**INCREMENT REPORT-4**

PROJECT NAME: GO EASY (**PG#3**)

PAVAN KUMAR BOLLARAM (CLASS ID: 07)

PREETHAM KUMAR DANABOINA (CLASS ID: 11)

LAKSHMI PRIYANKA (CLASS ID: 49)

VARAPRASAD REDDY JAGGU (CLASS ID: 22)

CS5551 Increment #4 PG3

**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?[...](http://api.eventful.com/docs/auth)&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://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:**

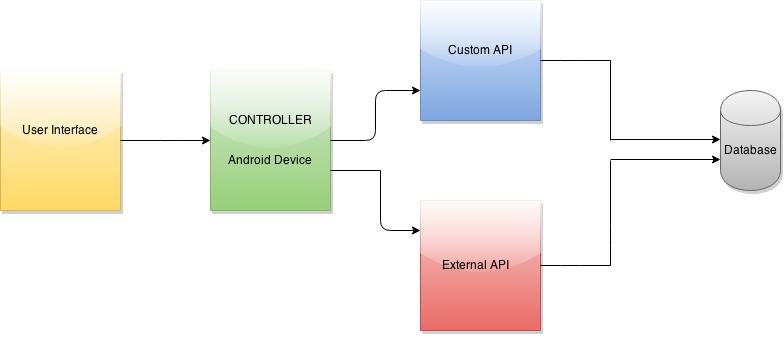
User interface provides feasible access to all events included in the android screen. Firstly, student signs in or sign up at login page instead with Gmail account student can login. Then, it re- directs after verification from database of form inputs given by users to login successfully. Secondly, student 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.

