GIVE AWAY

CS551-Advanced Software Engineering-Spring-2015

Instructor: *Lance Faegan*
TA: *Bharath Viswanadam*

Team Members:

Sashidhar Reddy Gowra	*12428313*
Venkataramana Yashwant Kumar Palisetty	*16202251*
Ravi Kanth Devanaboyina	*16198171*
Anudeep Reddy Gujjula	*16190413*

Table of Contents:

Project Deployment

The way we have organized our project, the working flow, the methodologies is captured in the form of a video and is uploaded into the YouTube for people reference. It can be accessed through the below URL:

https://youtu.be/EBJBI0gHe6A

During the entire phase of the project we have followed agile methodology and used the agile tool Scrum do for managing the work contributed by the team members. The following is the Scrum do link for our project:

https://www.scrumdo.com/projects/project/giveaway/summary

We have managed our entire source code using the GitHub and the entire source code was published to the following path:

https://github.com/sashi987/ASE/tree/master/Final%20Project

This Final Report can be accessed from the below path of the GitHub:

https://github.com/sashi987/ASE/tree/master/Final%20Project

User Manual

System Requirements:

- Operating System: Android
- Android Version Support: Froyo to KitKat
- Memory Required: 256 MB
- Internet Connection: Yes

Devices: Android Devices

- Bluetooth 2.0 Compatible Android devices
- Android v2.3 and above

Goal:

The prime objective of this project “Give Away” is restricted only to students, where the students can login to our online portal Give Away using their University mail-id to access the catalogue displayed by the non-profit organizations or any particular individual. The students can select the items and request them for delivery.

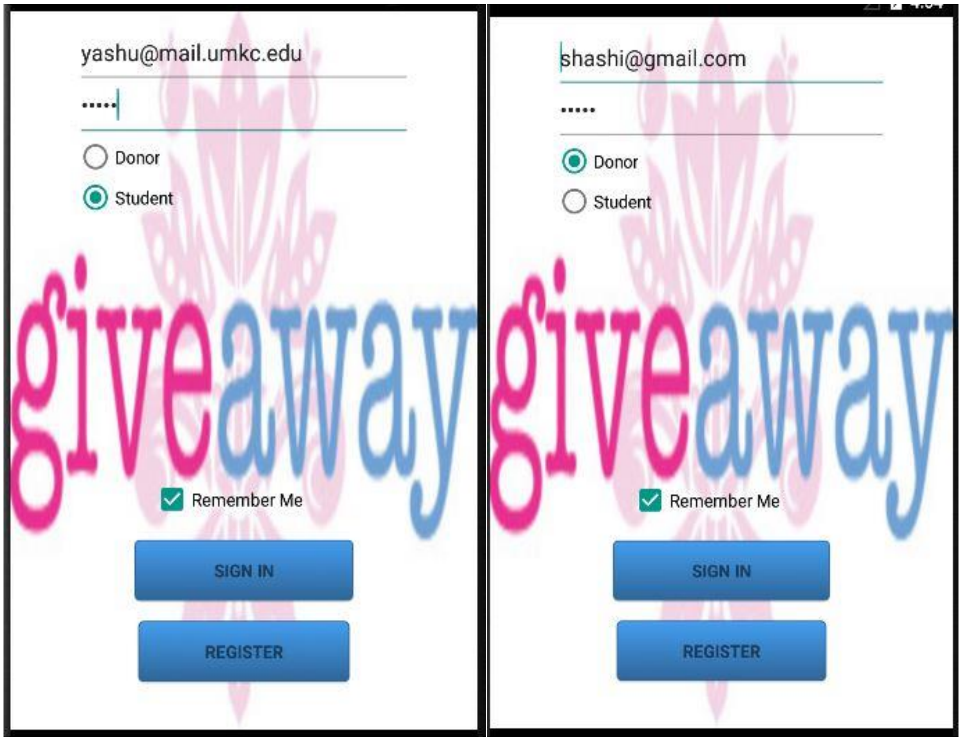
Guidelines:

- Download our application “Give Away” and install it in your Android mobile device.
- By clicking on the Give Away application on the user screen of the mobile device, our application gets launched.
- There will be screens for login and registration. One can login either as a Donor or a Student.
- If logged in as a Donor, he can contribute the items to his screen which he is willing to donate. He can also view those items that are subscribed by students. Depending on the distance and time of travel between the student and the donor, the donor can approve the item from his end.
- If logged in as a Student, he can view the list of items that are ready for donation. Student can subscribe for an item of his choice. Depending on the donor’s approval, the student will be confirmed with the item he has selected.
- Finally, you can logout of the application and close it.

Sample Execution:

The following are the screenshots of our entire application. It starts off with the user login screen.

Login Screen of Donor and Student:


yashu@mail.umkc.edu

•••••

○ Donor

● Student

☑ Remember Me

shashi@gmail.com

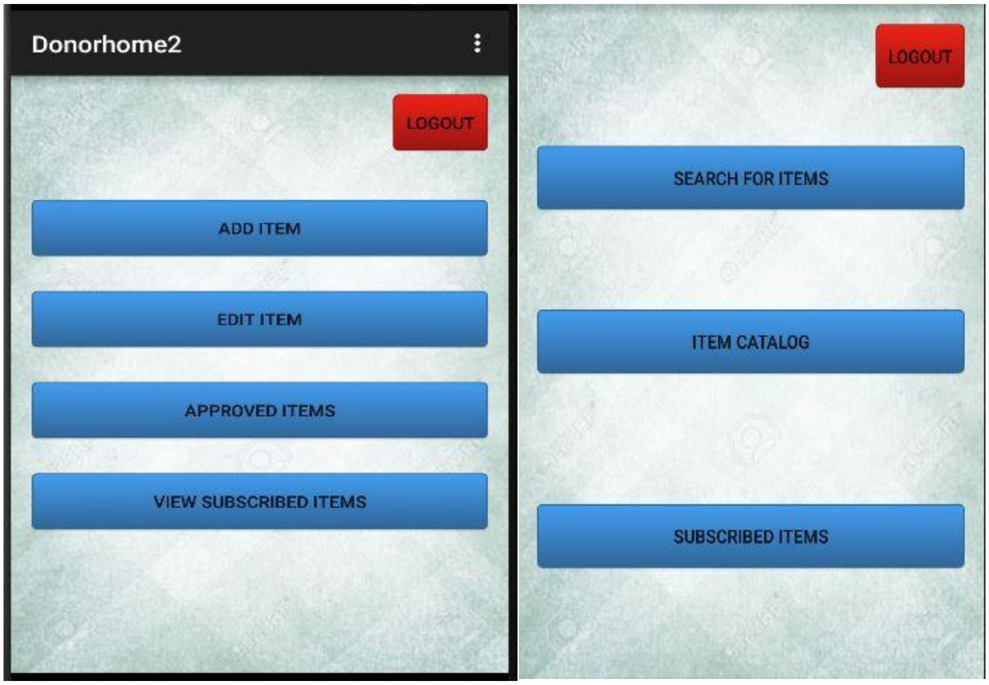
•••••

● Donor

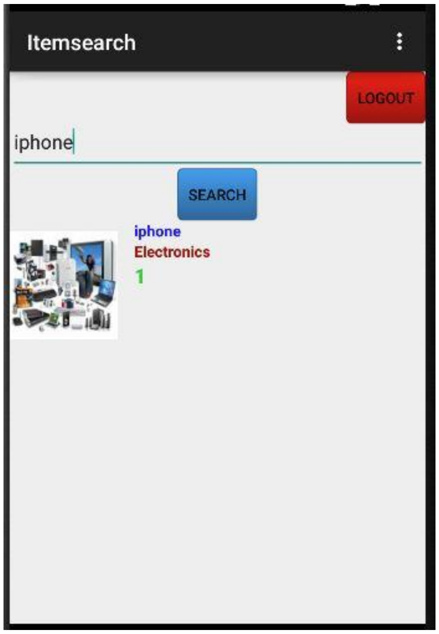
○ Student

☑ Remember Me

DashBoard Screens of Donor and Student

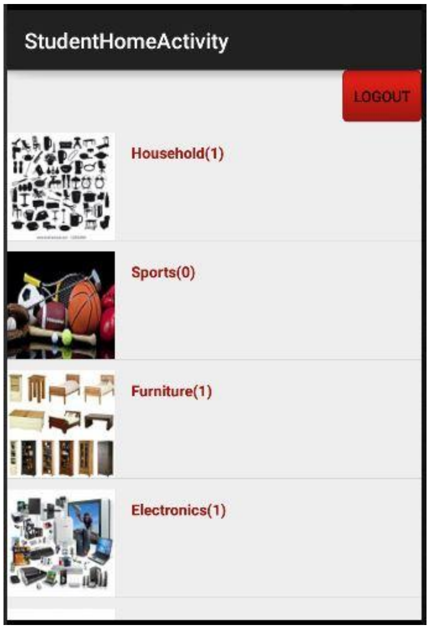


Student Screens:



