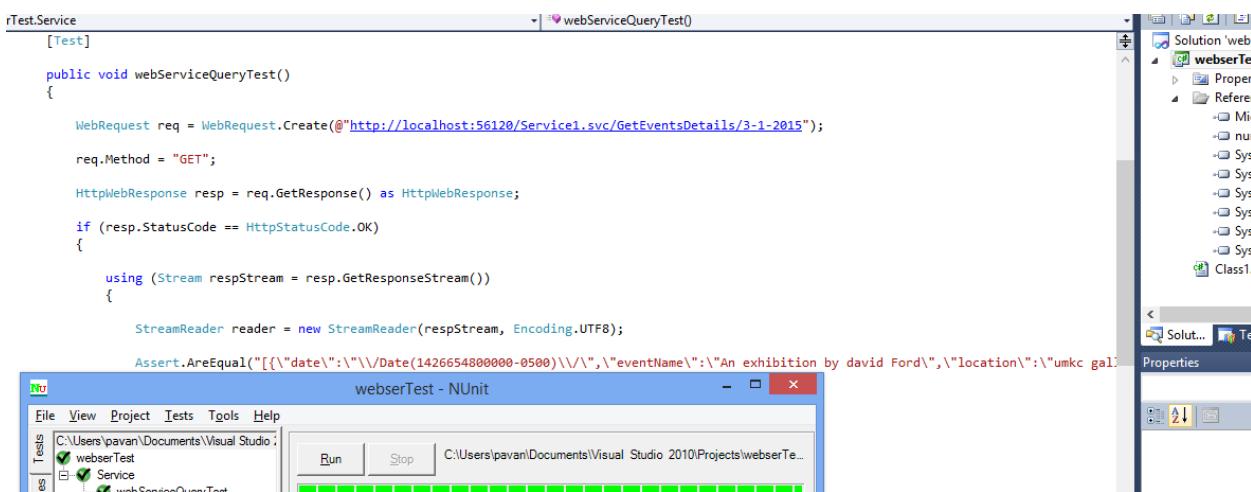
## NUnit TESTING

Tested web API's using NUnit testing tool which works on .NET platform and works same as Junit for Java. Successfully tested all the web service api's

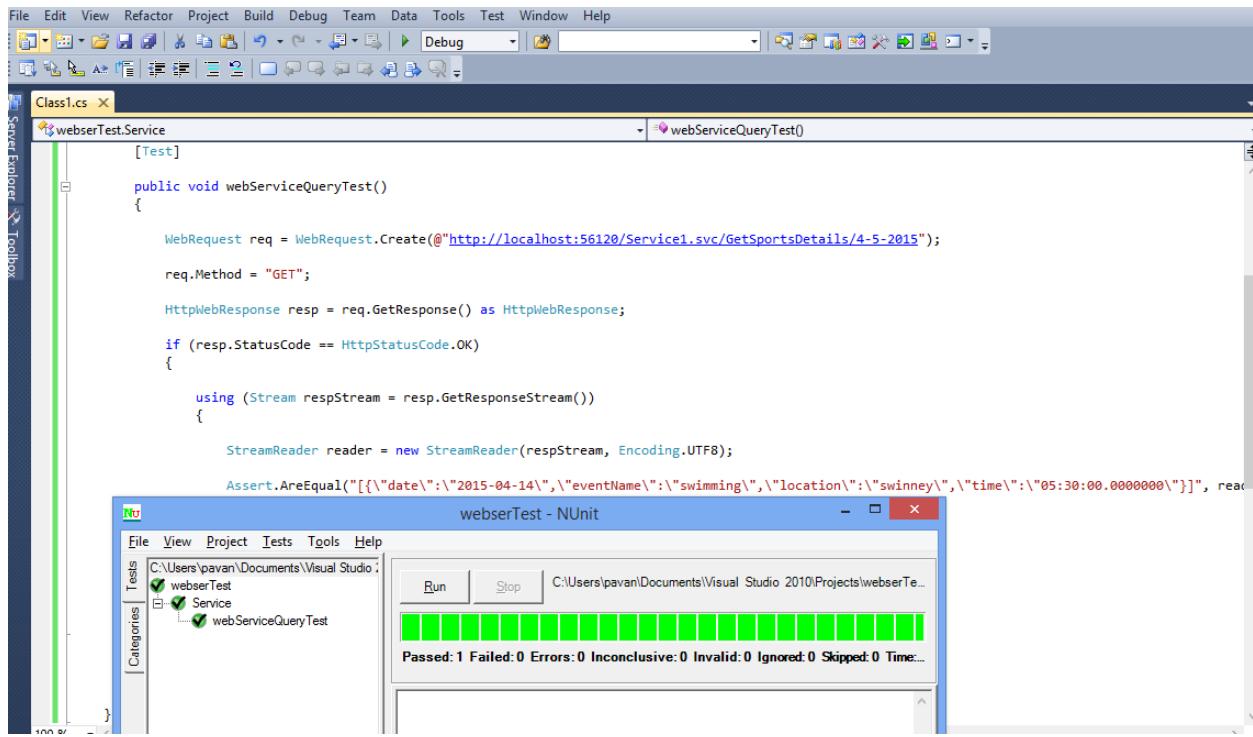
### 1. [http://localhost:56120/Service1.svc/GetPwd/](http://localhost:56120/Service1.svc/GetPwd)



### 2. <http://localhost:56120/Service1.svc/GetEventsDetails/3-1-2015>



### 3. <http://localhost:56120/Service1.svc/GetSportsDetails/4-5-2015>



## DEPLOYMENT:

Scrum do link:

<https://www.scrumdo.com/projects/project/ase-project8/summary>

GIT Hub Link:

Source code- <https://github.com/pavankumar-b/ASEspringSem/Increment3>

Documentation-<https://github.com/pavankumar-b/ASEspringSem/IncrementDoc3>

Work Completed:

## Description

- Design of Android Login Screen.

- Successfully implemented Login Functionality using web API and database. Validation of login users with already registered email and password or even when registered with Gmail.
- Created custom web API's to provide all the info regarding university
- Integrated Google maps engine into application so that all navigation related info can be available very easily
- Creating, validating and the database tables of user registered and while signing in.
- Tables for university live events and transportation tables were created.
- Tested application using NUnit testing tool
- Performance test of application was done

## **Responsibility**

Task 1: Implementation of web service / University Event API

Pavankumar Bollaram/ Varaprasad jaggu

Task 2: Implementation of web service / Transportation API

Lakshmi Priyanka / Preetham Kumar

Task 3: Implementation of web service / SportsAPI

Varaprasad jaggu / Preetham kumar /PavanKumar

Task 4: Database table creation

Pavan kumar Bollaram, varaprasad, preetham kumar, Priyanka

Task 4: User Interface (All screens)

Pavan Kumar, Preetham kumar, Varaprasad jaggu, Priyanka

Project management:

## **Time Taken**

300 Man Hours

## **Contribution**

Pavan Kumar Bollaram (25%)

Preetham Kumar Danaboina (25%)

Lakshmi Priyanka (25%)

Vara Prasad (25%)

### VIII. Risk management

1. working with Web API specially with visual studio was bit complex
2. Emulator runs poorly
3. Very less performance testing tools for android applications.
4. Data Tables needs to refreshed regularly to make up to date information in application

# **PROJECT INCREMENT 4**

## **Introduction**

GO EASY, is an innovative university mobile app usually helpful for the novice students who are attending a university at new place. Every student curious about knowing information of his/her university. This Application helps to find out University details, events in the university, transportation details and events around the city by using eventful API. Friendly user interface makes easy access to existing services. This project is done on android platform.

## **Objectives**

The main objective of the Application is to be handy to the new students of the UMKC. It provides the entire details of the university. It gives the details of the events going in the university during the current month with the location and the timing details. Coming to the additional feature. It provides all the live events going at a particular City.

## **Significance**

Although there are various web sites and applications specific to the school but we inferred from the apps that they do not cover all the information needed by the end user. So we planned to create a mobile based application uses existing API's and customized web services. We are proud to assure that our app serves a virtual guide to student to know about the university.

## **Technologies used**

- Java programming language
- .NET programming
- Android programming and UI design
- REST services
- Ajax and java script languages

## **API's and web services used**

The below services are consumed in our project:

- <http://api.eventful.com/rest/events/search?...&keywords=books&location=San+Diego&date=Future>

Everyone likes to be busy in their life. There will some instances he may be idle. At that time this API helps person to search around make his life delightful.

Eventful API acknowledges us about current live concerts, sports and exhibitions that are occurred in and around the selected city. The search is may be Event search

or Venue search. Event search may be lists of events by Keyword, performer, time, Category or location.

Venue search may be by location, name or type.

- <https://data.cityofchicago.org/resource/alternative-fuel-locations.json?>

Using geo-graphical location basis API is used to retrieve data from a specified location.

- <https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true>

Google API is used to explore location and view maps for a specified place. It is not single API but a set of API's developed by google which allows google services to meet navigation facilities from one place to another. Usage of google services should be familiar such as how to navigate and explore directions is suggested. We used this google API in our project to navigate to the place where the event sport is taking place. On click of the event specified in the sport events navigation to the sport event location.

### **Custom API's used in project**

- [http://kc-sce-cs551.kc.umkc.edu/aspnet\\_client/Group3/Transportation/Service1.svc/transportation/details](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details)
- <http://localhost:56120/Service1.svc/GetEventsDetails>
- <http://localhost:56120/Service1.svc/GetSportsDetails>

### **Detail Design of Services:**

Architecture: Whenever user tries to use the application he uses his credentials. It authenticated by the controller and then it hits the Web Service. It verifies with the Database. Once the user gets authenticated he can access the application. He can get the details of the live events going in the Kansas city based on his/her interest he can search the events. Those details were retrieved from the Kansas city API. Coming to the University details. Once we hit the specific the details it will directly hit the local API and retrieve the details.