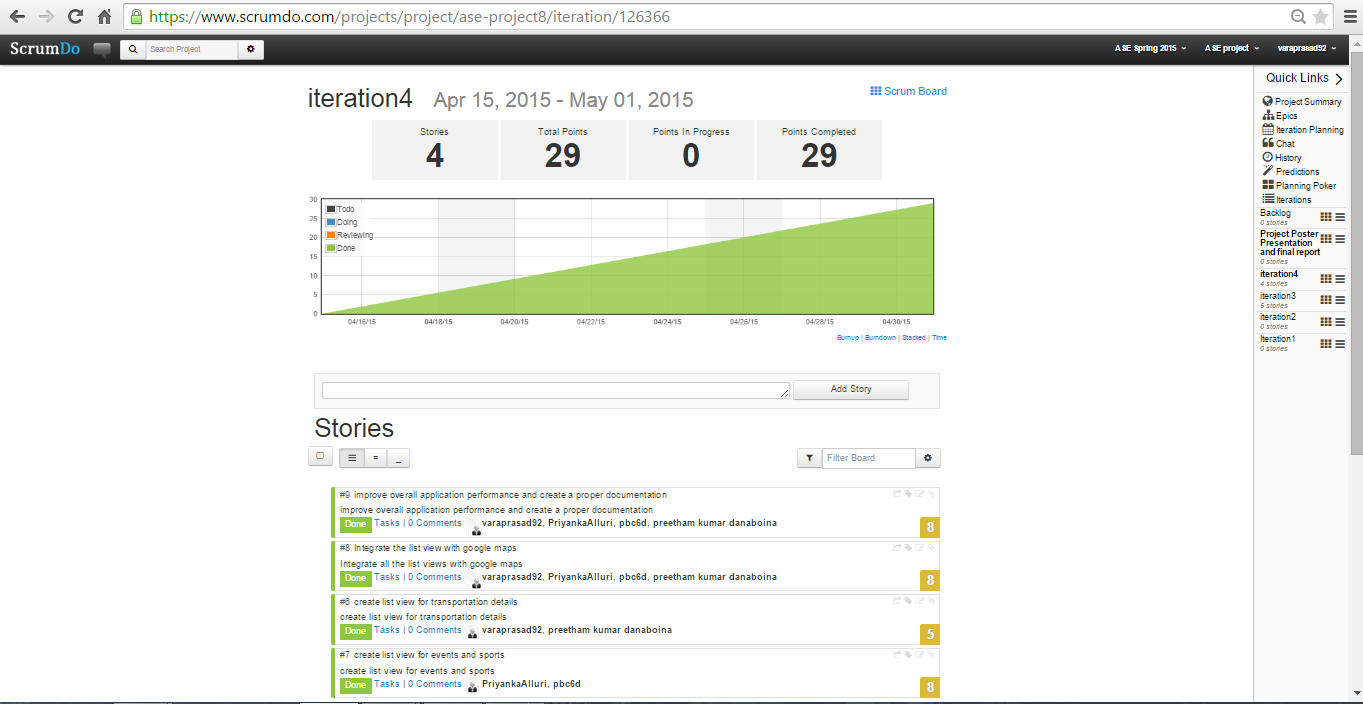


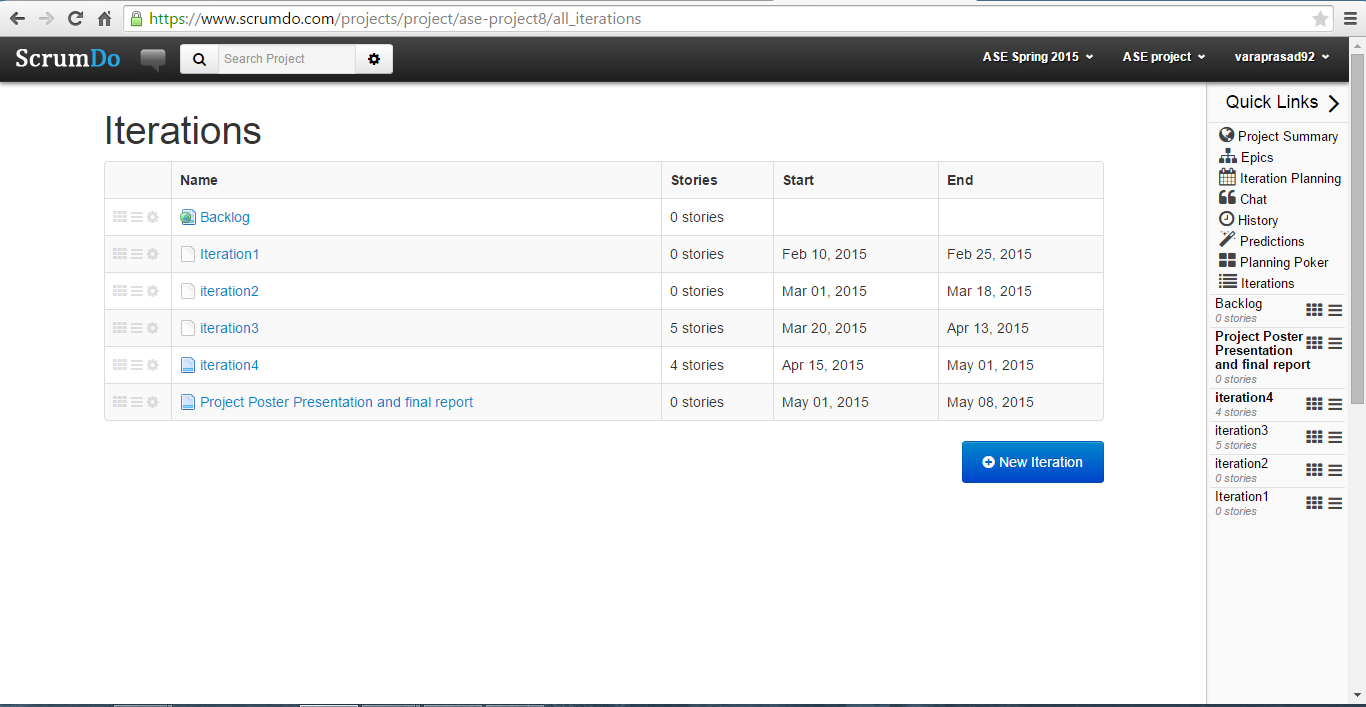
**Scrum Do and Stories:**

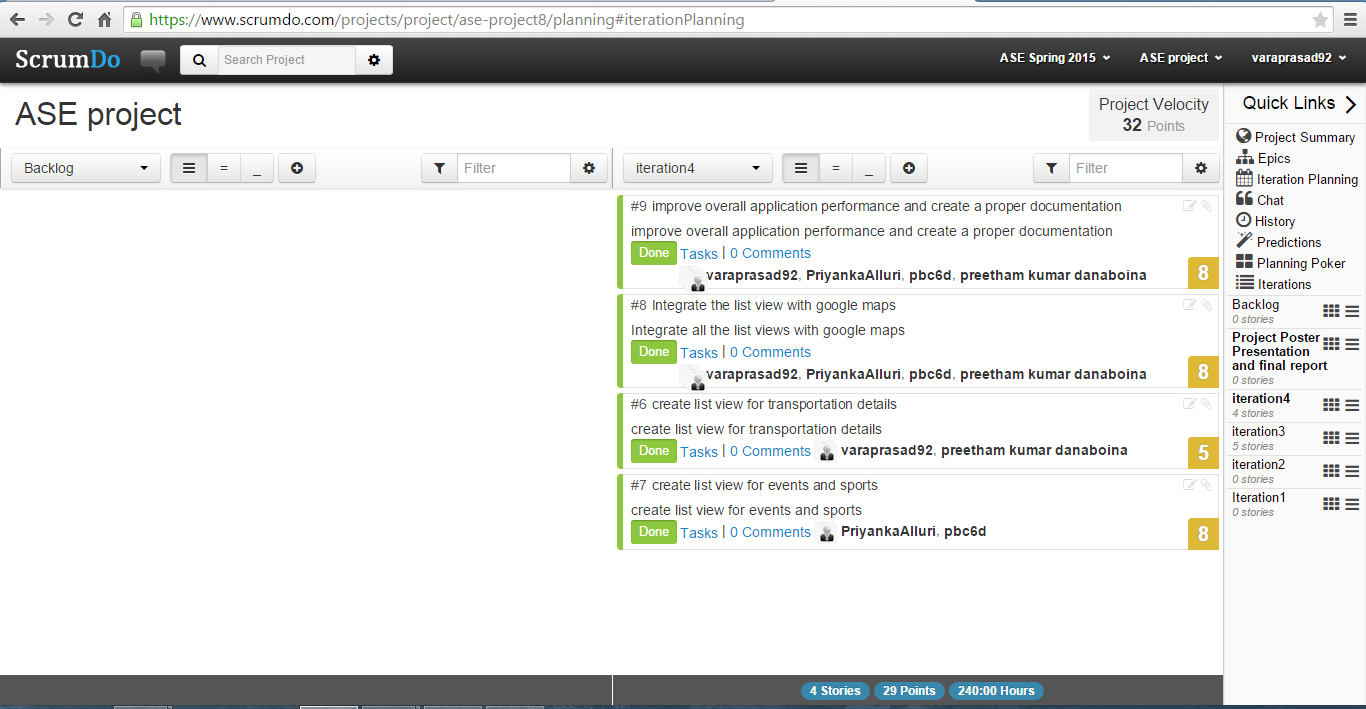
Effective product development is possible only when there is incremental and iterative methodology. This is can be done by using Scrum do which uses agile methodology by setting priority in backlog and adding stories. Iteration consists of stories to be done. If all the stories in the iteration is done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