iphone

iphone
Electronics
1

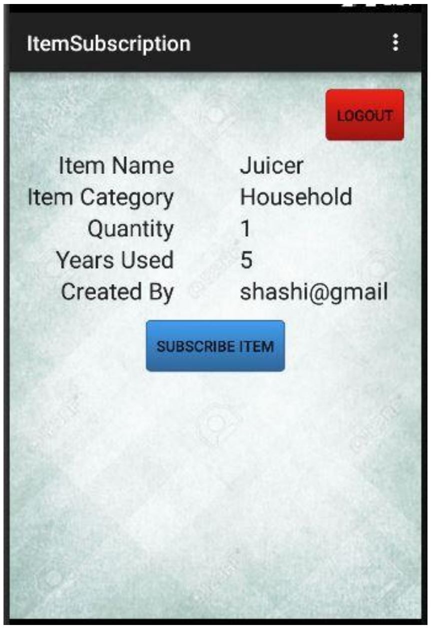


Household(1)

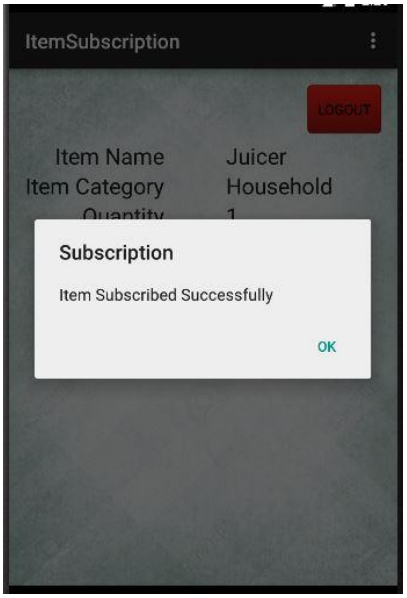
Sports(0)

Furniture(1)

Electronics(1)



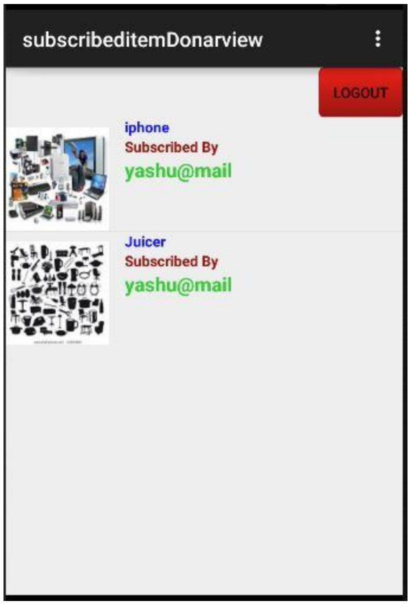
Item Name	Juicer
Item Category	Household
Quantity	1
Years Used	5
Created By	shashi@gmail



Item Name	Juicer
Item Category	Household
Quantity	1

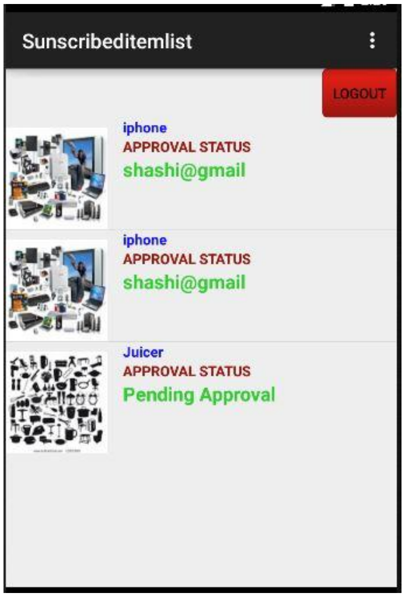
Subscription

Item Subscribed Successfully



iphone
Subscribed By
yashu@mail

Juicer
Subscribed By
yashu@mail

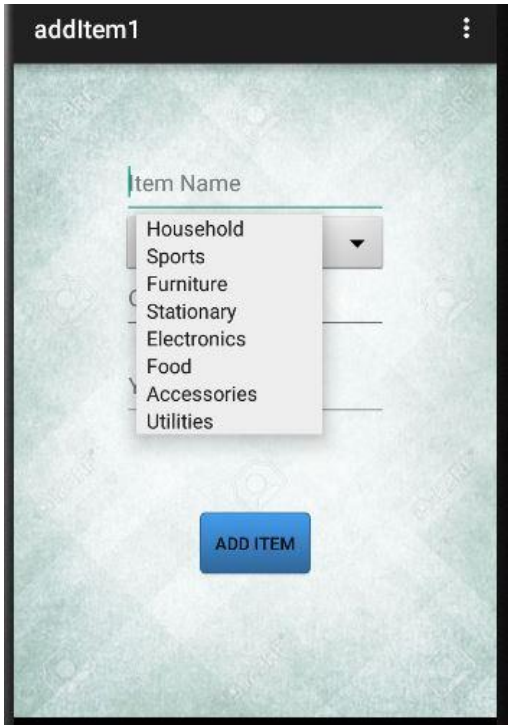


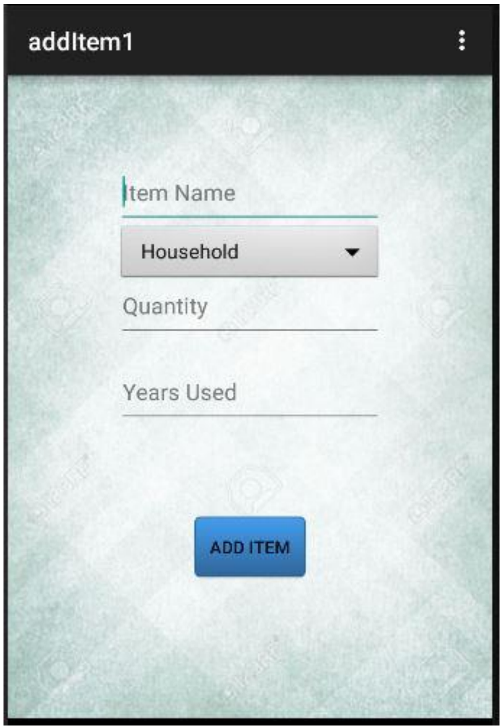
iphone
APPROVAL STATUS
shashi@gmail

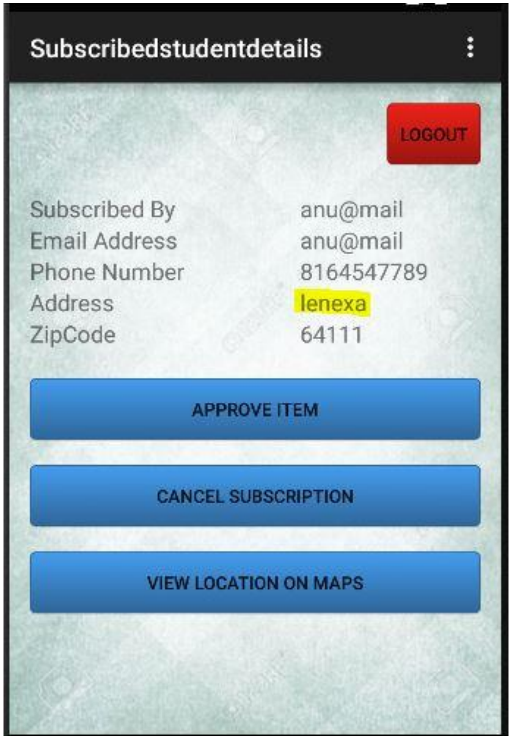
iphone
APPROVAL STATUS
shashi@gmail

Juicer
APPROVAL STATUS
Pending Approval

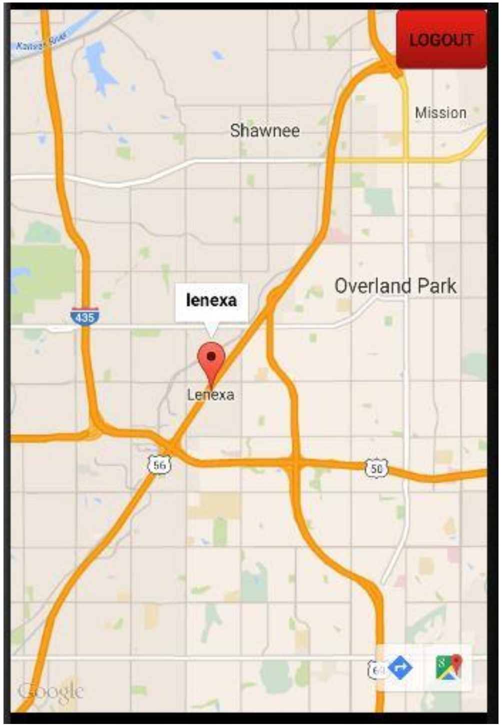
Donor Screens:







Subscribed By	anu@mail
Email Address	anu@mail
Phone Number	8164547789
Address	lenexa
ZipCode	64111



Troubleshooting:

If you find any issues with the installation, kindly remove the GiveAway apk, download it again and install it.