### User Interface:

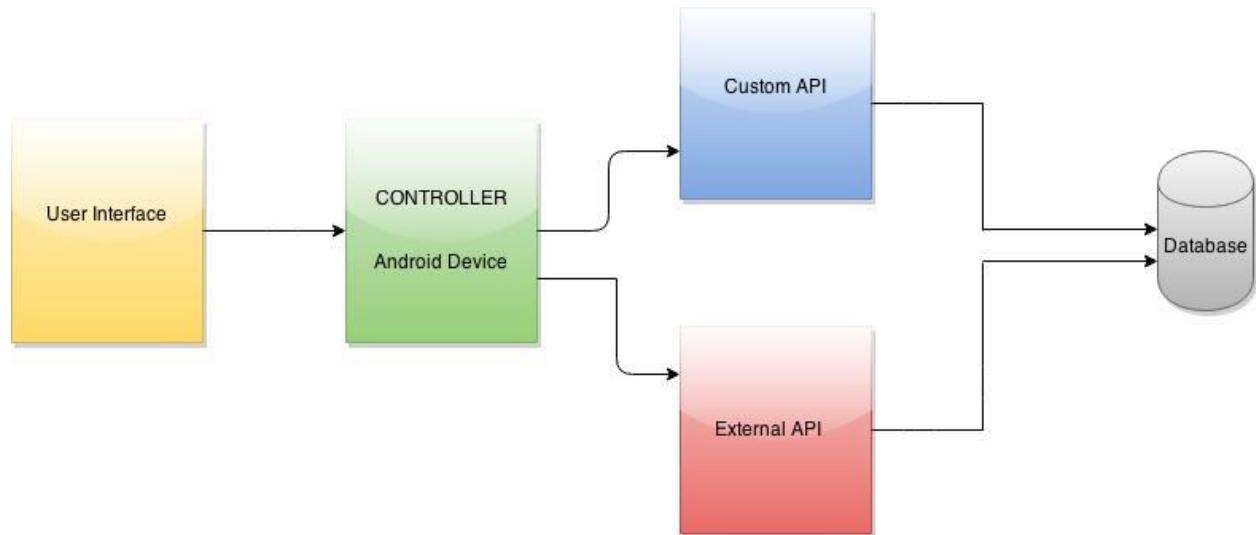
User interface provides feasible access to all events included in the android screen. Firstly, student signs in or sign up at login page instead with Gmail account student can login. Then, it re-directs after verification from database of form inputs given by users to login successfully. Secondly, student can select from the tab view. For suppose, if student selects the event tab it connects to available Eventful API and gets the response as events based on location, time-based, performer or by keyword related to event. Next, if student selects the university details, it transits to new web view consisting of image views of six categories i.e., Sports, Academics, Transportation, About University, Events and Maps. By using the available API and customized API to connect to end point and get the JSON response of all events. Then end user can sign out or may continue with it.

### Controller (Android Device):

Android platform is most widely used open source platform in the world and most effective user interfaces can be created by android platform. By the using real time actions such as tapping, swiping, sliding on mobile screen it re directs to new page or results in an action we intend to get. Android device also helps to connect to database, custom API and external API effectively and performance of loading is at high level.

### Database:

In this project we created tables for customized web services. We created tables for events, transportation, sports, login and registration. By using this tables it is easily retrieve or store the details accordingly needed for application.



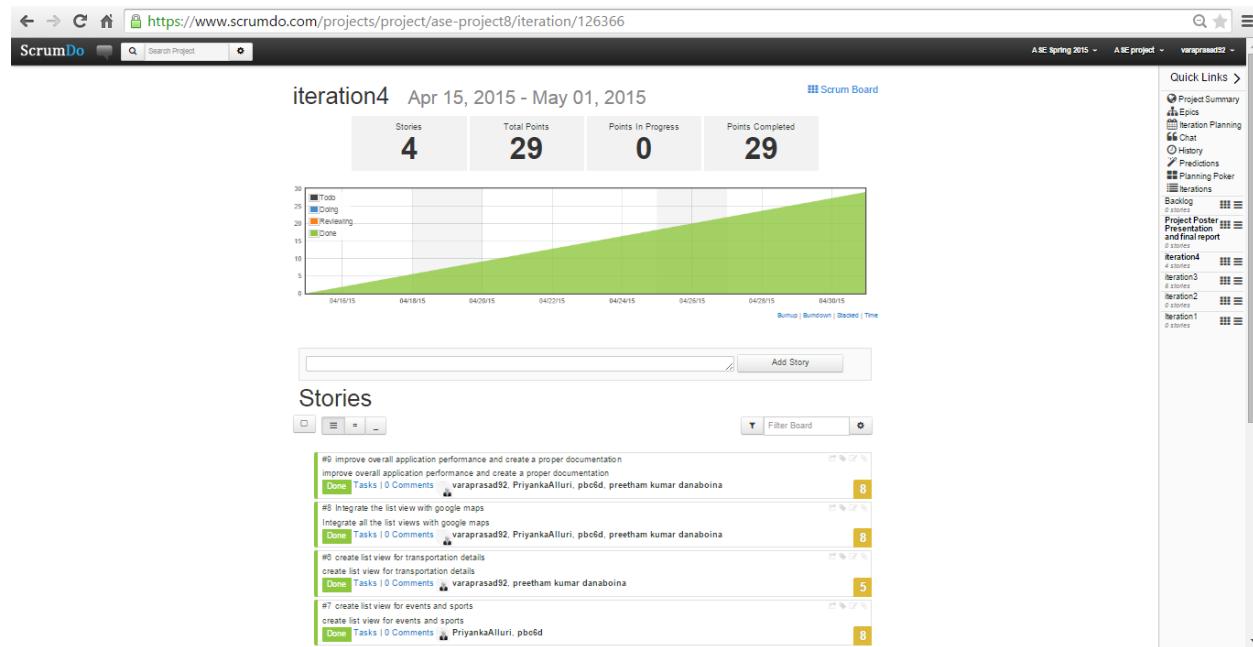
**Architecture**

## Scrum Do and Stories:

Effective product development is possible only when there is incremental and iterative methodology. This can be done by using Scrum do which uses agile methodology by setting priority in backlog and adding stories. Iteration consists of stories to be done. If all the stories in the iteration is done the iteration is closed and new iteration can be done. This iteration 4 includes:

- Creation of list view interface for transportation, events and sports.
- Integration of google maps API to display locations of university.
- Implementation of a common interface to search a location on maps.
- Testing application with fire bug and Yslow analyzer.
- Improving the User Interface.

The below figures are from the scrum do which describes the dynamic flow of iteration 4.



[https://www.scrumdo.com/projects/project/ase-project8/all\\_iterations](https://www.scrumdo.com/projects/project/ase-project8/all_iterations)

ScrumDo  ASE Spring 2015 ASE project varaprasad92

## Iterations

	Name	Stories	Start	End
Backlog	Iteration1	0 stories	Feb 10, 2015	Feb 25, 2015
	iteration2	0 stories	Mar 01, 2015	Mar 18, 2015
	iteration3	5 stories	Mar 20, 2015	Apr 13, 2015
	iteration4	4 stories	Apr 15, 2015	May 01, 2015
	Project Poster Presentation and final report	0 stories	May 01, 2015	May 08, 2015

Quick Links >

- Project Summary
- Epics
- Iteration Planning
- Chat
- History
- Predictions
- Planning Poker
- Iterations
- Backlog
- Project Poster Presentation and final report
- iteration4
- iteration3
- iteration2
- Iteration1

<https://www.scrumdo.com/projects/project/ase-project8/planning#iterationPlanning>

ScrumDo  ASE Spring 2015 ASE project varaprasad92

## ASE project

Project Velocity 32 Points

Iteration	Stories	Points
iteration4	8	8
#9 improve overall application performance and create a proper documentation	8	8
improve overall application performance and create a proper documentation	8	8
Done Tasks   0 Comments	8	8
varaprasad92, PriyankaAlluri, pbc6d, preetham kumar danaboina	8	8
#8 Integrate the list view with google maps	8	8
Integrate all the list views with google maps	8	8
Done Tasks   0 Comments	8	8
varaprasad92, PriyankaAlluri, pbc6d, preetham kumar danaboina	8	8
#6 create list view for transportation details	5	5
create list view for transportation details	5	5
Done Tasks   0 Comments	5	5
varaprasad92, preetham kumar danaboina	5	5
#7 create list view for events and sports	8	8
create list view for events and sports	8	8
Done Tasks   0 Comments	8	8
PriyankaAlluri, pbc6d	8	8

4 Stories 29 Points 240:00 Hours

Quick Links >

- Project Summary
- Epics
- Iteration Planning
- Chat
- History
- Predictions
- Planning Poker
- Iterations
- Backlog
- Project Poster Presentation and final report
- iteration4
- iteration3
- iteration2
- Iteration1

<https://www.scrumdo.com/projects/project/ase-project8/summary>

ScrumDo

ASE project

Velocity: 32 | Total Stories: 9 | Stories Completed: 9 | Stories In Progress: 0 | Iterations Left: 4

Add Story

iteration4 Apr 15, 2015 - May 01, 2015

[View Board](#) | [View Story List](#)

Quick Links >

- Project Summary
- Epics
- Iteration Planning
- Chat
- History
- Predictions
- Planning Poker
- Iterations
- Backlog
- 0 stories
- Project Poster
- Presentation and final report
- 0 stories
- iteration4
- 4 stories
- iteration3
- 5 stories
- iteration2
- 0 stories
- iteration1
- 0 stories