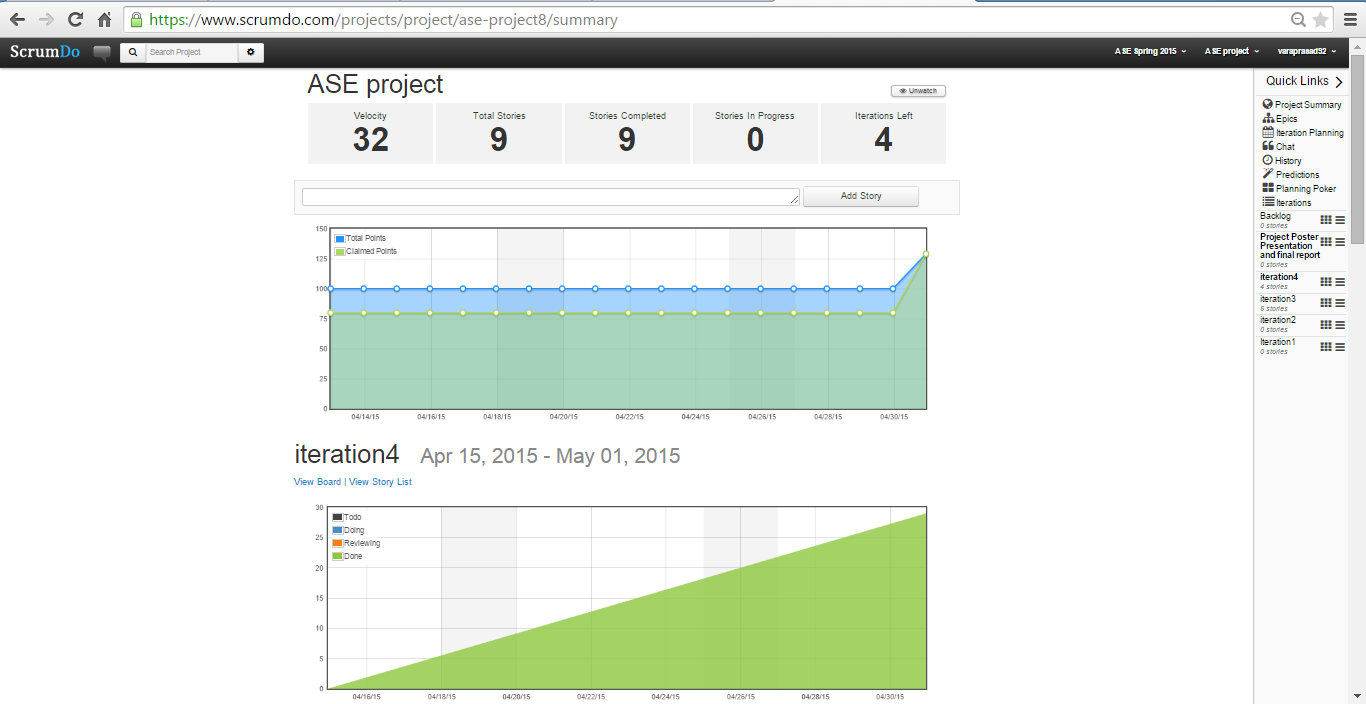
* + - * Creation of list view interface for transportation, events and sports.
      * Integration of google maps API to display locations of university.
      * Implementation of a common interface to search a location on maps.
      * Testing application with fire bug and Yslow analyzer.
      * Improving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

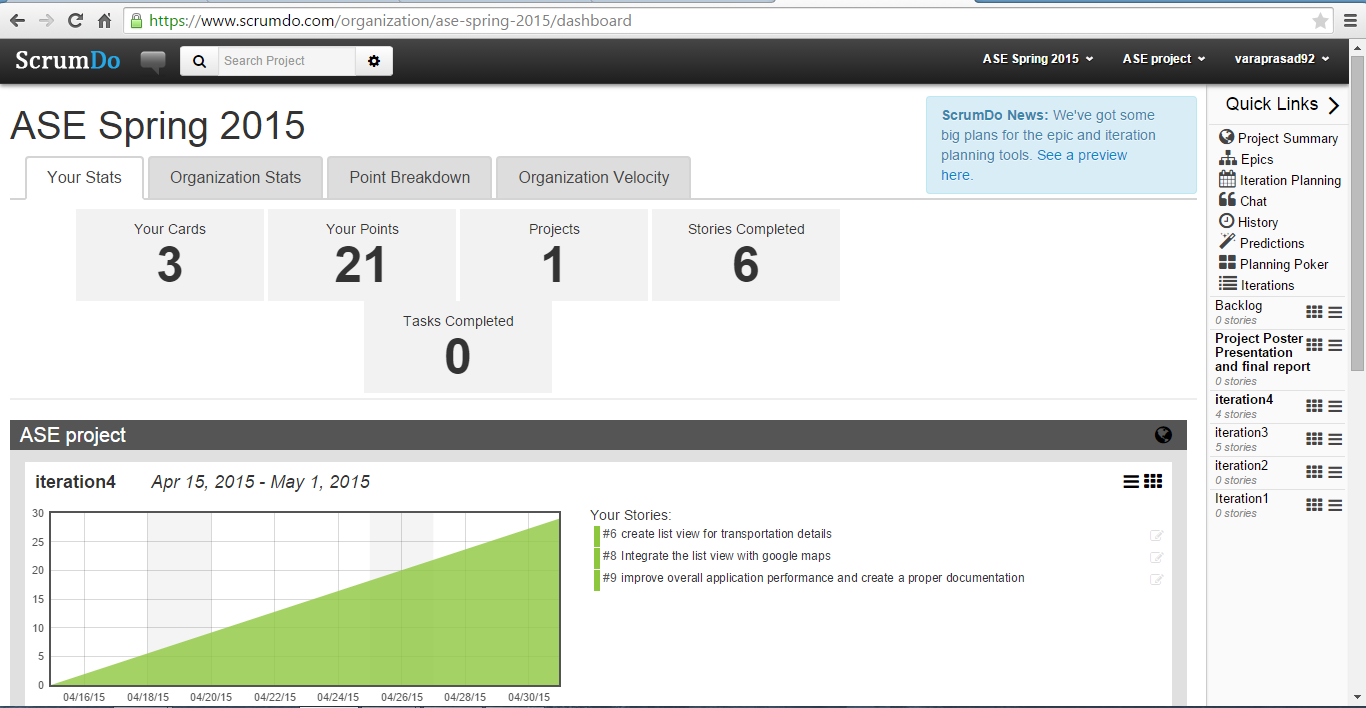
The below figures are from the scrum do which describes the dynamic flow of iteration 4.