Project Management

As part of this project, we have used an agile tool Scrum do, to manage our work, contributions from the individual team members, time lines etc. One can access this from the below scrum do link:

https://www.scrumdo.com/projects/project/giveaway/summary

The following are our team members contributed individual for the project,

1. Sashidhar Reddy Gowra
2. Venkataramana Yashwant Kumar Palisetty
3. Ravi Kanth Devanaboyina
4. Anudeep Reddy Gujjula

Project Proposal

Project Goal and Objectives:

- **Motivation:**

  In the modern world, many students are flying to different countries for their higher education, which requires many basic things to start a new life. The students might not immediately afford for household things. Meanwhile in developed countries like United States, there are so many non-profitable organizations and even individuals who donate household items for International students. We need a platform to connect these newly arrived international students and people who are ready to give away the household items.

- **Significance:**

  Makes a student life easy in a new place by reducing his expenses to buy basic things.

  A donor can easily find people for donating their items.

- **Objectives:**

  Main intention of this project is to automate the process of giveaway where a donor can easily donate the items and a new student can easily grab the required items.

- **System features:** Using this application,
  - A donor can post his interested set of things to donate.
  - Students can search for required items and can find the list of donors.
  - Students can order the set things required and share their location for delivery.

- **Technologies to be used:**

  Android SDK, Web Services, jQuery, JavaScript, CSS.

- **Related Work:**

  Ask & give App in AppStore which is traditional giveaway app but it is not university specific and any user can grab the items.

  There are some web applications like ellentv.com and stacksocial.com which offer some items at discounted price in the name of Giveaway but not for free.

- **Backup Project:**

  Automation of Airport pickups for students through a mobile application. Whenever a new student lands in a particular country and cannot afford a cab to reach destination can interact through this application and can find someone who is offering rides from airport at some fare or for free.

- **Bibliography:**

  http://www.ellentv.com/giveaways/

  http://www.askandgiveapp.com/

Project Plan

Introduction:

These days, students travel to different countries in order to pursue their higher studies. These students face a lot of difficulties in getting adjusted to the new environment, cross culture and in various other aspects. The major problem they face is to gather the household items in the initial stages. The basic household items can be arranged by themselves, but when it comes to the point of major ones, it is difficult for them to manage their money on these large household items. In well developed countries like US and UK, there are some non-profit organizations which donate household items to the newly arrived students. But, it is a bit difficult for the students to get in touch with these organizations. So, we need a tool, which can bring these organizations and the students closer.

Project Goal and Objectives:

- **Overall Goal:** Developing an online system, where the students can directly interact with the non-profit organizations or individuals who are willing to donate the household items.
- **Specific Objectives:** The specific objective of this project "Give Away" is restricted only to students, where the students can login to the online portal using their University mail-id to access the catalogue displayed by the non-profit organizations or any particular individual. The students can select the items and request them for delivery.
- **Significance:** Till date, we have seen many traditional give-away events, where students have to travel to a designated place, select the items and carry them back home. Also, there are few online portals with the same concept, but the items are not free of cost.

Project Background and Related work:

There are quite a few applications of this kind, but the major difference between them and our project is, our application is restricted to University students and also cost free.

For example, "Ask and Give" application is one such application. Any person can login to this app and can grab the item. So, there is a risk that the application can be misused, which can't be ignored.

http://www.askandgiveapp.com/

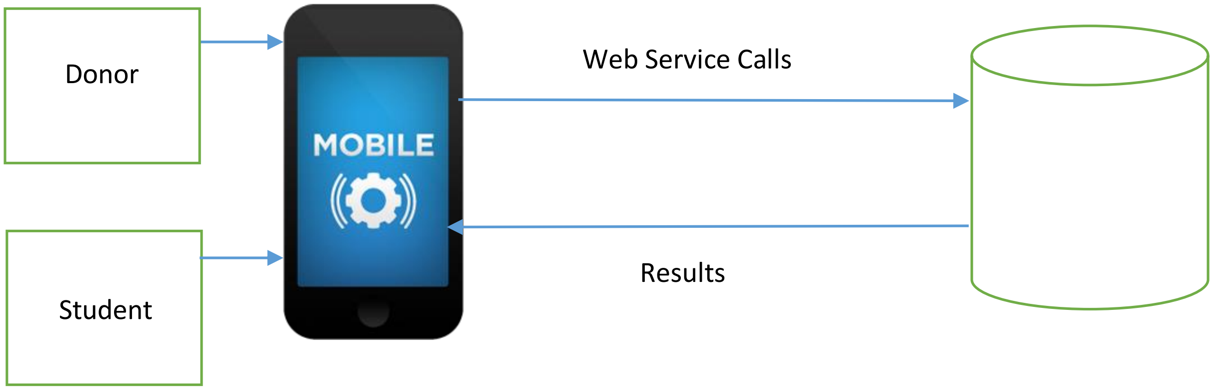
The other such examples are "ellentv.com" and "stacksocial.com". But these portals sell the items, instead of giving them for free.

http://www.ellentv.com/giveaways/

Our project is exclusively for University students, who have an official University specific mail-id (Ex: xxx@yyy.**edu**) by using which they can login.

Technological & Architectural Requirements:

- An application for automation of give-away is required.
- Android App will present the presentation layer for the user.
- A relational database handles the persistence layer.
- The database used is Oracle 11g.
- The application will receive the data from the database using restful web services.
- The application runs seven days a week, twenty four hours per day.



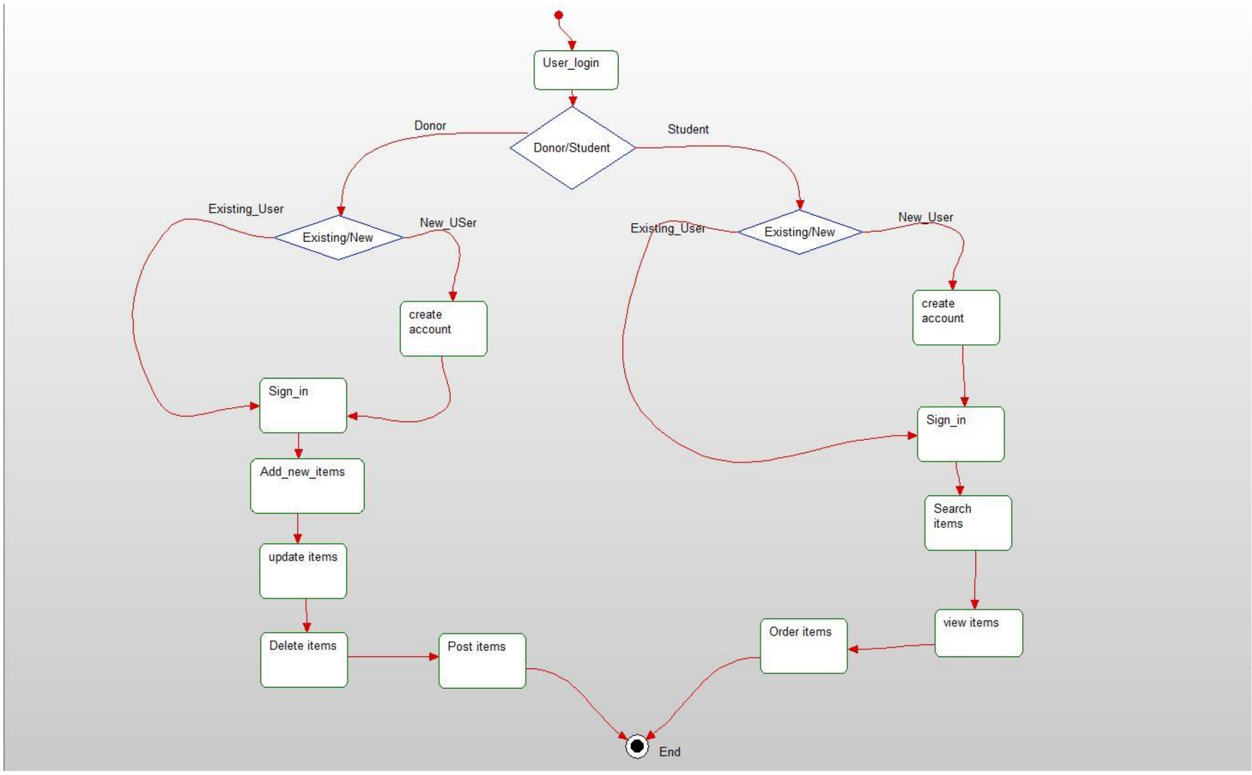
Proposed system:

- **Requirement Specification:**
  - **Functional Requirements:** In this system there are two set of users like people who like to give the items and people who like to grab the items. This system has two separate login pages for different set of users like donor and grabbers. In donor login, the donor might be an already existing user or a new user, if he is a new user he need to create an account for himself with a username and password where username must be unique. It is same in grabber login too. Once an account is created for a donor he can add his list of items with name, description and corresponding images of the items to be donated and post them to the system. Once the items have been posted to the system, people who like to grab the items can view them on the system and can order the items. One more feature of the system is like the grabbers can not only look for set items of items which are posted, they can also search for interested set of items which are posted for a giveaway.
  - **Non-functional Requirements:** The main purpose of the system is to make international students arriving at campus feel comfortable in adjusting to new environment by providing them with household items which are kept for giveaway. So, the target set of grabbers is international students and the system validates the set of users by their email address, as the students will have the education domain email address which ends with ‘edu’.The system should be consistent in blocking the items

whenever a user orders the item, so that other users cannot have access to that set of items.

- **Technical Requirements:** As the project is a mobile application we would go for Android SDK to develop a basic android application and use HTML, CSS and JQuery in the front end and for data storage we are using Oracle 11g and would like to develop a web service to access data from data storage.

Workflow Analysis:



- **Framework Specification:**

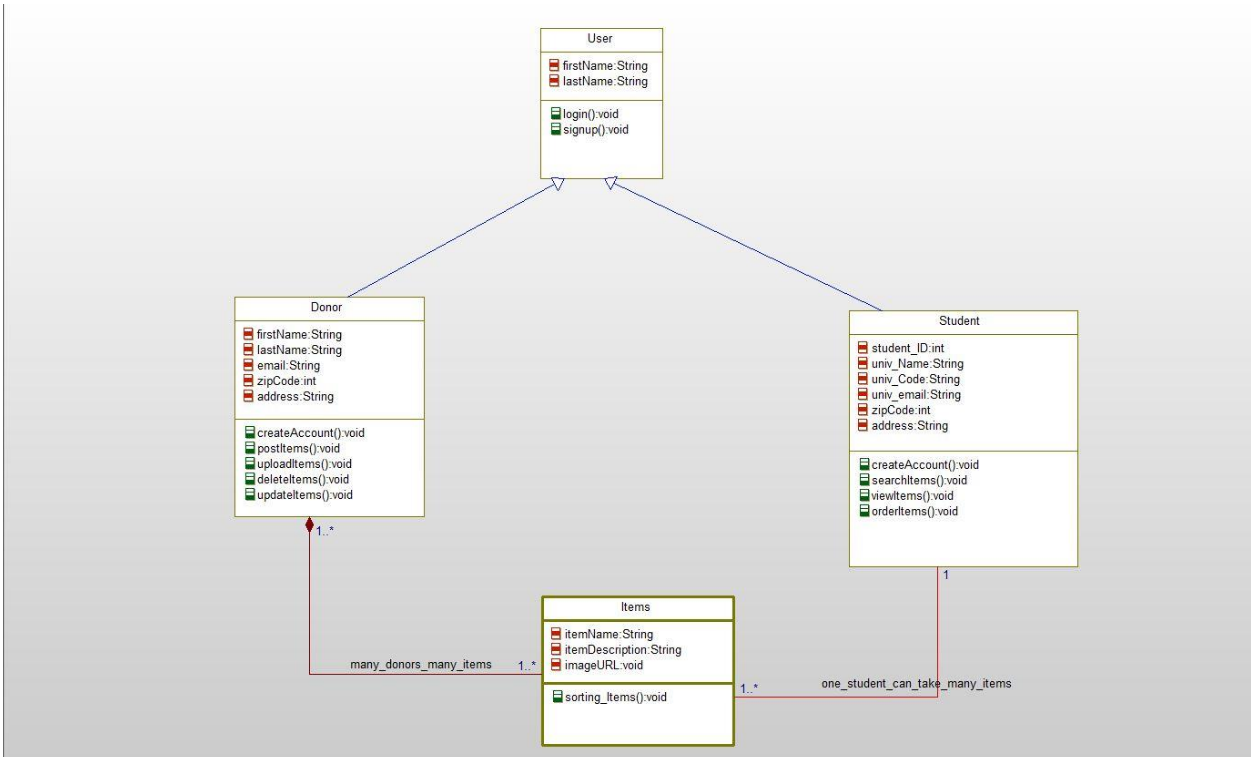
- **Assumptions and Principles:** As the application is intended for students we assume a grabber
as a student and verify the same using their email address. The other user set is assumed to be
an individual or a non-profit organization willing to donate items for the sake of students.
- **Methodologies and Algorithms:** The intended method of application development is through agile methodology where we go for dynamic requirements and make the system flexible to incorporate the dynamic changes. The application has to perform certain sorting and searching operations when a user like to sort the items based on the date of posting and search for a specific item. For sorting the items a quick sort approach is followed and or searching an item binary search is being followed.