<https://www.scrumdo.com/organization/ase-spring-2015/dashboard>

ScrumDo

ASE Spring 2015

Your Stats Organization Stats Point Breakdown Organization Velocity

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

Your Cards: 3 | Your Points: 21 | Projects: 1 | Stories Completed: 6

Tasks Completed: 0

ASE project

iteration4 Apr 15, 2015 - May 1, 2015

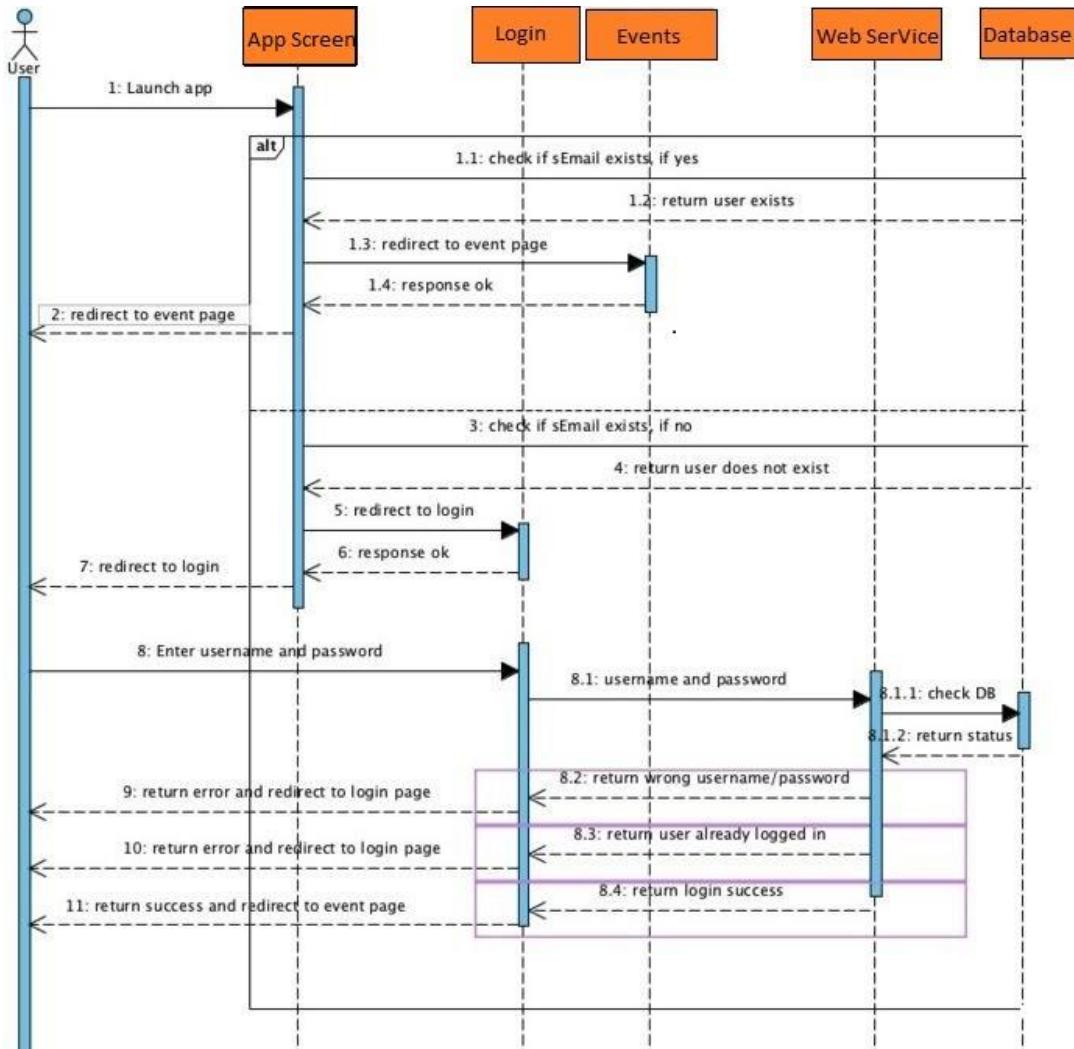
Your Stories:

- #6 create list view for transportation details
- #8 Integrate the list view with google maps
- #9 Improve overall application performance and create a proper documentation

Quick Links >

- Project Summary
- Epics
- Iteration Planning
- Chat
- History
- Predictions
- Planning Poker
- Iterations
- Backlog
- 0 stories
- Project Poster
- Presentation and final report
- 0 stories
- iteration4
- 4 stories
- iteration3
- 5 stories
- iteration2
- 0 stories
- iteration1
- 0 stories

## Sequence Diagrams:

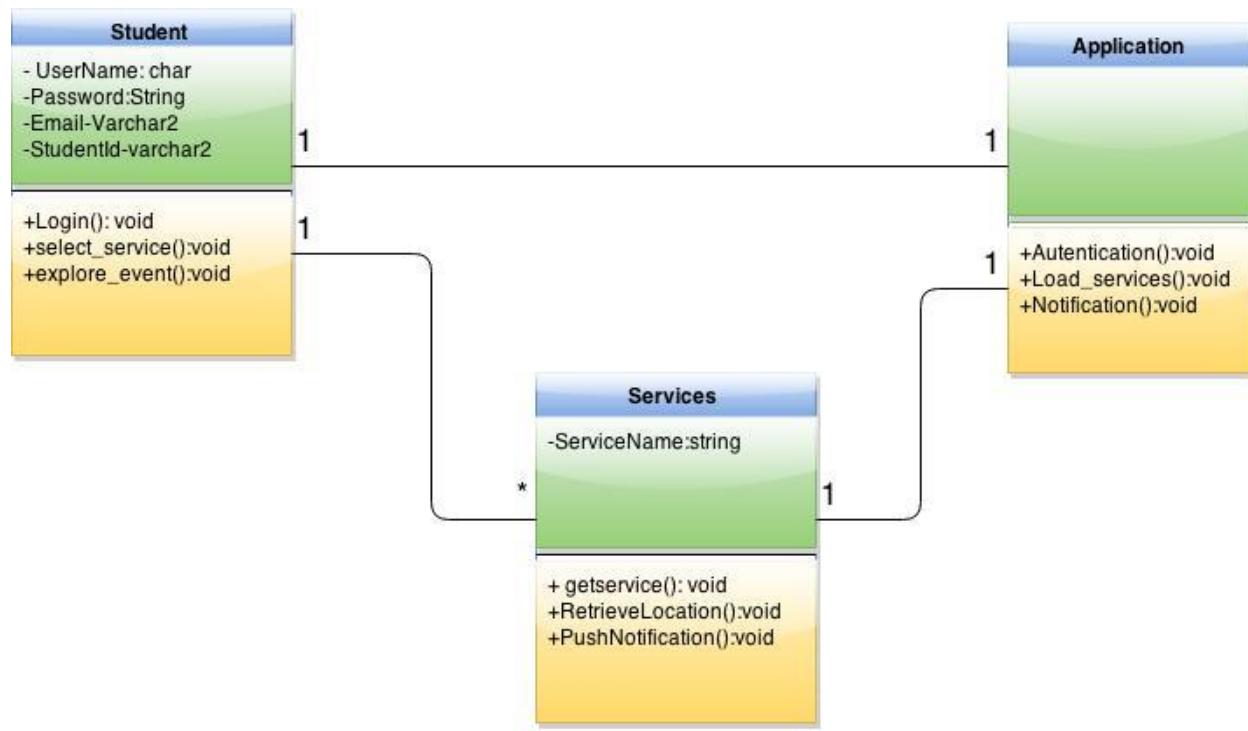


Message interaction chart diagrams which are sometimes called event diagrams are sequence diagrams explaining how the ordering of processes is done and process they do. In the diagram we consider objects and classes that interact in time sequence. The vertical lines are life lines that objects live on time sequence. Horizontal lines depicts the message interactions between the objects that carried in the process. The above sequence diagram is abstract level representation of services and flow of them. In above diagrams, the message interaction between objects exhibiting the role of student, services and System is shown. First the student registers the app and validation

of details mentioned by the user is made by checking out the database present already regarding the user enrolled. At the database a query is run to check whether the student is registered beforehand or need to create a new account. If old user, displaces the services to the user window. If new user, an account created and then displaces the services to the user window. This is abstract level of communication between the objects Student, system and Services.

### **Class Diagram:**

Structured representation of classes, attributes and methods (or Operations) which involved in a process are implemented using class diagram.



Here are the classes which we implemented in our project.

#### **Student Class:**

This class is primarily associated with students to register with the app by providing his/her login credentials. If the user is already registered then the application prompts the user to login in to the app or else a new login id is created. After logging the user can select the services that are in the app.

#### **Service Class:**

This class consists of all the services that we implemented in the application. The primary attribute of this class is a service name and the operations are to push and retrieve the data.

### **Application Class:**

This class authenticates the users with the help of API's. Along with that this class is responsible for loading the services requested by the user.

### **Implementation:**

Implementation of REST Services:

Implementation of two services Login API and Registration API

#### **Login-API:**

Validation of user inputs is main functionality check performed for registered user or not. Unique database table is created which separate database table is created which stores the registered details and does validations by fetching the details from this database table.

dbo.Table4: Tabl...\\APP_DATA\\DB.MDF		
	username	password
	pavan	pavan
	pk	pk
▶	bollarampavan...	bpavan
	pbk	pbk
	pvn	pvn
	pbc6d@umkc	pavankumar
	preetham@gmail	preetham
	sm8xd@umkc	srikarroanoke
	google@gmail	google

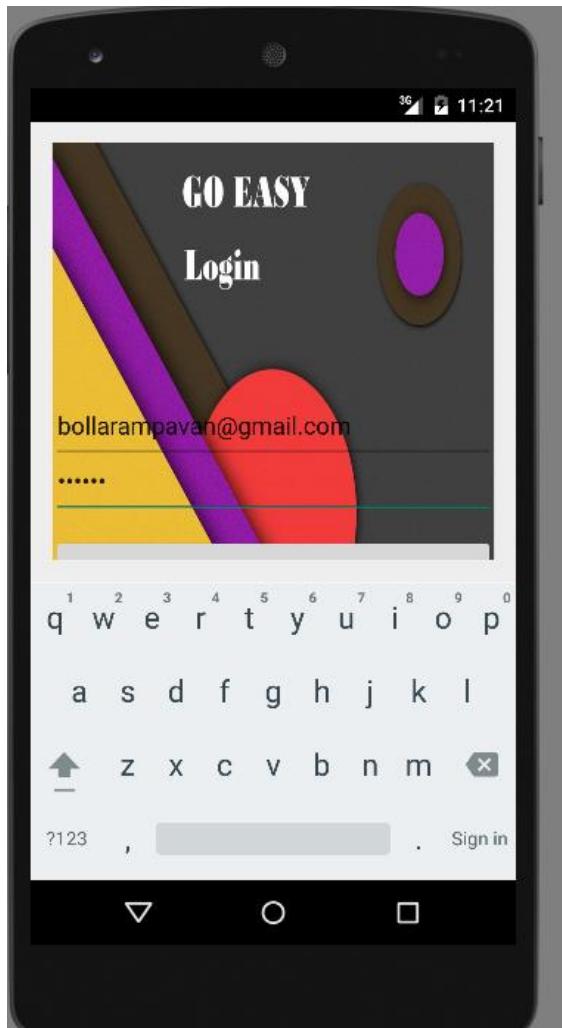
table created to store Login details and Login API connects to this table and fetches data.

#### **Registration API:**

Registration API deals with registration of end user into application as of now we are accepting email id as primary attribute and password as second. End user with email id and password which were given at the moment of registration can login into application.

Implementation of User Interface (Mobile Interface):

Screen I: Login Screen



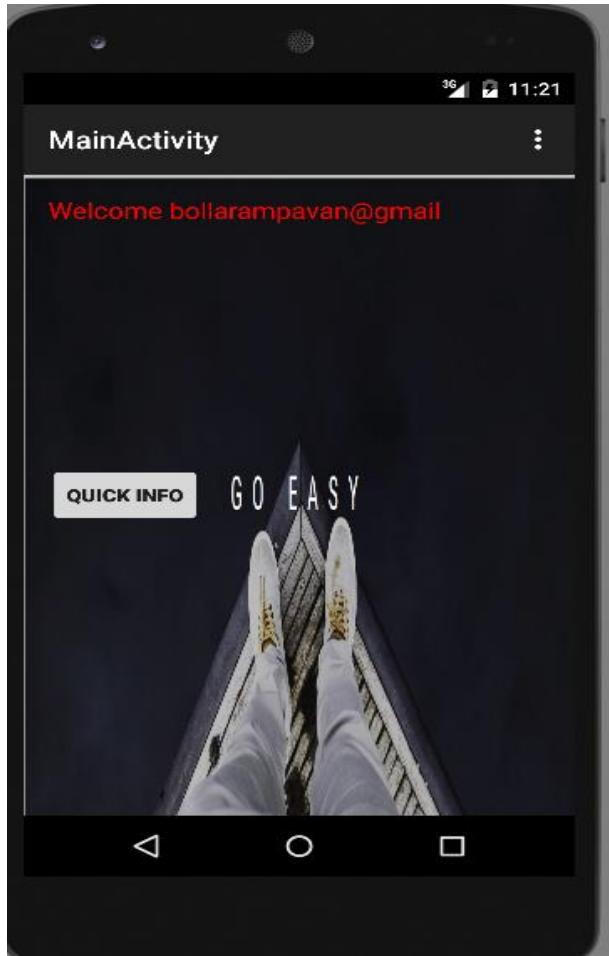
Login Screen

Login to system uses existing google+ sign in services so that end user can easily login with his/her Gmail id's which reduces the use of separate database for our application .This login screen contains two separate parts which are

**Sign in and Register:** This button used incase user is already registered with our application. He can directly go into access of further screens. Or else if he is not registered already it generates into new screen showing you are not registered / wrong password; login again.