****

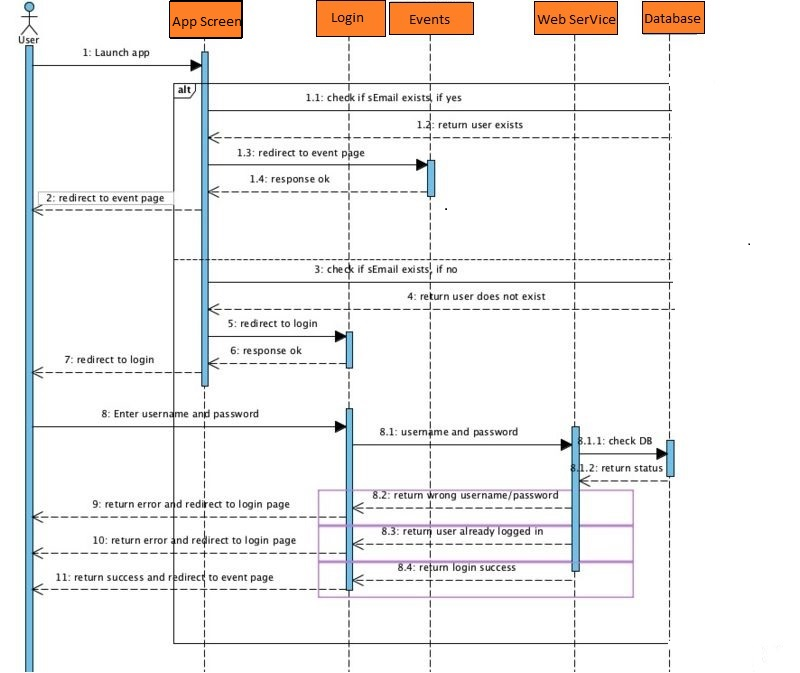
****

****

****

****

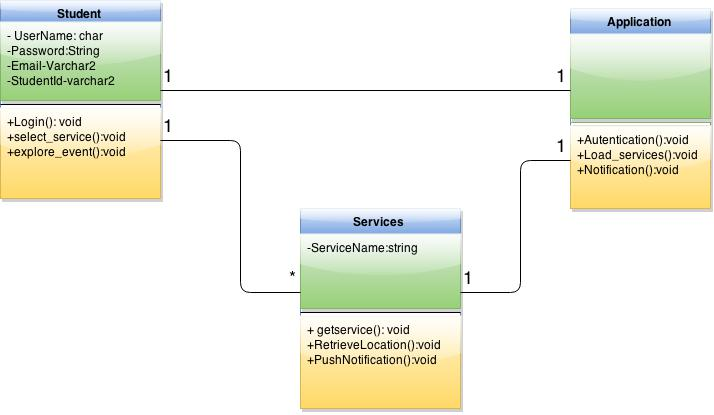
**Sequence Diagrams**:



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.

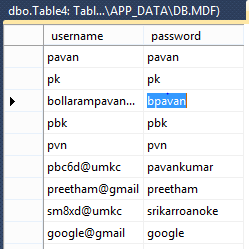


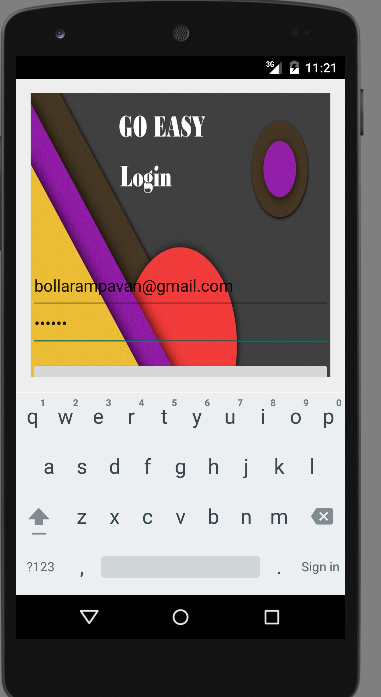
table created to store Login details and Login API connects to this table and fetches data.

**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

****

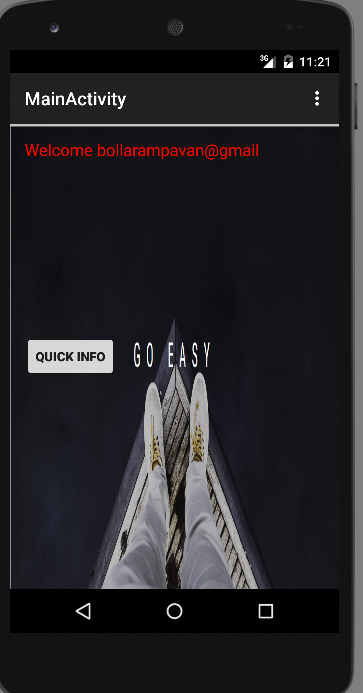
Login Screen

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.

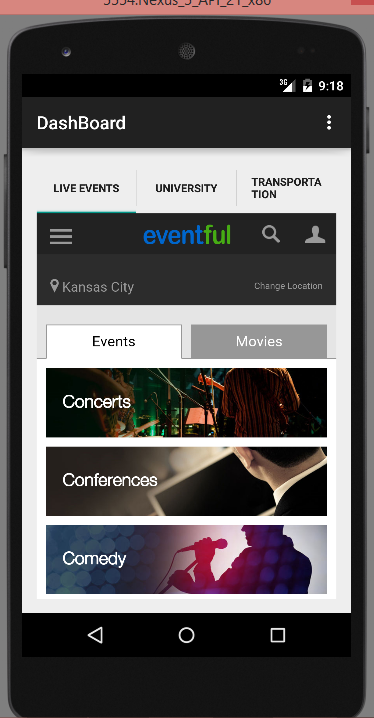
Email and Password: These fields which takes input from user and validates the user details which are already into the database when registered. Once validated user can login to application it pops into his dashboard.

Welcome Page (Screen II)

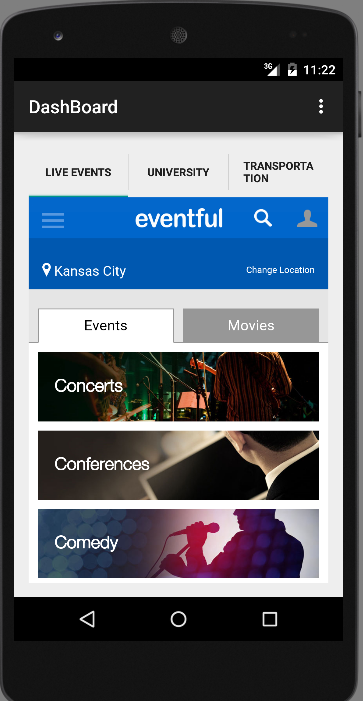
****

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.

****

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.