- **System Architecture design:**

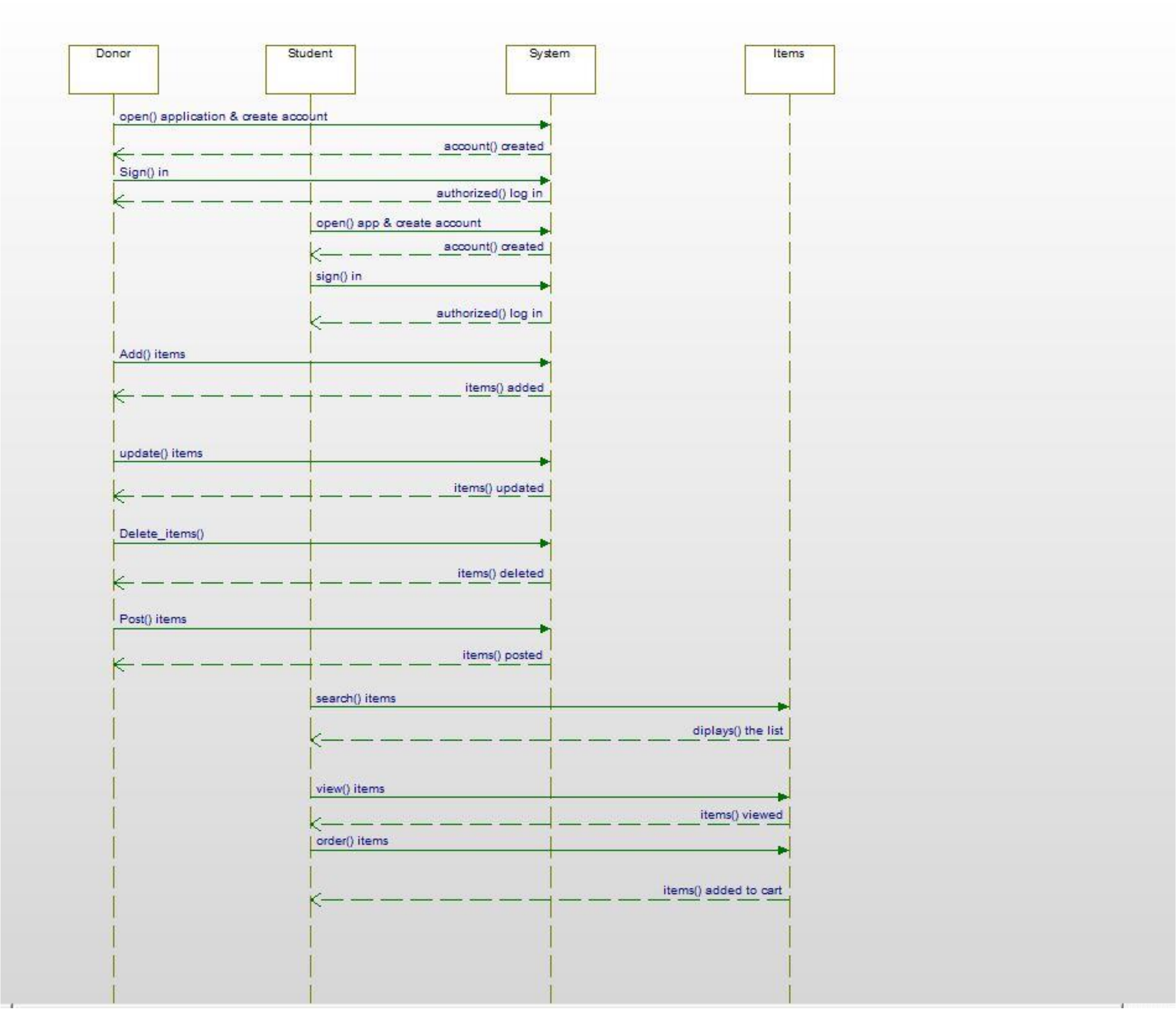
- **System Specification:**

  - **Existing Service:** This system uses Google maps API for locating the user address location
    whenever he tries to order a product for delivery.
  - **New Service:** The application requires more and more images data to be populated on screen, we like to build a Restful service for retrieving data from database and populating the data on the mobile screen.

Class Diagram



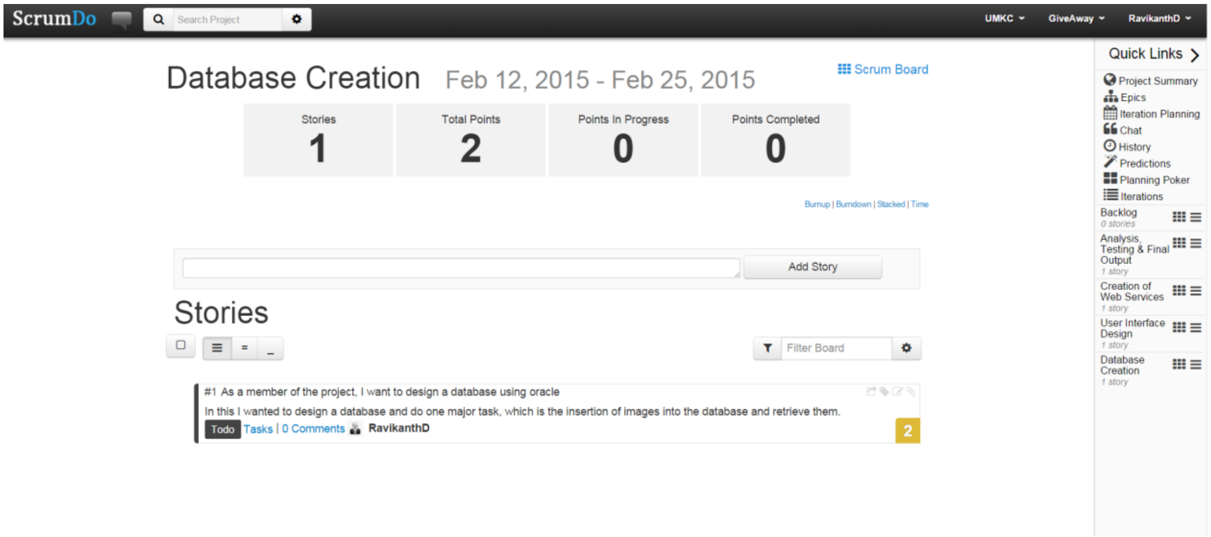
Sequence Diagram



- **Service Specification**
  - **Operational description:** The service should take the input data from the mobile application and should insert the same in database. The other thing is that stored data should be sent back to mobile screen to populate the screen.
  - **Input/output service:** The input service is like taking user data from the create account page or login page validate them and store them in database. The output service should get back the stored data whenever required.
  - **Constraints/expectations:** This application deals with much of user data like student data and items data and also the donors data, items will have images to be uploaded to the system whenever a donor likes to donate the item. So once a student like to grab an item he needs to order that item and that item will be in blocked status and will be removed once the item has been delivered to student. We need to maintain lock on the status of item and whenever a new session begins for a user so that a blocked item should not be visible to other logged in users.

    - **Service flow**: The flow of service calls is like getting data from user input storing it, modifying the data, deleting the data and displaying the data back to mobile screen.
    - **Priorities**: The main goal of this application is to help students and finding some useful things and we need to verify the users and only students should be able to login and search for required items.
- **Design of Mobile Client:**
    - **Features:** The mobile client will have the required UI pages to make a user login to the application look for items or even post some items for giveaway. Separate login for people who look donate items and people who look to grab items and also student verification for people who would like to grab items.
    - **Styles:** As the application has many images of items which are posted in the application to donate, we need to use effective CSS styles to separate text and images and to make them fit to the screen without overlapping. Bootstrap gives an effective way CSS styling where we can maintain a single stylesheet file and include all the style elements in that sheet with references.
    - **Technologies:** For the UI screen design we would go with HTML5, JQuery as they provide some advanced features for UI design and scripting.

Plan by Service (ScrumDo):



Database Creation Feb 12, 2015 - Feb 25, 2015

Stories	Total Points	Points In Progress	Points Completed
1	2	0	0

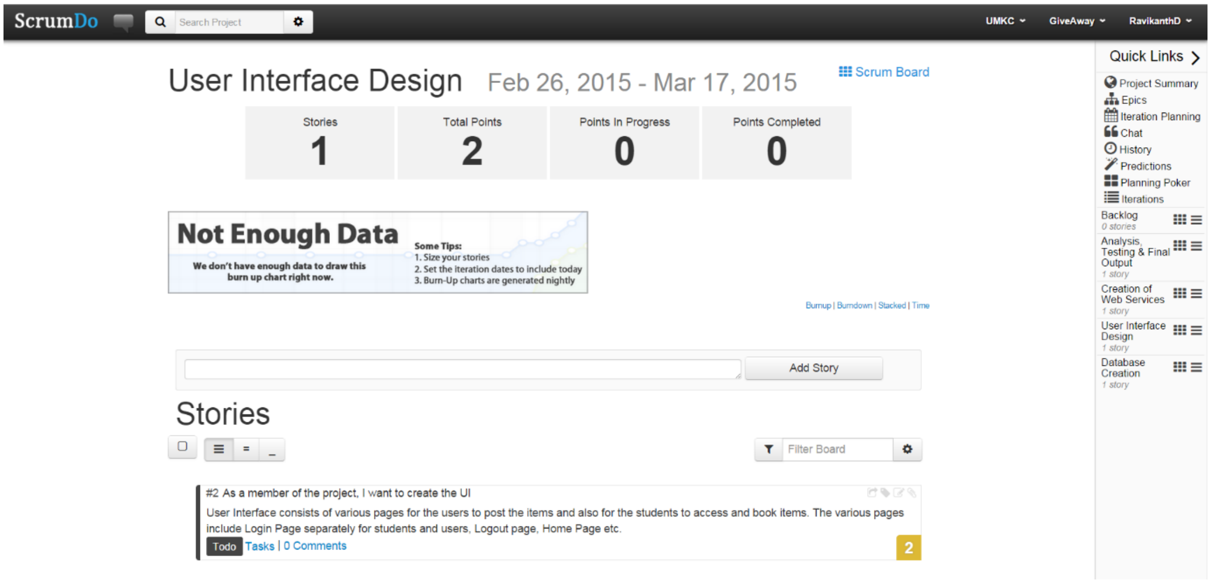
Stories

#1 As a member of the project, I want to design a database using oracle

In this I wanted to design a database and do one major task, which is the insertion of images into the database and retrieve them.