**Email and Password:** These fields which takes input from user and validates the user details which are already into the database when registered. Once validated user can login to application it pops into his dashboard.

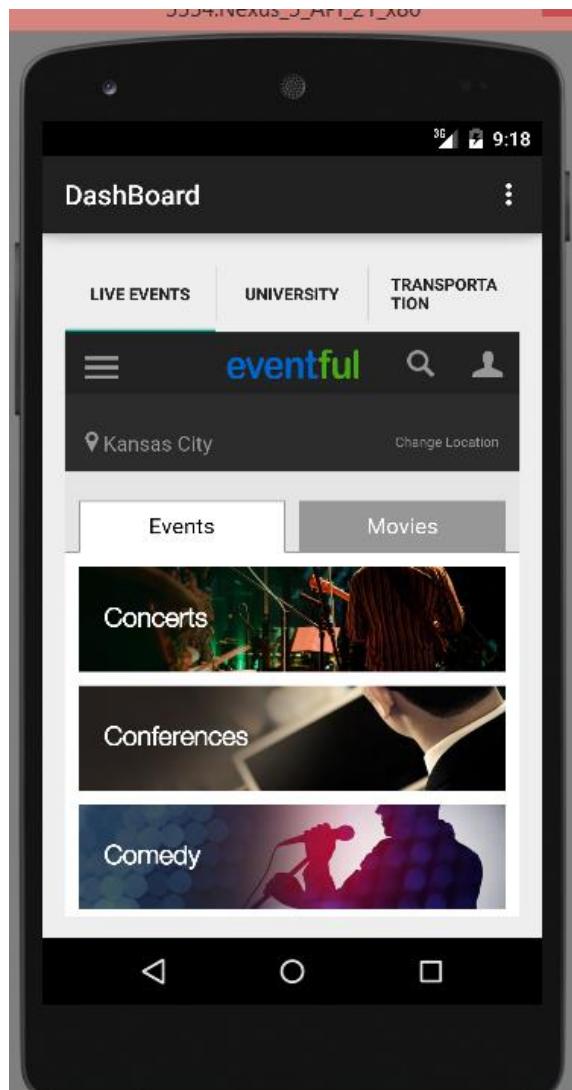
## Welcome Page (Screen II)



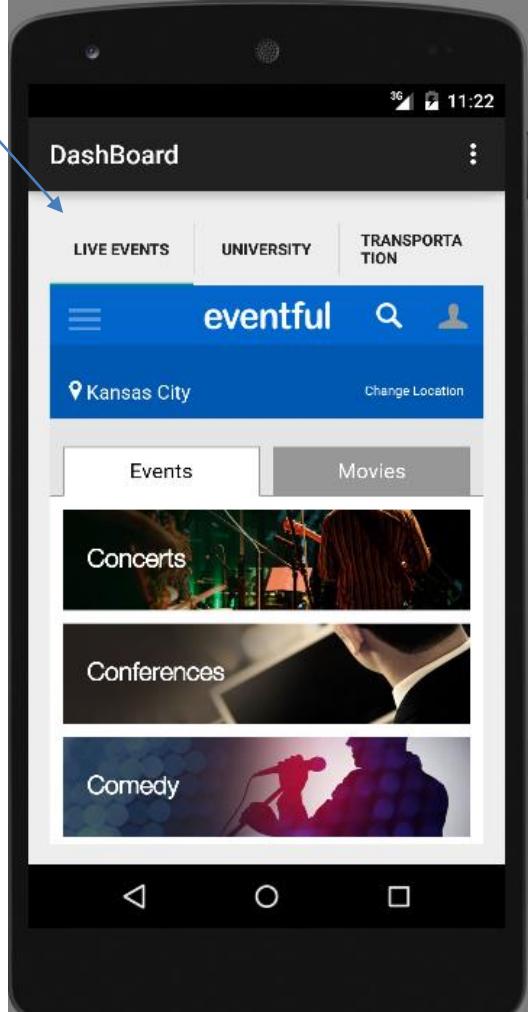
**Welcome Page:** The above figure depicts the welcome page displayed to the user when he / she login into the application. We included two Events buttons **Quick-Info**.

On click of Quick Info button page navigates to dashboard screen which is categorizes information on screen as **Live-Events, University and Transportation**.

## Dashboard (Screen III)



Live Event Tab.



## DASHBOARD

Dashboard: This is dashboard screen which will be displayed when user press the Quick info button in Screen II. It is completely mashup application where user can easily check the required tab.

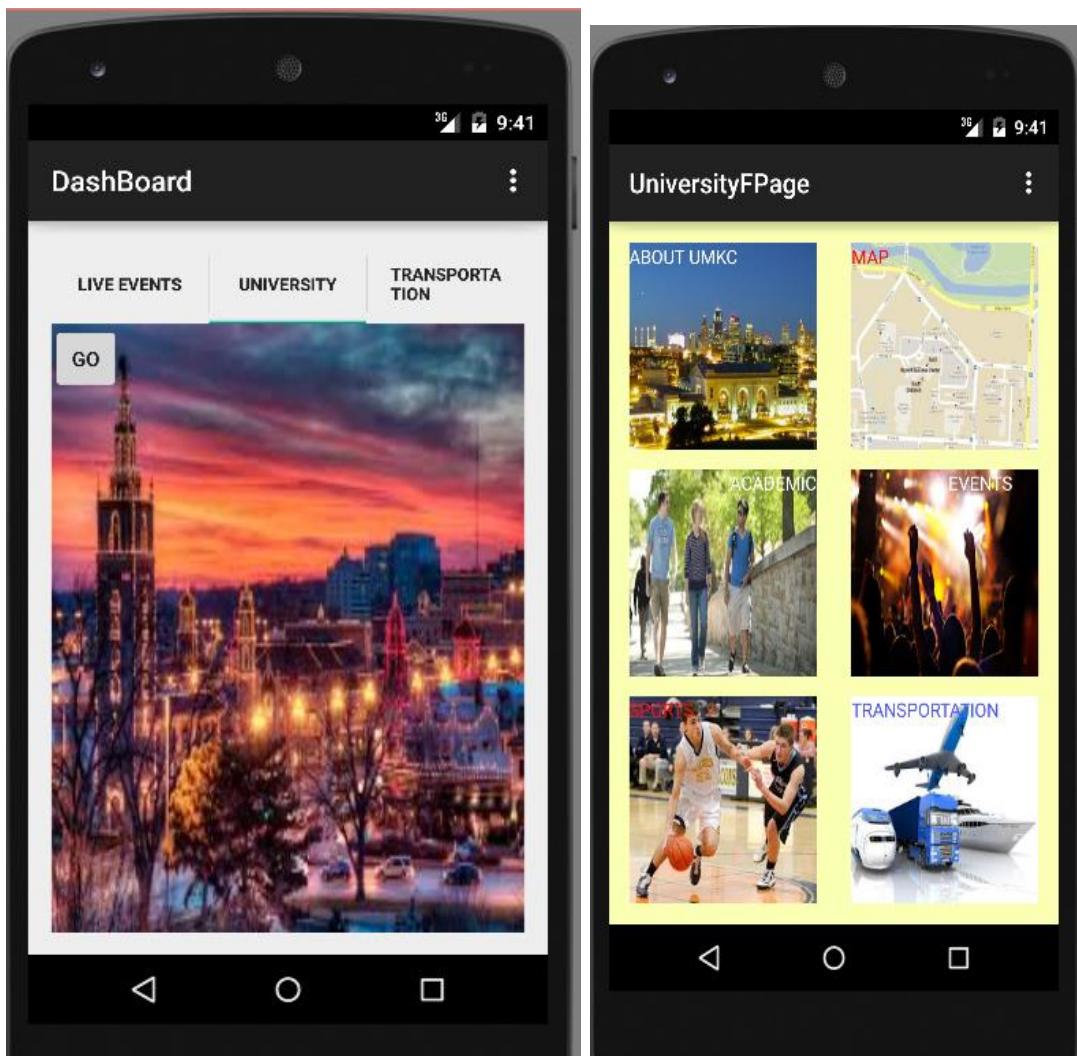
We simplified GUI with simple tab host concept so that end user can easily navigate between Live-events, University and Transportation tabs. The content is displayed according to the selection of tabs.

## University Screen (Screen IV)

University details are presented to the user with tab in the Dashboard and when clicked navigated to different screen.