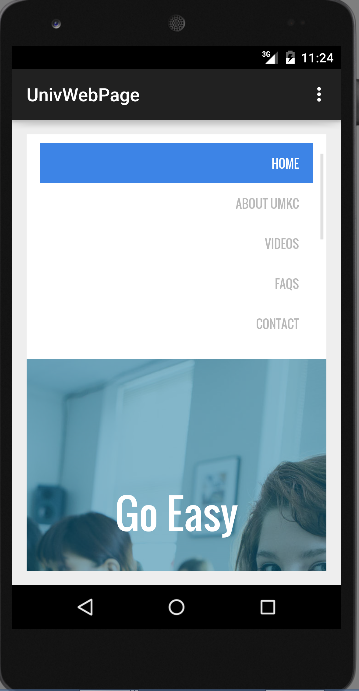
This screen consists of all data regarding university and it can be further improvised using deeper user interface

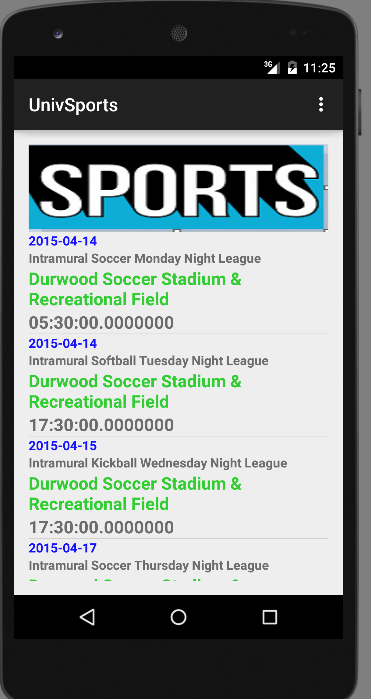
As of now we included few of the basic required data sections across screen they are as shown in the following images

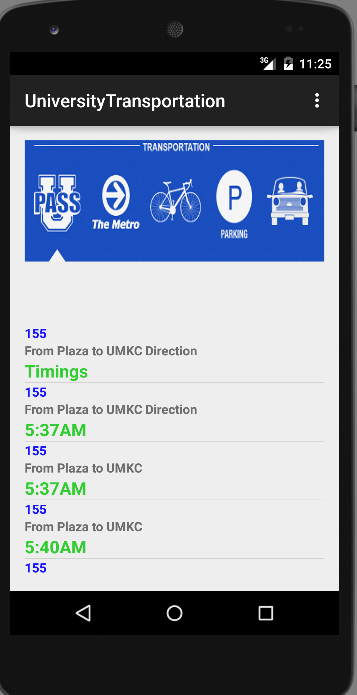
The main dashboard screen consists of data sections like About UMKC, Map, Transportation, Academic details and sports sections.

Images of About umkc

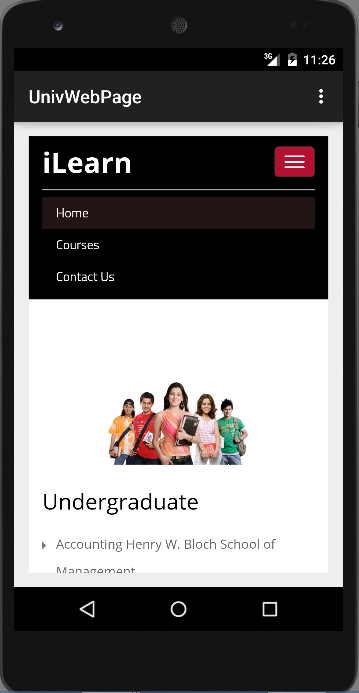
****

****

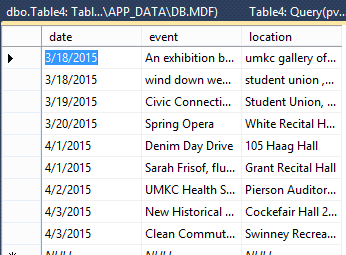
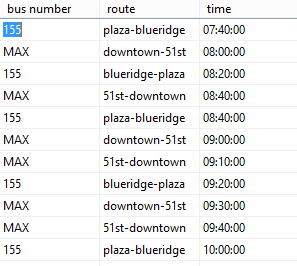
Transportation:

****

Academics:

****

**Database Design implementation**

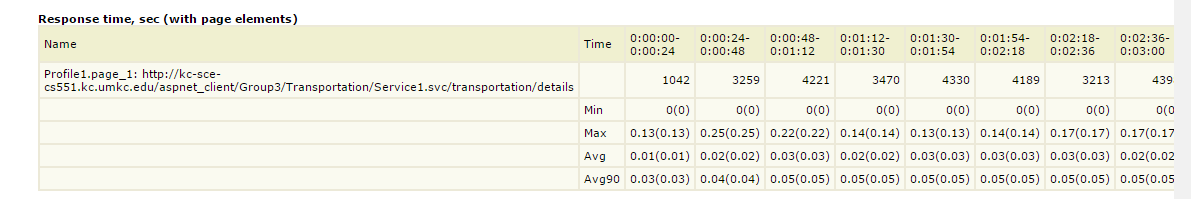
Created database to store live events happening around UMKC and web service is implemented to fetch these events

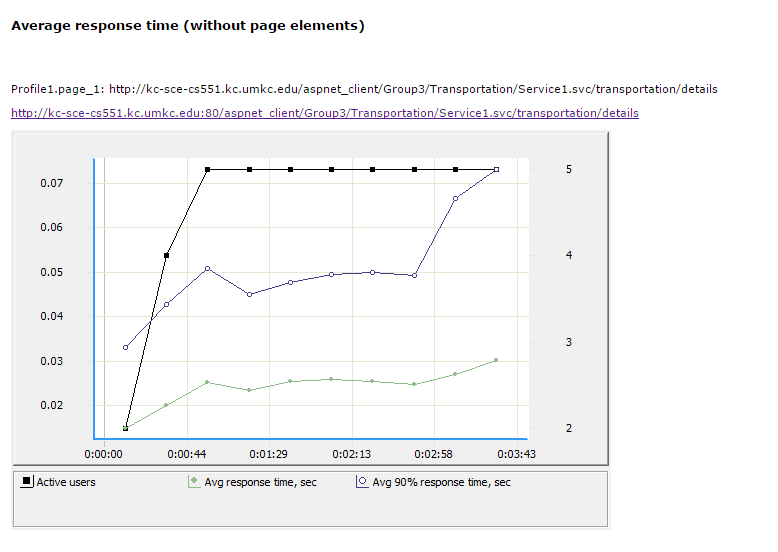
**Testing:**

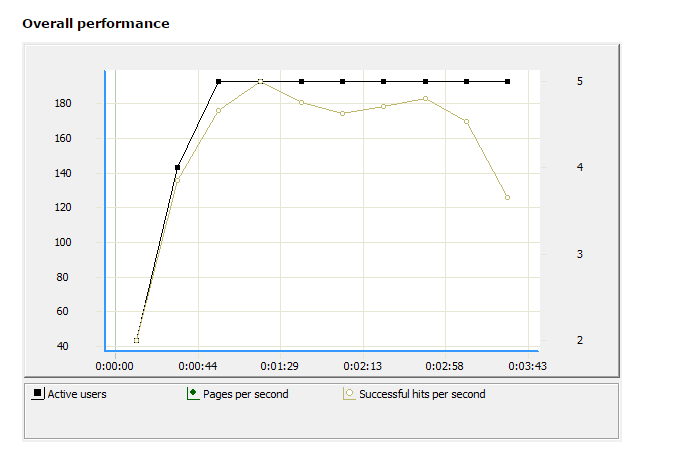
**Web Application Load, Stress and Performance Tool:**