Todo Tasks | 0 Comments RavikanthD 2

Backlog	0 stories
Analysis, Testing & Final Output	1 story
Creation of Web Services	1 story
User Interface Design	1 story
Database Creation	1 story



User Interface Design Feb 26, 2015 - Mar 17, 2015

Stories	Total Points	Points In Progress	Points Completed
1	2	0	0

Not Enough Data

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

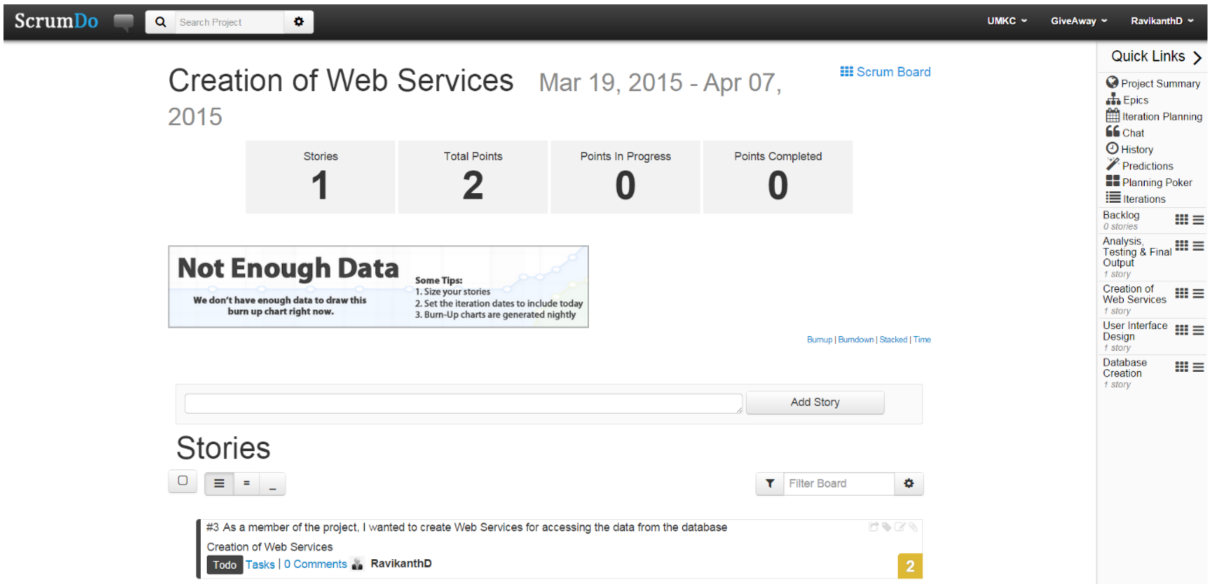
Some Tips:
1. Size your stories
2. Set the iteration dates to include today
3. Burn-Up charts are generated nightly

Stories

#2 As a member of the project, I want to create the UI

User Interface consists of various pages for the users to post the items and also for the students to access and book items. The various pages include Login Page separately for students and users, Logout page, Home Page etc.

Todo Tasks | 0 Comments 2



Creation of Web Services Mar 19, 2015 - Apr 07, 2015

Stories	Total Points	Points In Progress	Points Completed
1	2	0	0

Not Enough Data

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

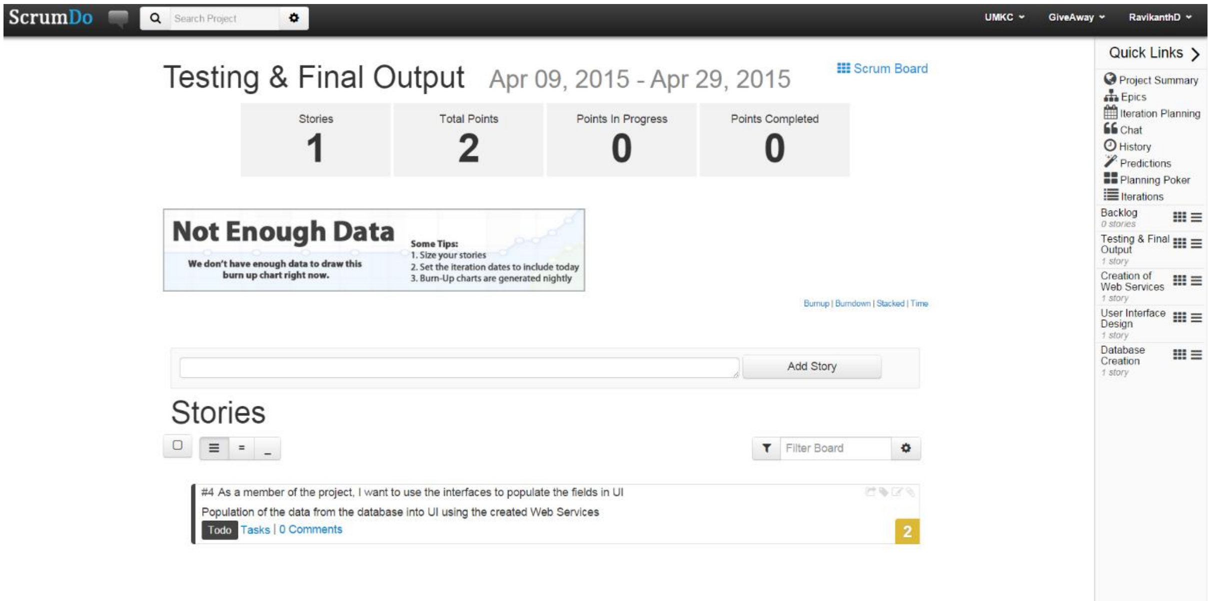
Some Tips:
1. Size your stories
2. Set the iteration dates to include today
3. Burn-Up charts are generated nightly

Stories

#3 As a member of the project, I wanted to create Web Services for accessing the data from the database

Creation of Web Services

Todo Tasks | 0 Comments RavikanthD 2



Testing & Final Output Apr 09, 2015 - Apr 29, 2015

Stories	Total Points	Points In Progress	Points Completed
1	2	0	0

Not Enough Data

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

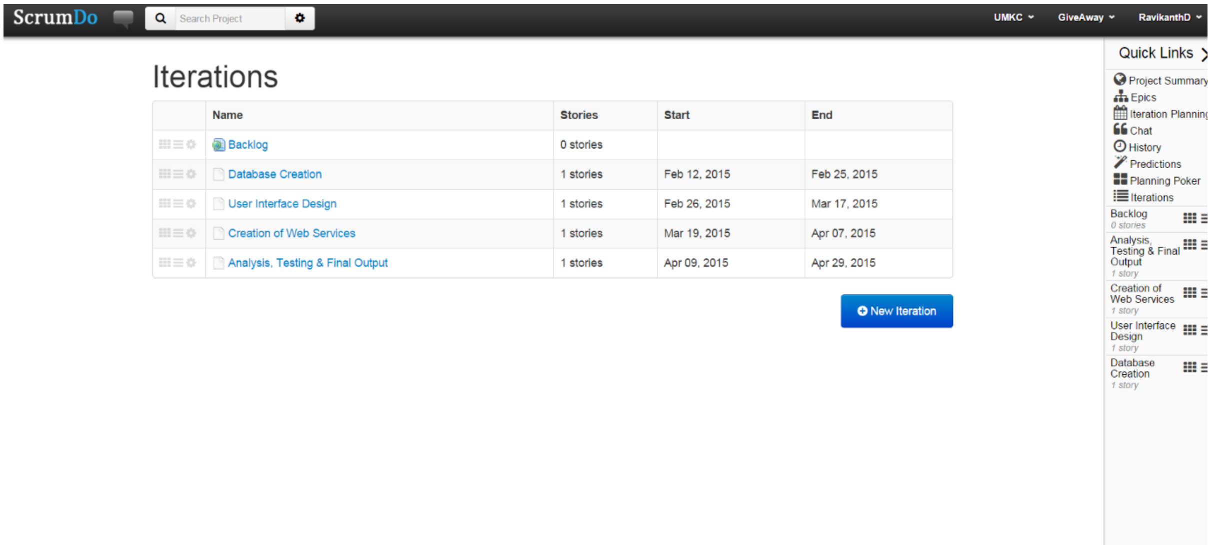
Some Tips:
1. Size your stories
2. Set the iteration dates to include today
3. Burn-Up charts are generated nightly

Stories

#4 As a member of the project, I want to use the interfaces to populate the fields in UI

Population of the data from the database into UI using the created Web Services

Todo Tasks | 0 Comments 2



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 17, 2015
Creation of Web Services	1 stories	Mar 19, 2015	Apr 07, 2015
Analysis, Testing & Final Output	1 stories	Apr 09, 2015	Apr 29, 2015

Risk management:

- **Technology and Architectural Requirements:** Here in this application we are using Android SDK as mobile platform to build the application with HTML5, CSS and JQuery. The UI part does not pose any risks as it is a straight forward android application which gets its data from web services**.** The data storage and retrieval part will be challenging part and may pose a five risks while choosing a web service to be used to get the data. Architecture wise the application is very clear and might be risk free as it deals with specific set of users. There would not be much burden on the system in managing the user traffic.