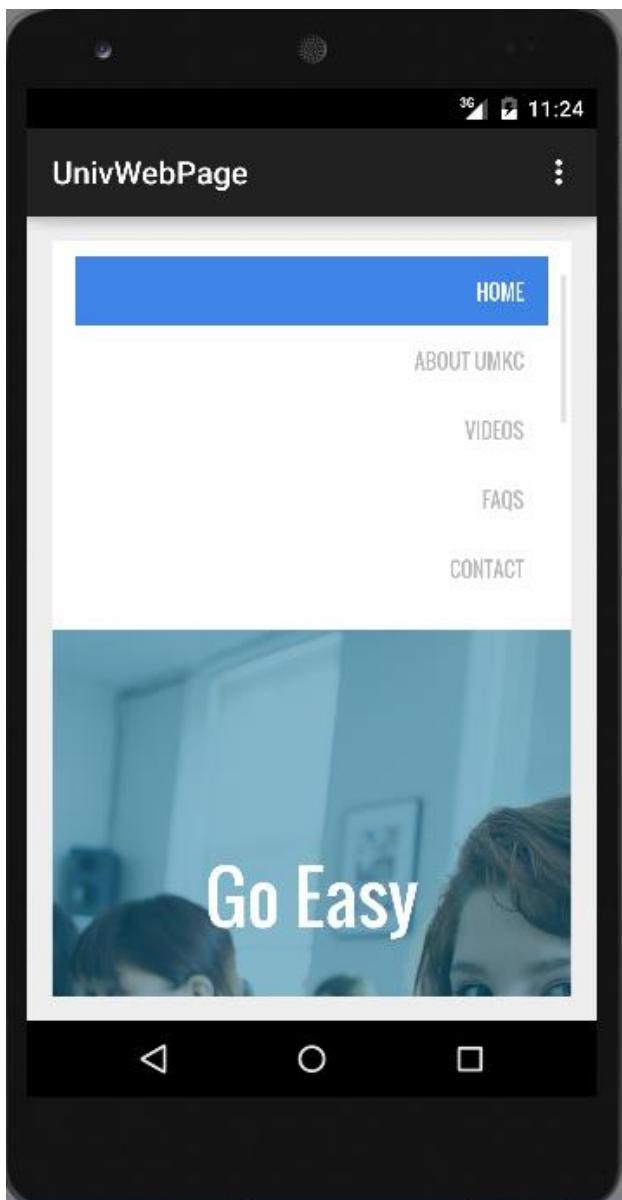
This screen consists of all data regarding university and it can be further improvised using deeper user interface

As of now we included few of the basic required data sections across screen they are as shown in the following images



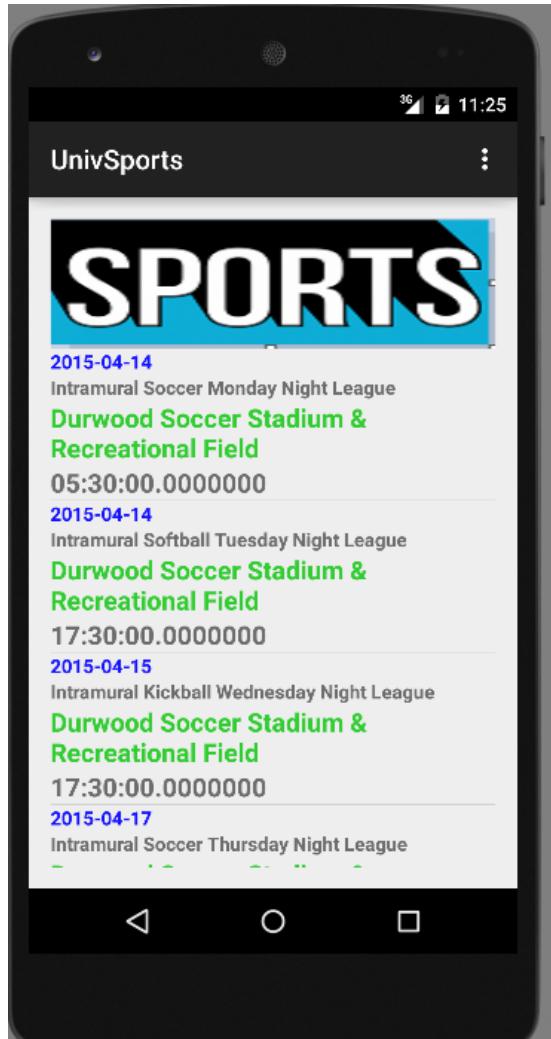
The main dashboard screen consists of separate data sections like About UMKC, Map, Transportation, Academic details and sports sections.

## 1. About umkc Page



This Page displays the end user details about the university, contact information and all useful information about departments.

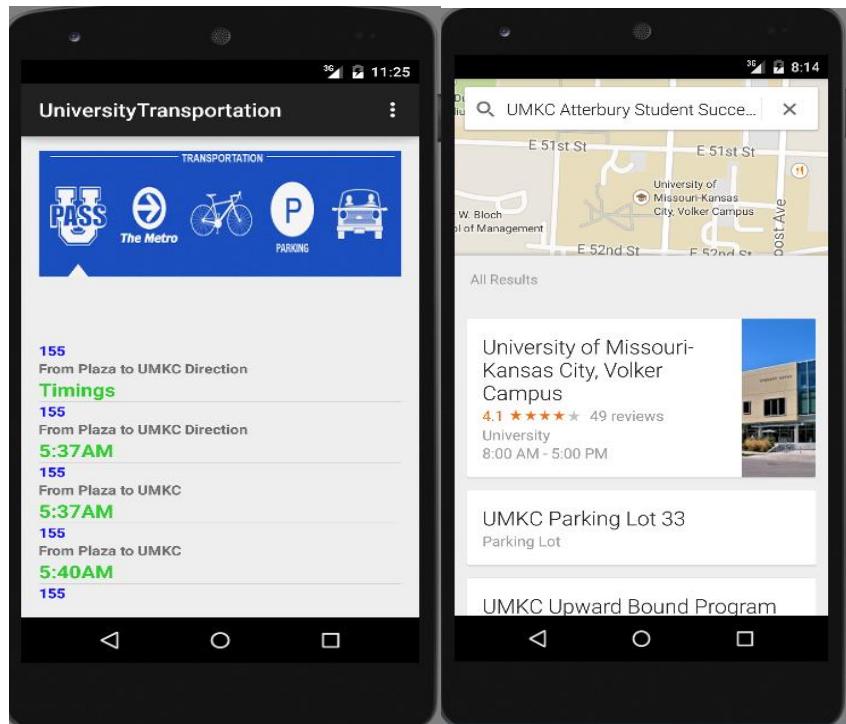
2. Sports page



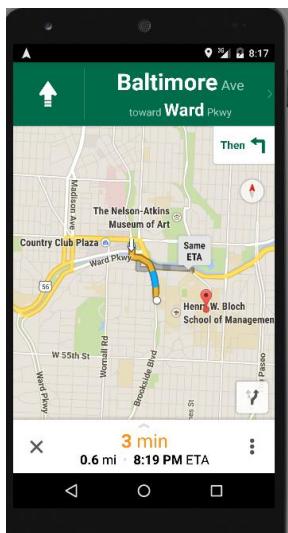
Sports Details Page

Information related to sports which are going around university are shown to User with their location and timing. User can click on the details so that it will show the navigation to that location on the map.

### 3. Transportation



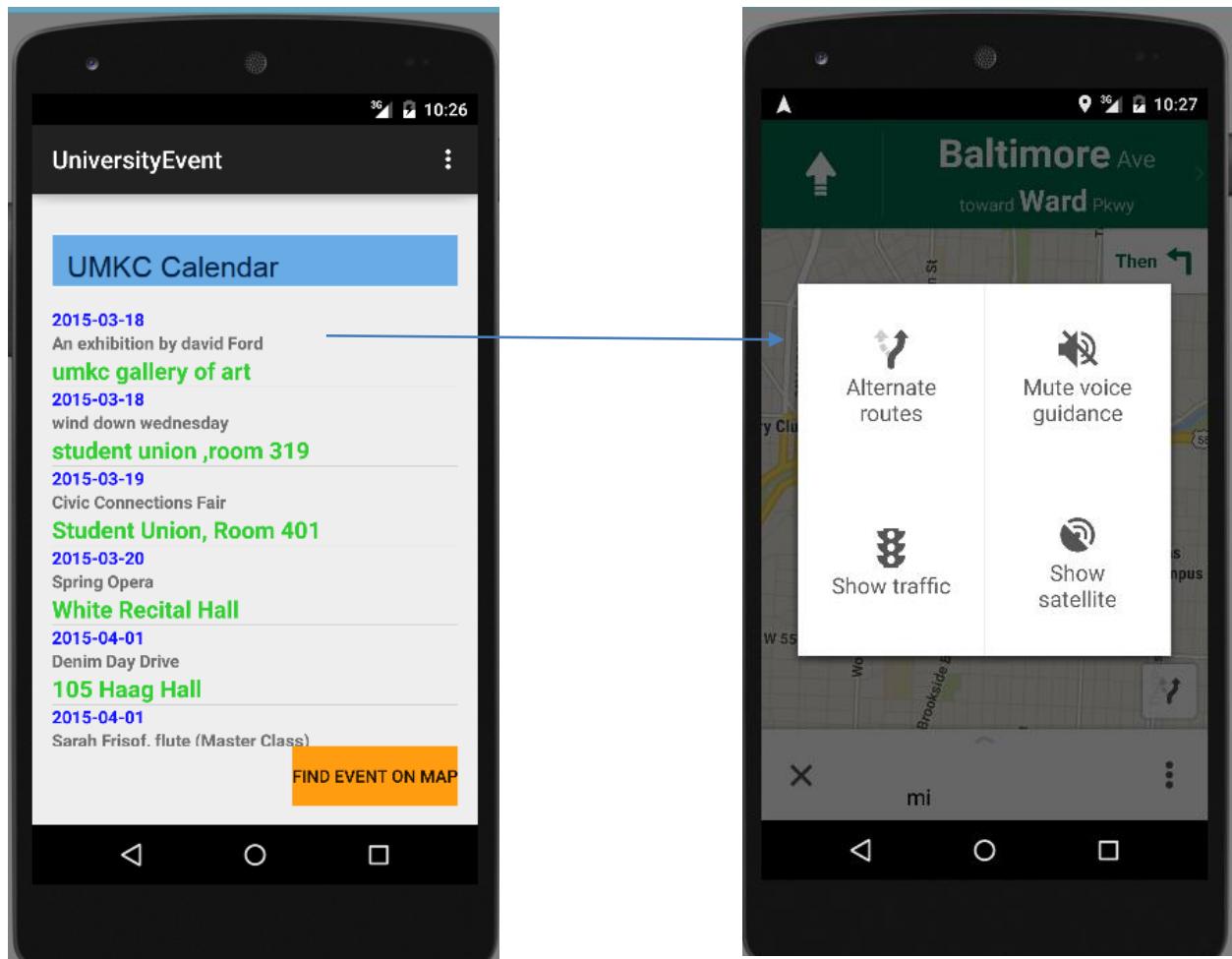
Transportation page is integrated to Google map so that user can check for navigation. Available bus services to university , timing and navigation routes are shown.