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>”



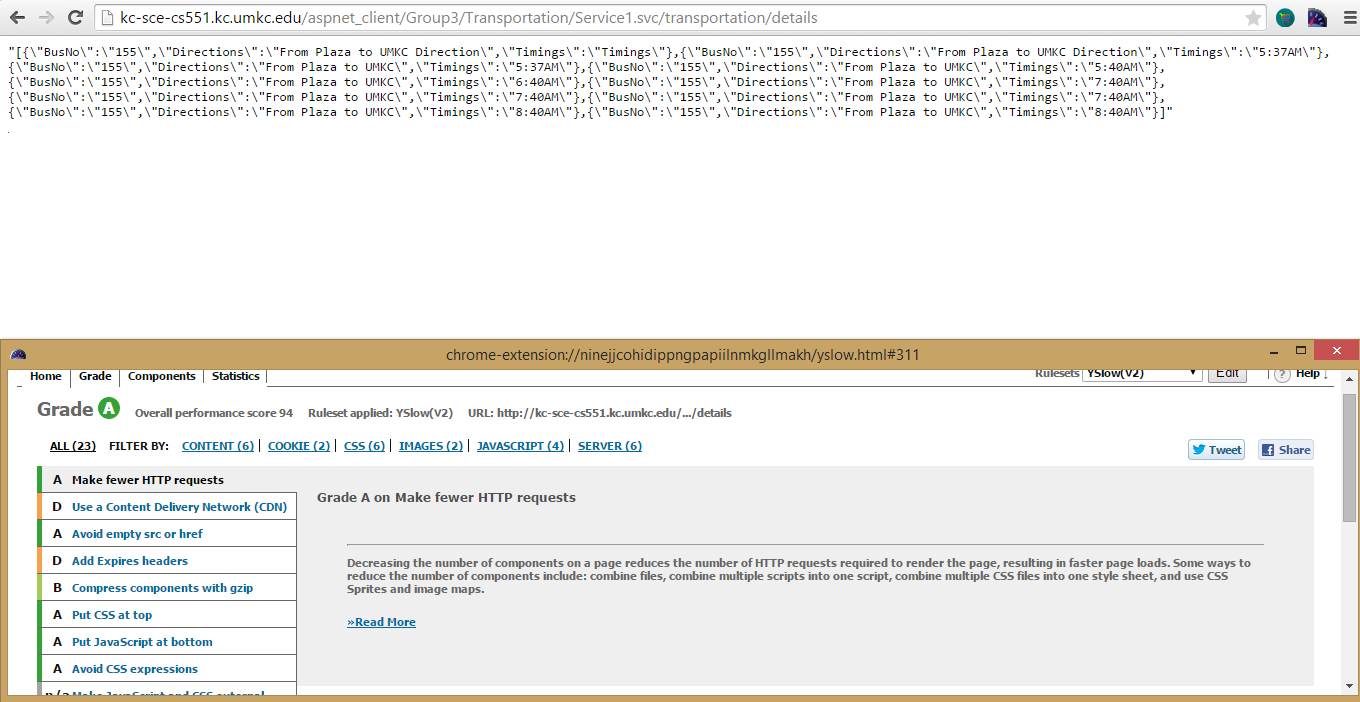


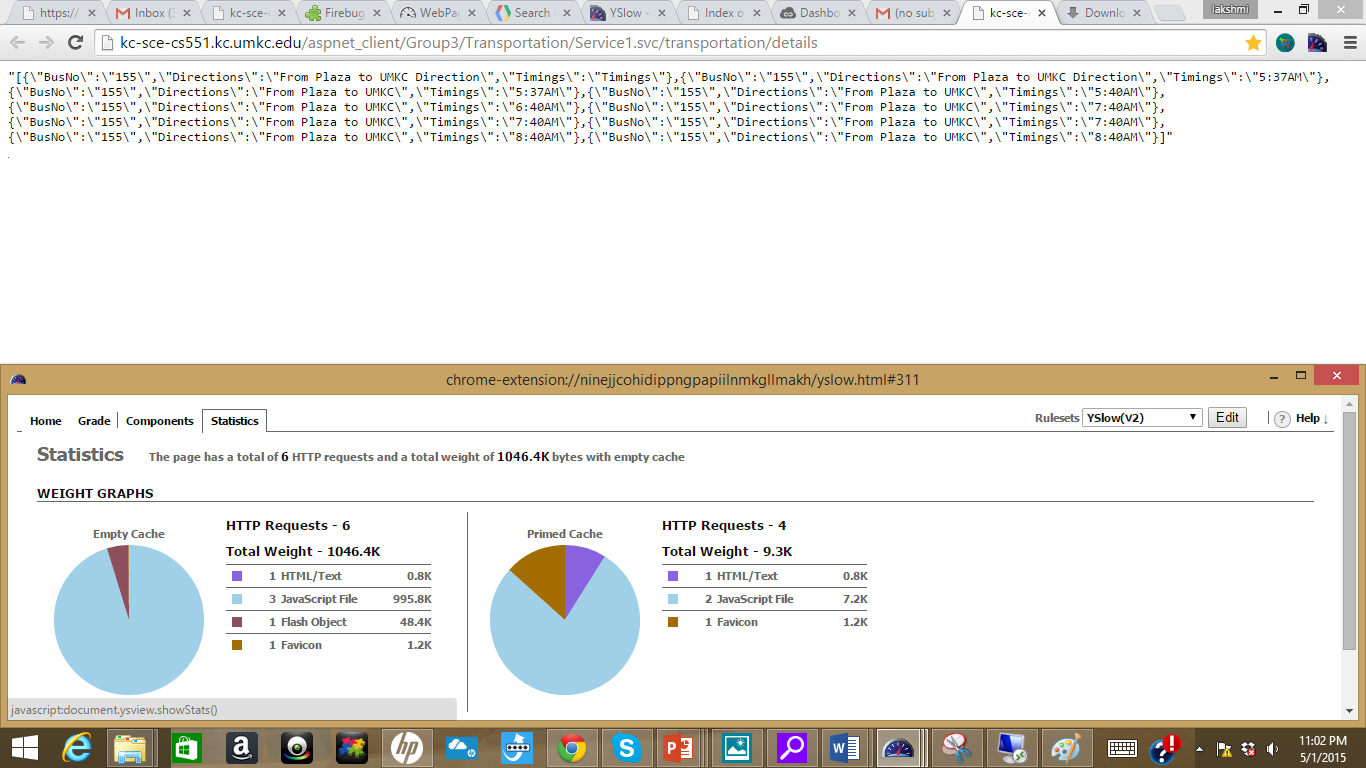


**YSlow Analyzer:**

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-scecs551.kc.umkc.edu/aspnet_client/Group3/Transportation/Service1.svc/transportation/details>.



****

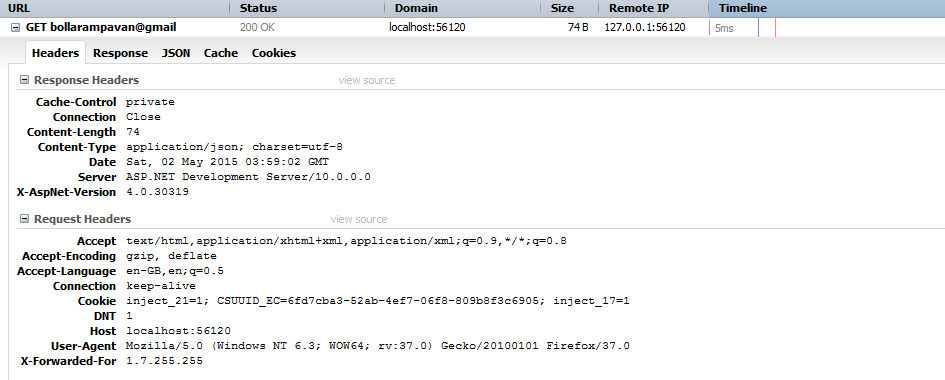
**Login API(<http://localhost:56120/Service1.svc/login/bollarampavan@gmail>)**