Bibliography:

http://www.askandgiveapp.com/

http://www.ellentv.com/giveaways/

http://developer.android.com/sdk/index.html

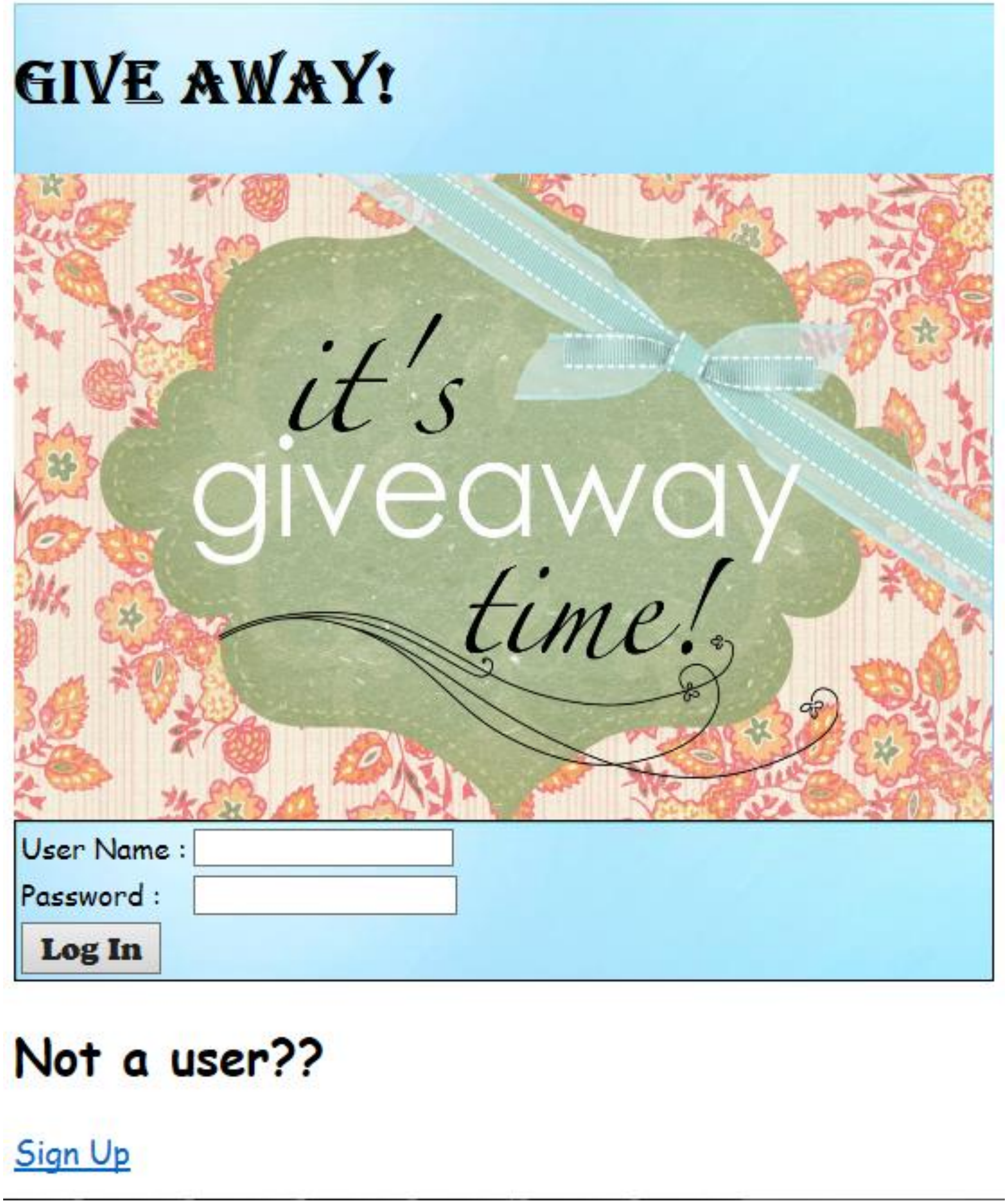
http://ws.apache.org/

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

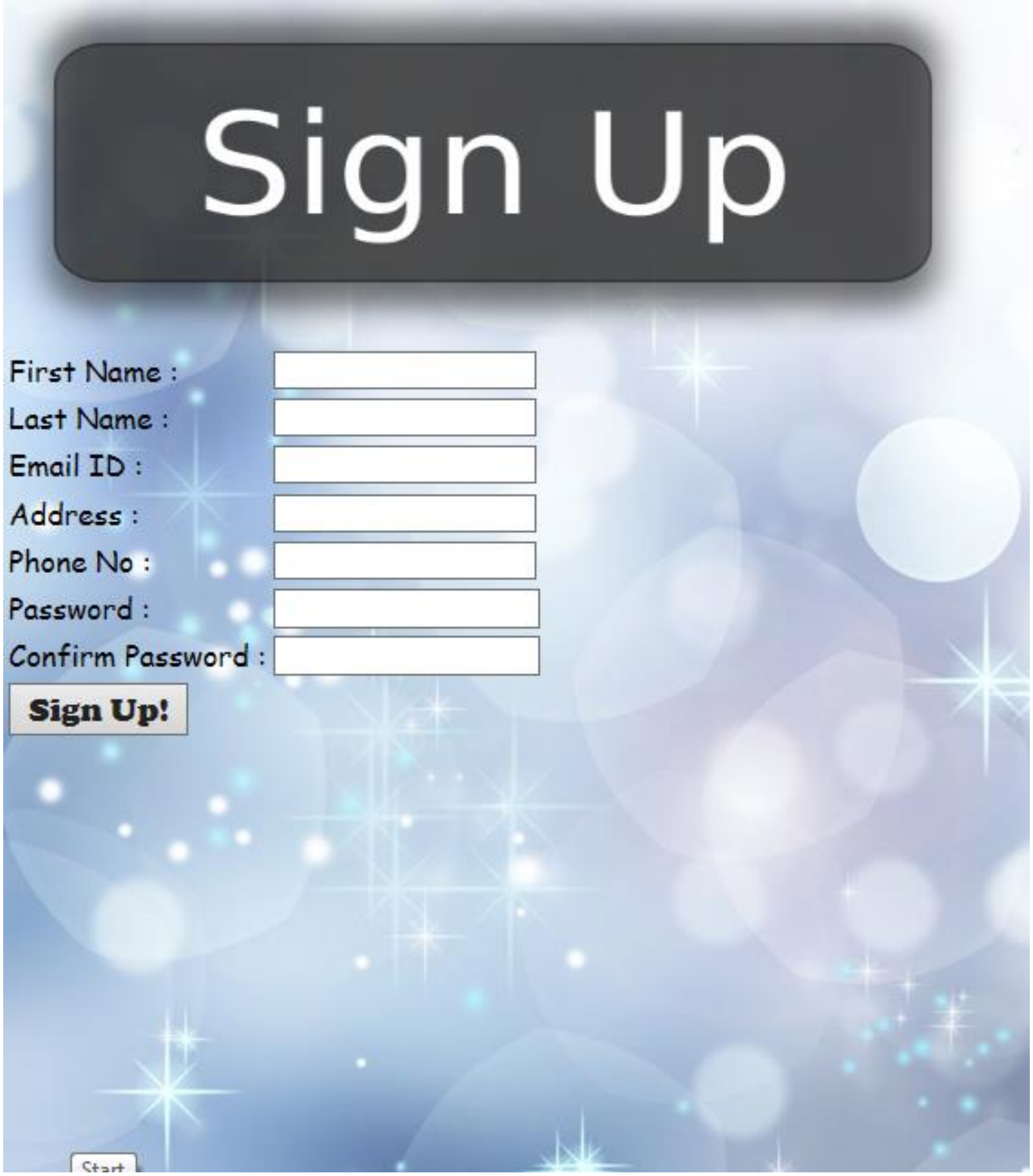
Project Increment 1

UI Design:

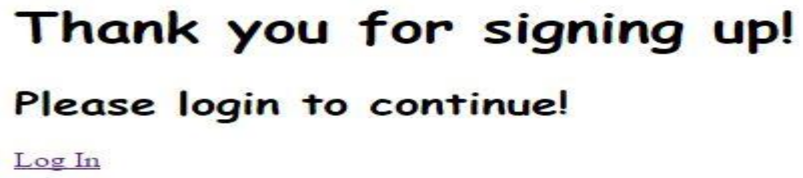
- **User Login Page:** The Login page is the first page of our application where a user needs to give his username and password which he has created for himself in user registration page.This page includes basic html elements like textboxes, buttons, images and certain CSS styles and some scripting functions to validate the user.