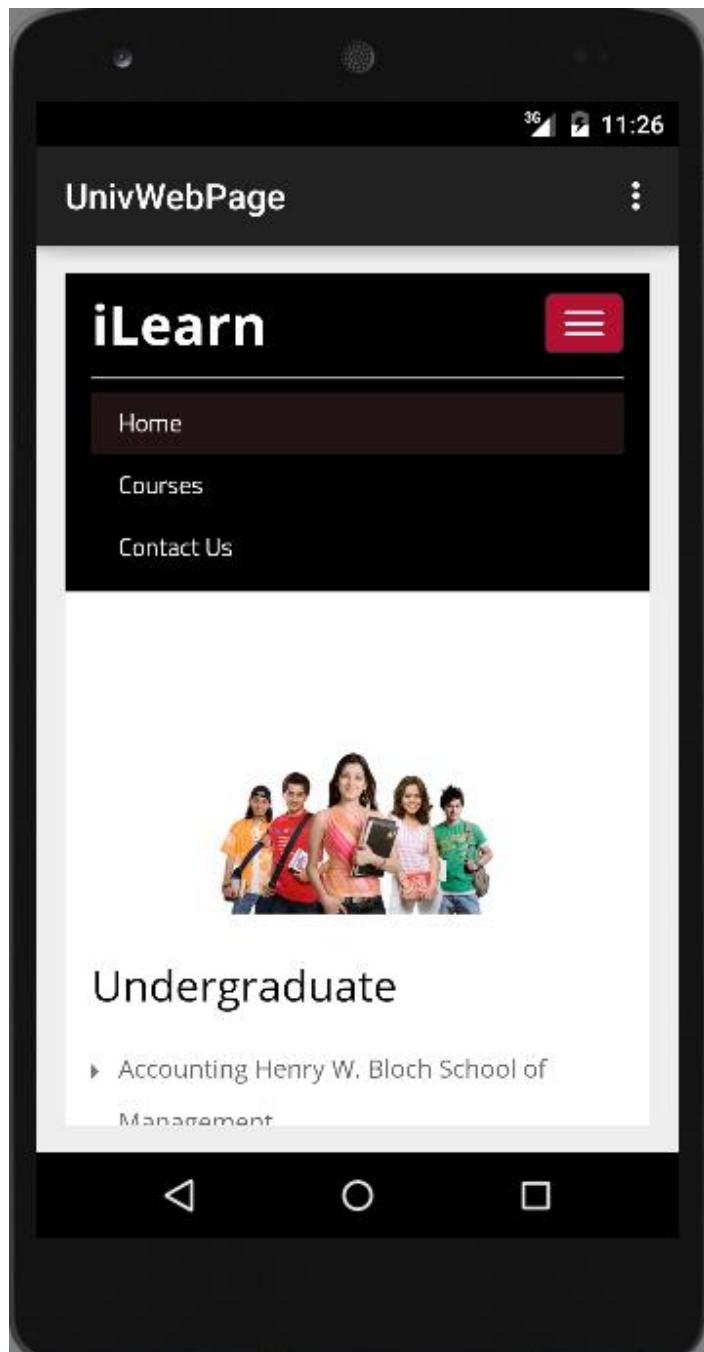
Available navigation options are displayed on application screen

#### 4. Events Page

All events happening around umkc are shown to user with a simple list display. User can click on the one of event details and system displays user the location of event and possible routing locations with all possible transport details . Alternate routes and voice guided navigation is provided with the application



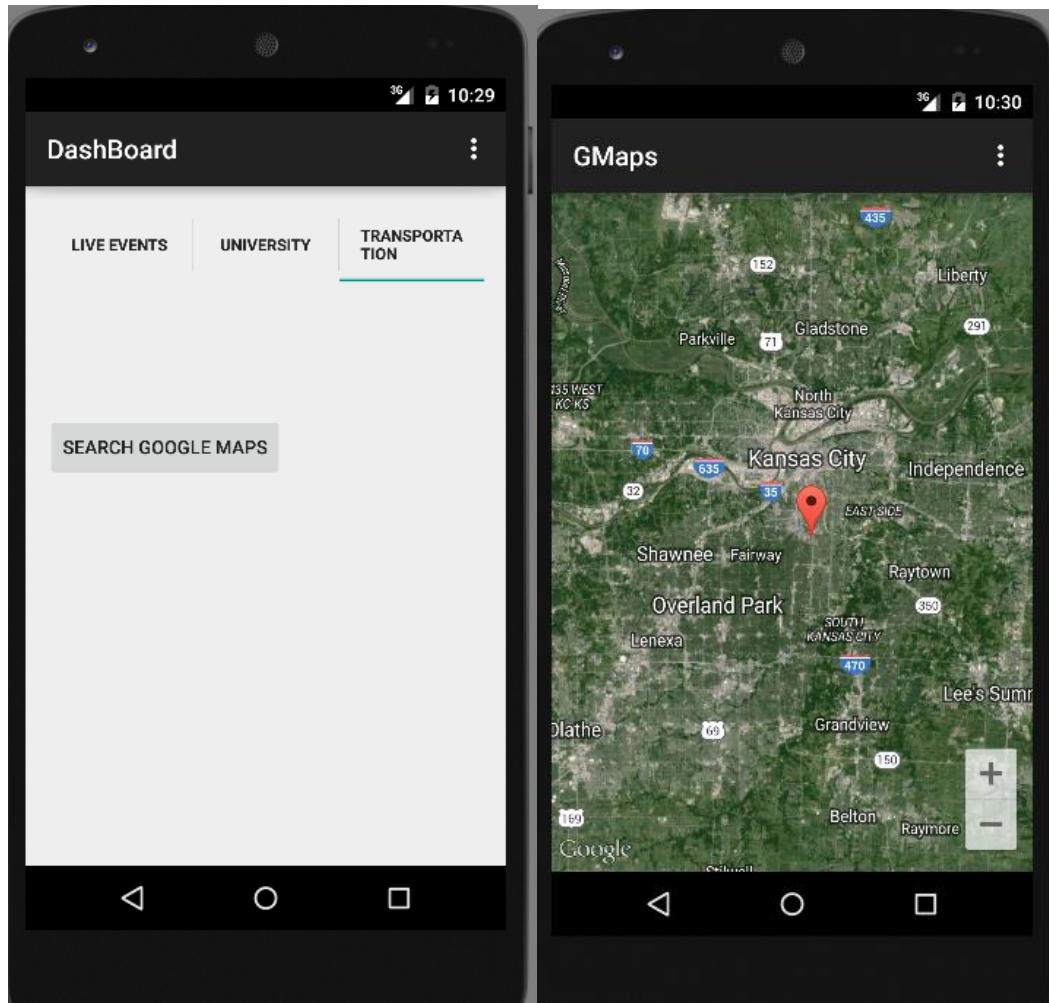
## 5. Academics Page



Information related to academics is available in this page where user can go and find out about available courses structures and time lines. Help line contacts info is also included.

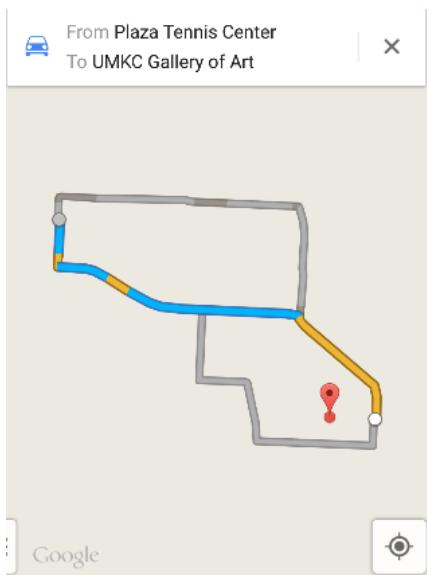
## Transportation Tab from Dashboard

We are providing an option user to select google maps so that he can select destination and can find out easy of transport and routes. Every time he doesn't need to open other application to search and instead he can click on a tab and find out these details here itself



Present

location with maps can be shown to user. He can search for other location very easily.



**Route is displayed to user on a map so that he can easily recognize the route he supposed to travel along continuous position tracking option**

## Database Design implementation

dbo.Table4: Tabl...\\APP_DATA\\DB.MDF				Table4: Query(pv..		
	date	event	location	bus number	route	time
►	3/18/2015	An exhibition b...	umkc gallery of...	155	plaza-blueridge	07:40:00
	3/18/2015	wind down we...	student union ,...	MAX	downtown-51st	08:00:00
	3/19/2015	Civic Connecti...	Student Union, ...	155	blueridge-plaza	08:20:00
	3/20/2015	Spring Opera	White Recital H...	MAX	51st-downtown	08:40:00
	4/1/2015	Denim Day Drive	105 Haag Hall	155	plaza-blueridge	08:40:00
	4/1/2015	Sarah Frisof, flu...	Grant Recital Hall	MAX	downtown-51st	09:00:00
	4/2/2015	UMKC Health S...	Pierson Auditor...	MAX	51st-downtown	09:10:00
	4/3/2015	New Historical ...	Cockefair Hall 2...	155	blueridge-plaza	09:20:00
	4/3/2015	Clean Commut...	Swinney Recrea...	MAX	downtown-51st	09:30:00
				MAX	51st-downtown	09:40:00
				155	plaza-blueridge	10:00:00

Created database to store live events happening around UMKC and web service is implemented to fetch these events

## Testing:

### Web Application Load, Stress and Performance Tool:

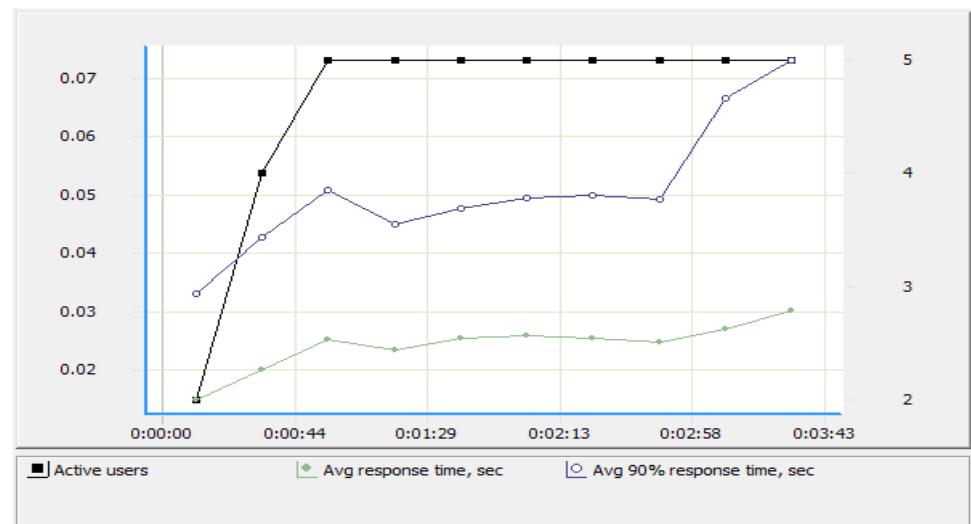
WAPT is a load and stress testing tool that provides effective way to test a web page, web portal etc. We performed WAPT on web service on transportation. Here are the screenshots of testing using WAPT on link:

[“http://kc-sce-cs551.kc.umkc.edu/aspnet\\_client/Group3/Transportation/Service1.svc/transportation/details”](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details)