**Using FireBug**

Loading time: 5ms

Request size: 74kB

Response: 200 OK

****

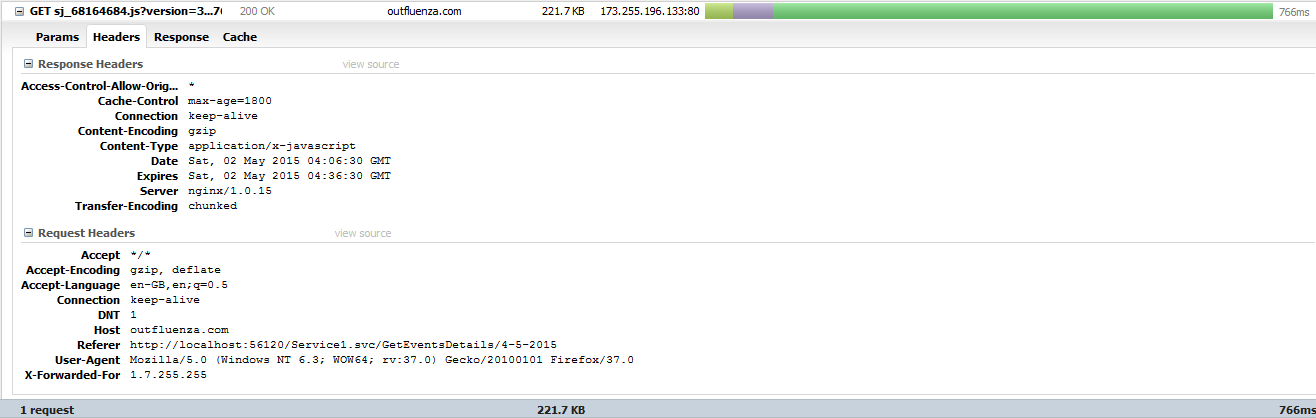
**Events API**

**Using Firebug**

Loading time: 766ms

Request size: 221.kB

Response: 200 OK

****

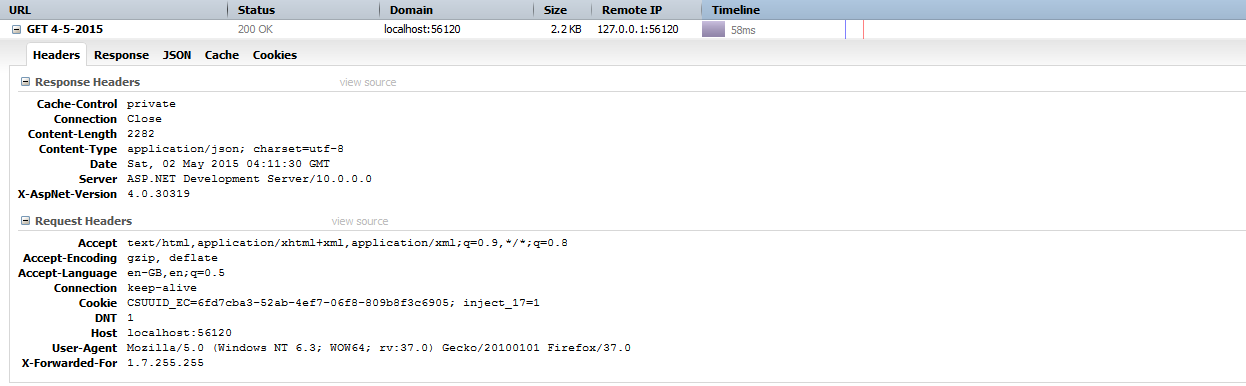
**Sports API**

**Using FireBug**

Loading time: 58ms

Request size: 2.2kB

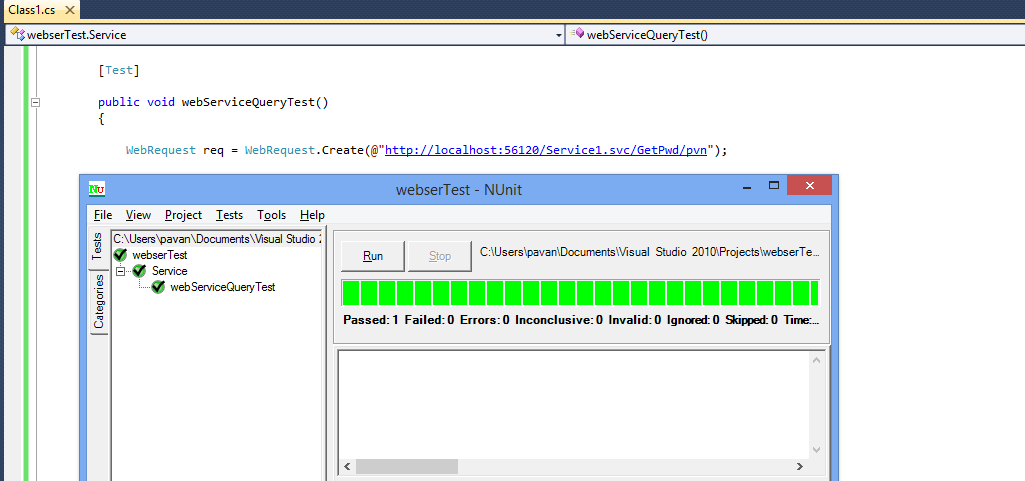
Response: 200 OK

****

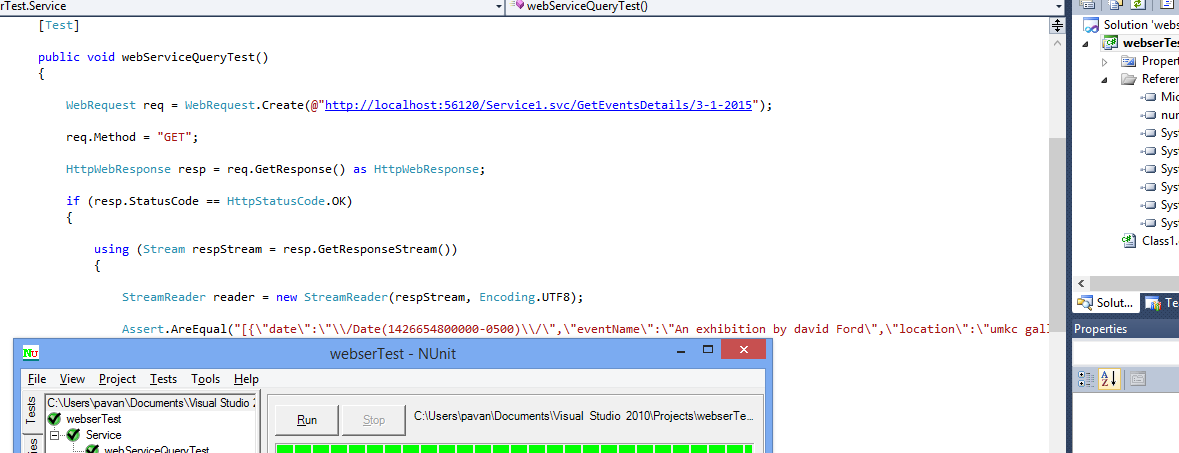
**NUnit TESTING:**

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

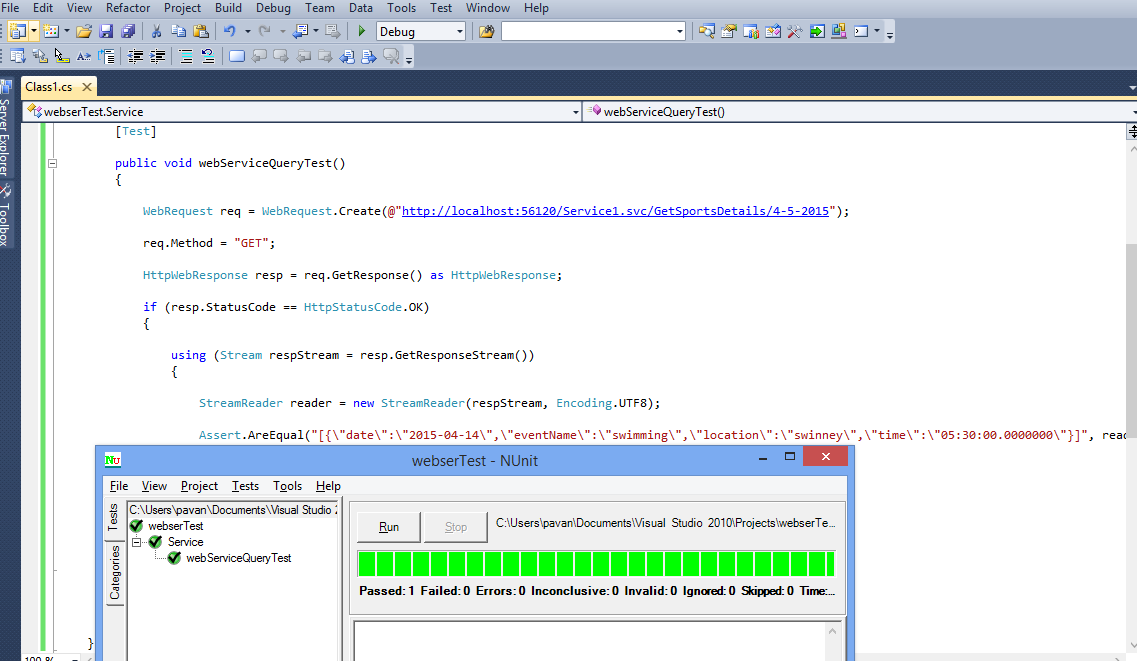
1. [**http://localhost:56120/Service1.svc/GetPwd**](http://localhost:56120/Service1.svc/GetPwd)/



**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 Link:

Source Code: https://github.com/pavankumar-b/ASEspringSem/Increment4

Documentation: <https://github.com/pavankumar-b/ASEspringSem/IncrementDoc>4

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. 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

**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](http://getbootstrap.com/)

V. University Related Info

<http://www.universityofmissourikansascity.com/>