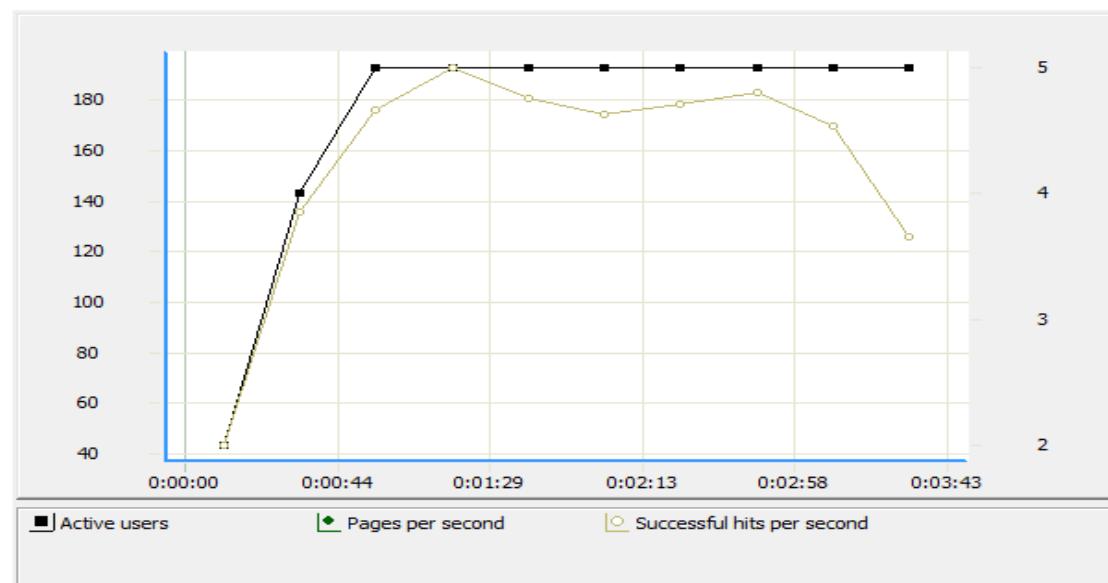
Response time, sec (with page elements)								
Name	Time	0:00:00-	0:00:24-	0:00:48-	0:01:12-	0:01:30-	0:01:54-	0:02:18-
		0:00:24	0:00:48	0:01:12	0:01:30	0:01:54	0:02:18	0:02:36
Profile1.page_1: http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details		1042	3259	4221	3470	4330	4189	3213
	Min	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
	Max	0.13(0.13)	0.25(0.25)	0.22(0.22)	0.14(0.14)	0.13(0.13)	0.14(0.14)	0.17(0.17)
	Avg	0.01(0.01)	0.02(0.02)	0.03(0.03)	0.02(0.02)	0.03(0.03)	0.03(0.03)	0.03(0.03)
	Avg90	0.03(0.03)	0.04(0.04)	0.05(0.05)	0.05(0.05)	0.05(0.05)	0.05(0.05)	0.05(0.05)

### Average response time (without page elements)

Profile1.page\_1: [http://kc-sce-cs551.kc.umkc.edu/aspnet\\_client/Group3/Transportation/Service1.svc/transportation/details](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details)  
[http://kc-sce-cs551.kc.umkc.edu:80/aspnet\\_client/Group3/Transportation/Service1.svc/transportation/details](http://kc-sce-cs551.kc.umkc.edu:80/aspnet_client/Group3/Transportation/Service1.svc/transportation/details)



### Overall performance



## YSlow Analyzer:

Yslow is to analyze the web pages and knowing the latency of the website based on Yahoo's rules. We used Yslow for testing the performance of our web service link for transportation.

[http://kc-sce-cs551.kc.umkc.edu/aspnet\\_client/Group3/Transportation/Service1.svc/transportation/details](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details).

The screenshot shows the YSlow analysis results for the URL [http://kc-sce-cs551.kc.umkc.edu/aspnet\\_client/Group3/Transportation/Service1.svc/transportation/details](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details). The overall performance score is 94, achieved by applying the YSlow(V2) ruleset. The analysis is filtered by Content (6), Cookie (2), CSS (6), Images (2), JavaScript (4), and Server (6).

**Grade A** Overall performance score 94 Ruleset applied: YSlow(V2) URL: <http://kc-sce-cs551.kc.umkc.edu/.../details>

**ALL (23)** FILTER BY: [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(4\)](#) | [SERVER \(6\)](#)

**A Make fewer HTTP requests**

**D Use a Content Delivery Network (CDN)**

**A Avoid empty src or href**

**D Add Expires headers**

**B Compress components with gzip**

**A Put CSS at top**

**A Put JavaScript at bottom**

**A Avoid CSS expressions**

Grade A on Make fewer HTTP requests

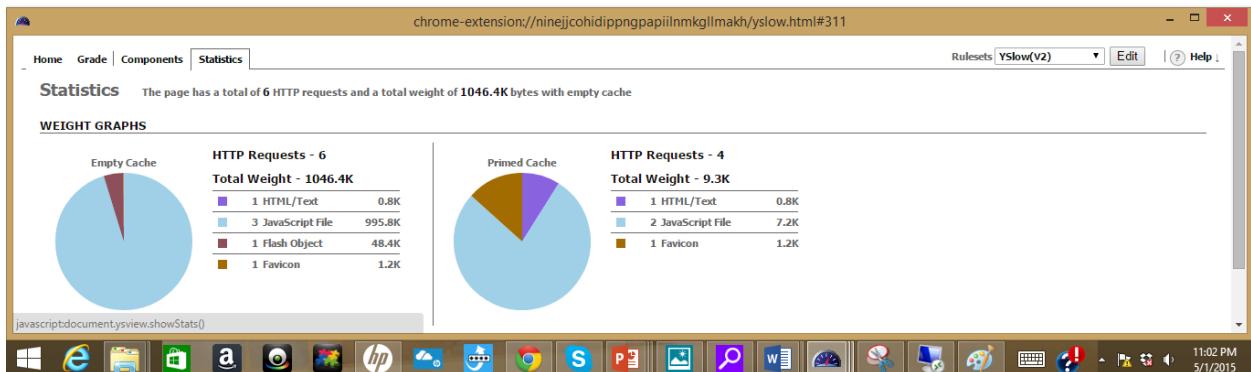
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[Read More](#)

```

https:// kc-sce-cs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details
[{"BusNo": "155", "Directions": "From Plaza to UMKC Direction", "Timings": "\'155\'", "Directions": "\'From Plaza to UMKC\'", "Timings": "\'5:37AM\'"}, {"BusNo": "\'155\'", "Directions": "From Plaza to UMKC", "Timings": "\'5:37AM\'"}, {"BusNo": "\'155\'", "Directions": "\'From Plaza to UMKC\'", "Timings": "\'5:40AM\'"}, {"BusNo": "\'155\'", "Directions": "From Plaza to UMKC", "Timings": "\'5:40AM\'"}, {"BusNo": "\'155\'", "Directions": "\'From Plaza to UMKC\'", "Timings": "\'6:40AM\'"}, {"BusNo": "\'155\'", "Directions": "From Plaza to UMKC", "Timings": "\'6:40AM\'"}, {"BusNo": "\'155\'", "Directions": "\'From Plaza to UMKC\'", "Timings": "\'7:40AM\'"}, {"BusNo": "\'155\'", "Directions": "From Plaza to UMKC", "Timings": "\'7:40AM\'"}, {"BusNo": "\'155\'", "Directions": "\'From Plaza to UMKC\'", "Timings": "\'8:40AM\'"}, {"BusNo": "\'155\'", "Directions": "From Plaza to UMKC", "Timings": "\'8:40AM\'"}]

```



**Login API(<http://localhost:56120/Service1.svc/login/bollarampavan@gmail.com>)**

## Using FireBug

Loading time: 5ms

Request size: 74kB

Response: 200 OK

URL	Status	Domain	Size	Remote IP	Timeline
GET bollarampavan@gmail.com	200 OK	localhost:56120	74 B	127.0.0.1:56120	5ms

Headers Response JSON Cache Cookies

Response Headers

Cache-Control	private
Connection	Close
Content-Length	74
Content-Type	application/json; charset=utf-8
Date	Sat, 02 May 2015 03:59:02 GMT
Server	ASP.NET Development Server/10.0.0.0
X-AspNet-Version	4.0.30319

Request Headers

Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding	gzip, deflate
Accept-Language	en-GB,en;q=0.5
Connection	keep-alive
Cookie	inject_21=1; CSUUID_EC=6fd7cba3-52ab-4ef7-06f8-809b8f3c6905; inject_17=1
DNT	1
Host	localhost:56120
User-Agent	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0
X-Forwarded-For	1.7.255.255

## Events API

### Using Firebug

Loading time: 766ms

Request size: 221.kB

Response: 200 OK

GET /js\_68164684.js?version=3...7 200 OK  
outfluenza.com 221.7 KB 173.255.196.133:80 766ms

Params Headers Response Cache

Response Headers

Access-Control-Allow-Origin: \*

Cache-Control: max-age=1800

Connection: keep-alive

Content-Encoding: gzip

Content-Type: application/x-javascript

Date: Sat, 02 May 2015 04:06:30 GMT

Expires: Sat, 02 May 2015 04:06:30 GMT

Server: nginx/1.0.15

Transfer-Encoding: chunked

Request Headers

Accept: \*/\*

Accept-Encoding: gzip, deflate

Accept-Language: en-GB,en;q=0.5

Connection: keep-alive

DNT: 1

Host: outfluenza.com

Referer: http://localhost:56120/Service1.svc/GetEventsDetails/4-5-2015

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0

X-Forwarded-For: 1.7.255.255

1 request 221.7 KB 766ms

## Sports API

### Using FireBug

Loading time: 58ms

Request size: 2.2kB

Response: 200 OK

GET 4-5-2015 200 OK localhost:56120 2.2 kB 127.0.0.1:56120 58ms

Headers Response JSON Cache Cookies

Response Headers

Cache-Control: private

Connection: Close

Content-Length: 2282

Content-Type: application/json; charset=utf-8

Date: Sat, 02 May 2015 04:11:30 GMT

Server: ASP.NET Development Server/10.0.0.0

X-AspNet-Version: 4.0.30319

Request Headers

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Encoding: gzip, deflate

Accept-Language: en-GB,en;q=0.5

Connection: keep-alive

Cookie: CSUID\_E0=6fd7cba3-52ab-4ef7-06f8-809b8f3c6905; inject\_17=1

DNT: 1

Host: localhost:56120

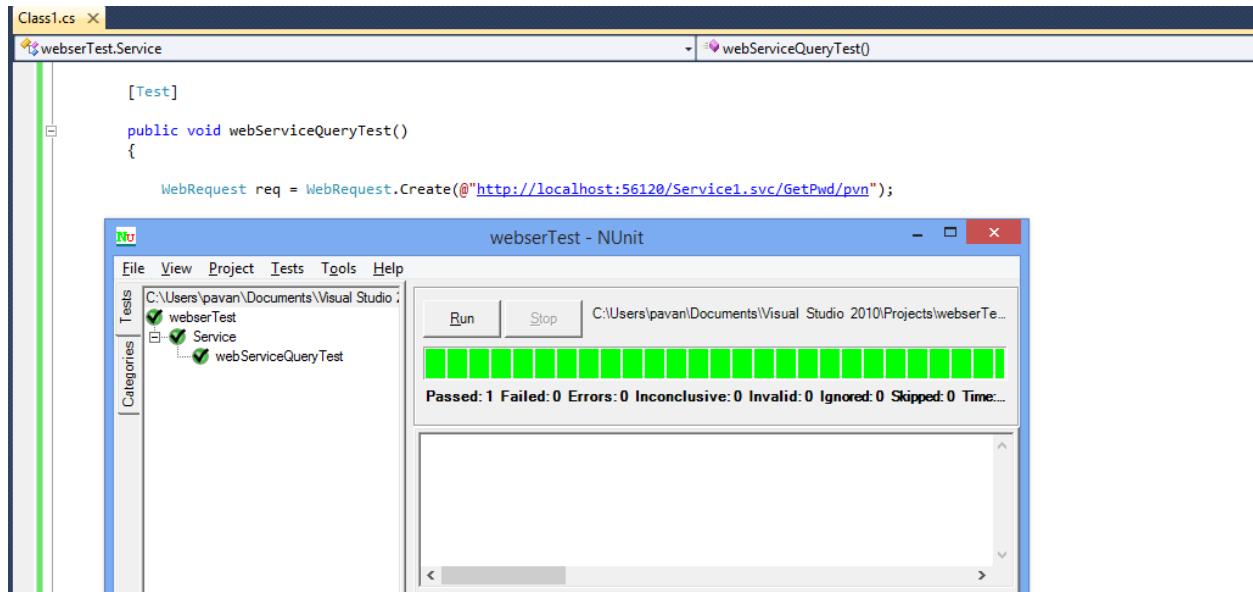
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0

X-Forwarded-For: 1.7.255.255

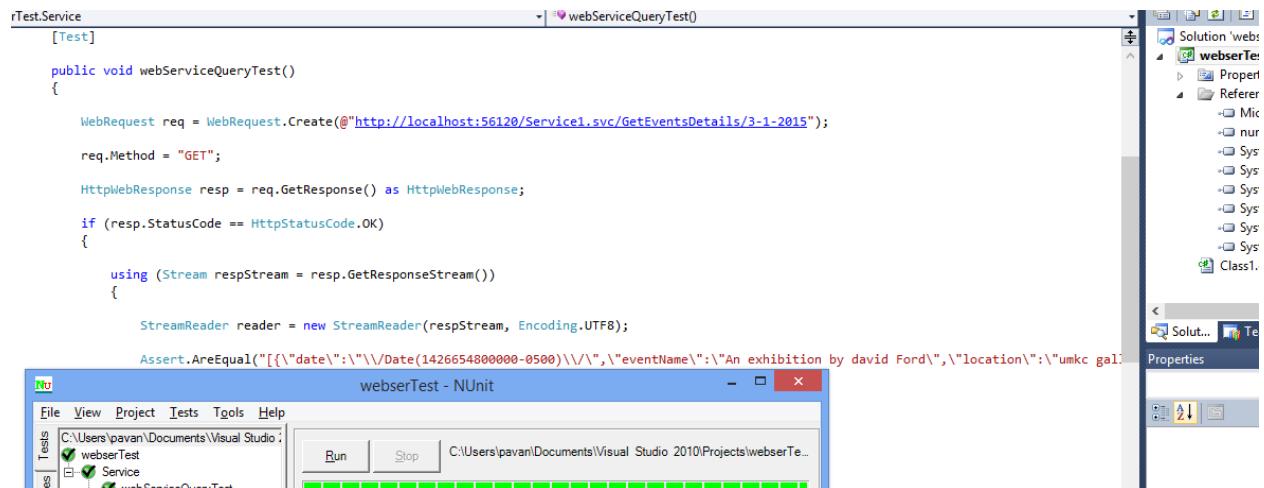
## NUnit TESTING:

NUnit testing tool which works on .NET platform for testing web service API's.

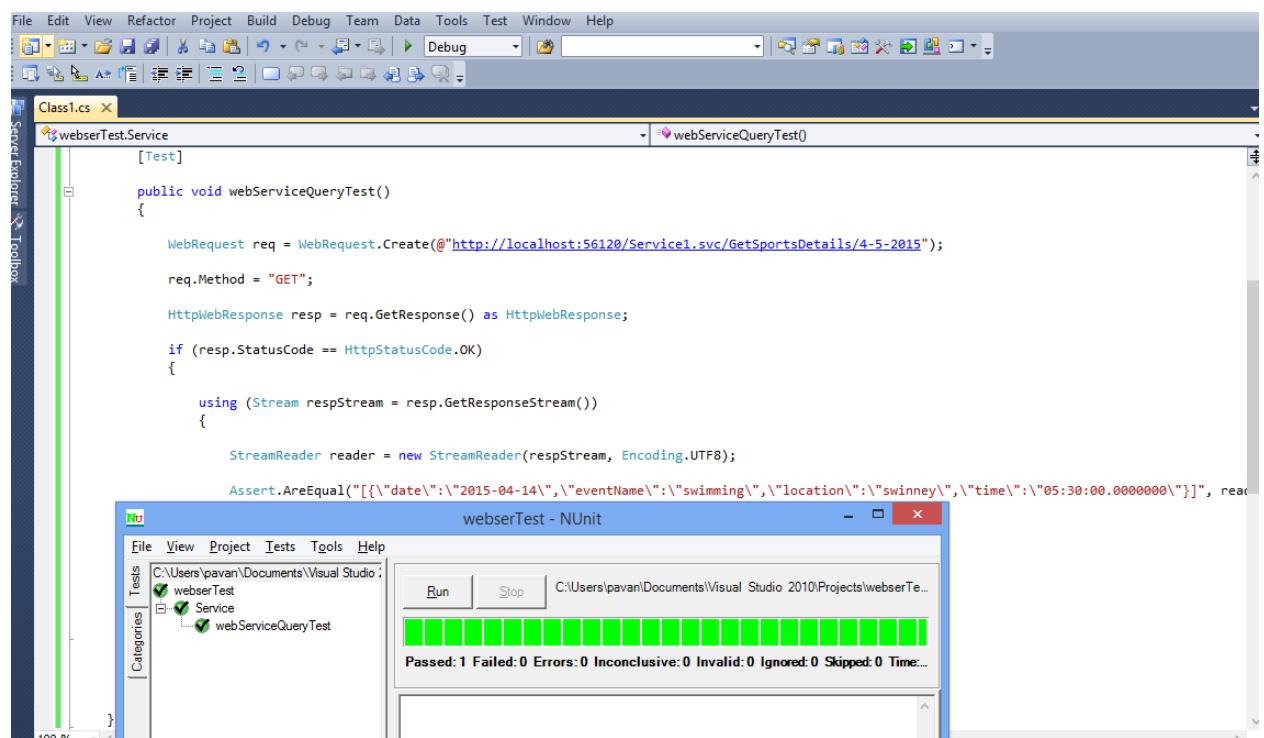
### 2. http://localhost:56120/Service1.svc/GetPwd/



## 2. http://localhost:56120/Service1.svc/GetEventsDetails/3-1-2015



## 3. http://localhost:56120/Service1.svc/GetSportsDetails/4-5-2015



## **Deployment:**

Scrum Do Link:

<https://www.scrumdo.com/projects/project/ase-project8/summary>

GIT Link:

Source Code: <https://github.com/pavankumar-b/ASeSpringSem/Increment4>

Documentation: <https://github.com/pavankumar-b/ASeSpringSem/IncrementDoc4>

YouTube link:

<https://www.youtube.com/watch?v=DtsX-dtiAIE>

## **Work Completed:**

Description:

- Creation of list view interface for transportation, events and sports.
- Integration of google maps API to display locations of university.
- Implementation of a common interface to search a location on maps.
- Testing application with fire bug and Yslow analyzer.
- Improving the User Interface.

## **Responsibility**

- Task 1: Creation of list view interface for transportation, events and sports.

Pavankumar Bollaram/ Varaprasad jaggu

- Task 2: Integration of google maps API to display locations of university.

Lakshmi Priyanka / Preetham Kumar

- Task 3: Implementation of a common interface to search a location on maps.

Varaprasad jaggu / Preetham kumar /PavanKumar

- Task 4: Testing application with fire bug and Yslow analyzer.

Pavan kumar Bollaram,varaprasad, preetham kumar, Priyanka

- Task 5: Improving the User Interface.

Pavan Kumar, Preetham kumar, Varaprasad jaggu, Priyanka

## **Project Management:**

Time Taken:

300 Man Hours

Contribution:

Pavan Kumar, Bollaram (25%)

Preetham Kumar, Danaboina (25%)

Lakshmi Priyanka (25%)

Vara Prasad Reddy, Jaggu (25%)

## **Risk Management**

1. Working with Web API especially with visual studio was bit complex.
2. Very less performance testing tools for android applications.
3. Emulator runs poorly
4. Data Tables needs to refreshed regularly to make up to date information in application

## **Bibliography**

I. Android Programming

<http://www.androidhive.info/>

<http://stackoverflow.com/>

II. API's

External Events API-<http://kansascity.eventful.com/events>

Google maps API-<https://developers.google.com/maps/>

III. Testing

<http://www.nunit.org/>

[www.yslow.org](http://www.yslow.org)

IV. Themes

<http://w3layouts.com/>

<http://getbootstrap.com>

V. University Related Info : <http://www.universityofmissourikansascity.com/>

## **FINAL PROJECT EVALUATION**

As part of our project completion we would like to evaluate our system. Our main aim in taking up this project was to provide a single platform to the students to access information related university they are attending and the location of the events going around.

Before starting of this project we started with UML diagram which gave us clear picture of system architecture and the requirements needed. We followed agile approach which helped us lot in creating rapid and incremental development of our modules and each stage we improved modules based on our requirement. As part of this approach we followed **Scrum Do**.

Team level contribution and individual task are assigned using this scrums and scrum stories are evaluated at each iteration and meeting held between team members. This helped us to track the deliverables. In case of any missing deliverables we increased the speed of the work to finish up the tasks assigned. Developing user interface is very crucial in this application because of heavy data involved and finally we came up with nice UI. Custom API are created according to our requirement and integrated successfully into application. Team is contributed equally and provided timely help to make deliverables on time.