- **User Registration Page:** This page is like signup page or user registration page which pops up whenever a user clicks on the signup button on the login page. A user must be a registered user
  If he wants to use the application. A user might be a donor or a taker and he needs to fill the following fields like First name, last name, email address, phone number and he can choose a password for himself. On successful registration a success message will be popped on the screen and the registered user can navigate to login page and can give his credentials for logging in.



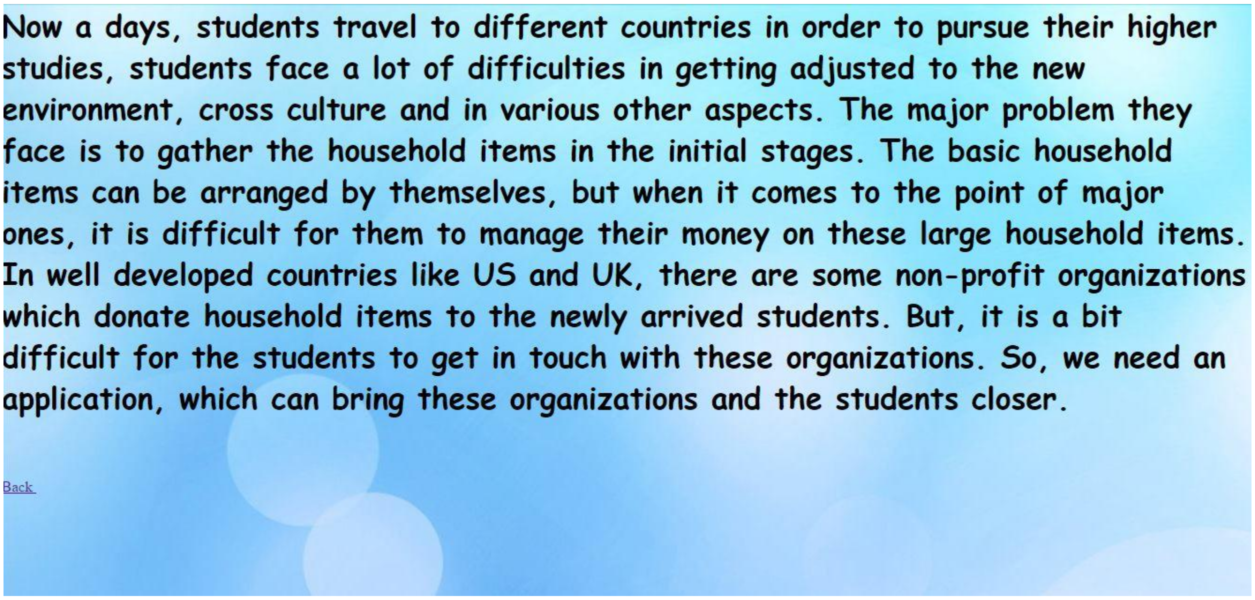
- **Registration Success Page:** This page will have the success message and a button to navigate the user back to login page.



Thank you for signing up!

Please login to continue!

- **Description Page:** This page is like the description page for purpose of the application and intended users who may benefit using this application.



Now a days, students travel to different countries in order to pursue their higher studies, students face a lot of difficulties in getting adjusted to the new environment, cross culture and in various other aspects. The major problem they face is to gather the household items in the initial stages. The basic household items can be arranged by themselves, but when it comes to the point of major ones, it is difficult for them to manage their money on these large household items. In well developed countries like US and UK, there are some non-profit organizations which donate household items to the newly arrived students. But, it is a bit difficult for the students to get in touch with these organizations. So, we need an application, which can bring these organizations and the students closer.

DATABASE Design:

- **Creation of Tables and Identification of Constraints:**
  We have created 4 tables namely, Authentication, Registration, GiverItemTable, GrabberItemTable discussed briefly as follows,

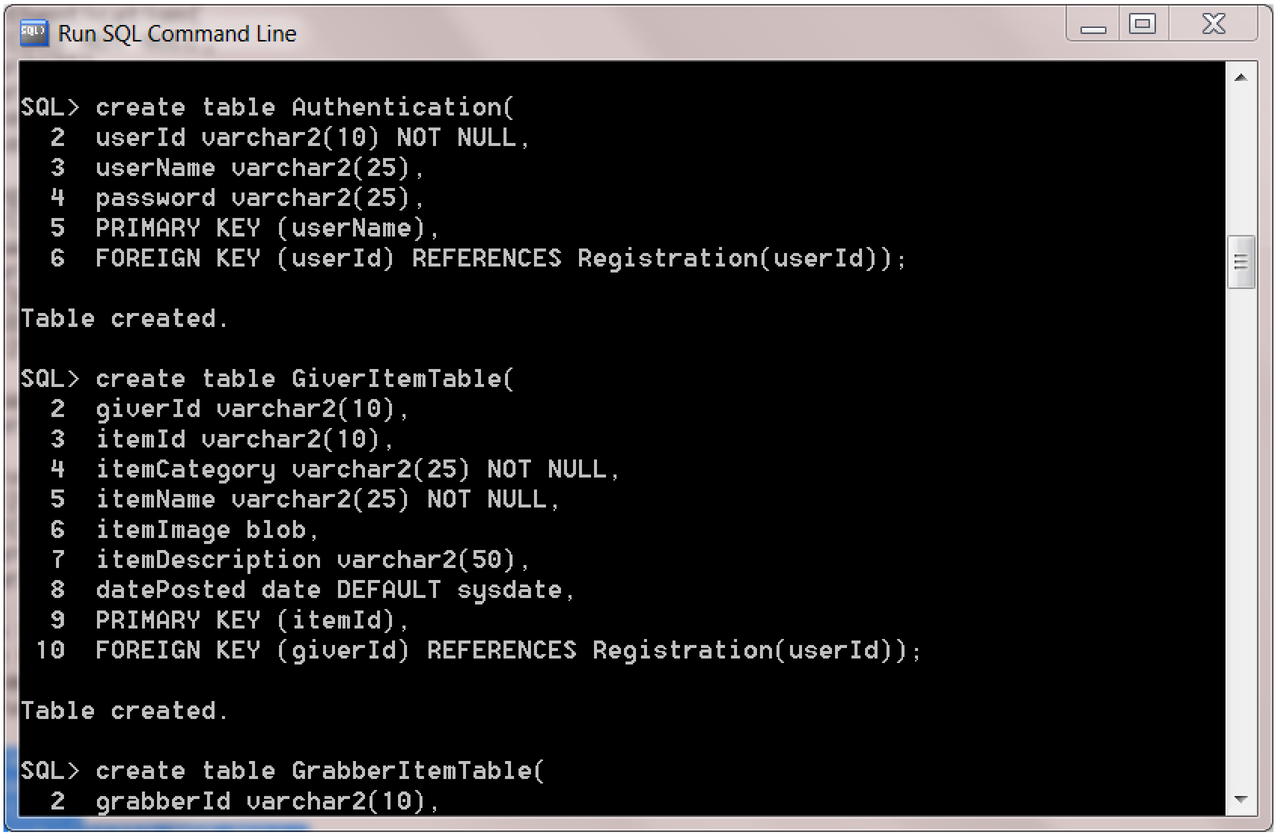
Authentication Table: This table has userId, userName and password columns and the main purpose for the creation of this table is to store the userName and passwords of the users. UserId is autogenerated. userName is identified as Primary Key and userId of this table refers to userId of Registration Table.

Registration Table: This table has userId, firstName, lastName, email, mobile, address, dob, user_flag and the main purpose for the creation of this table is to store the details with which the Student/Grabber will be registered in the application. userId in this table is identified as the primary key. User_flag specifies whether the user is a grabber or a giver.

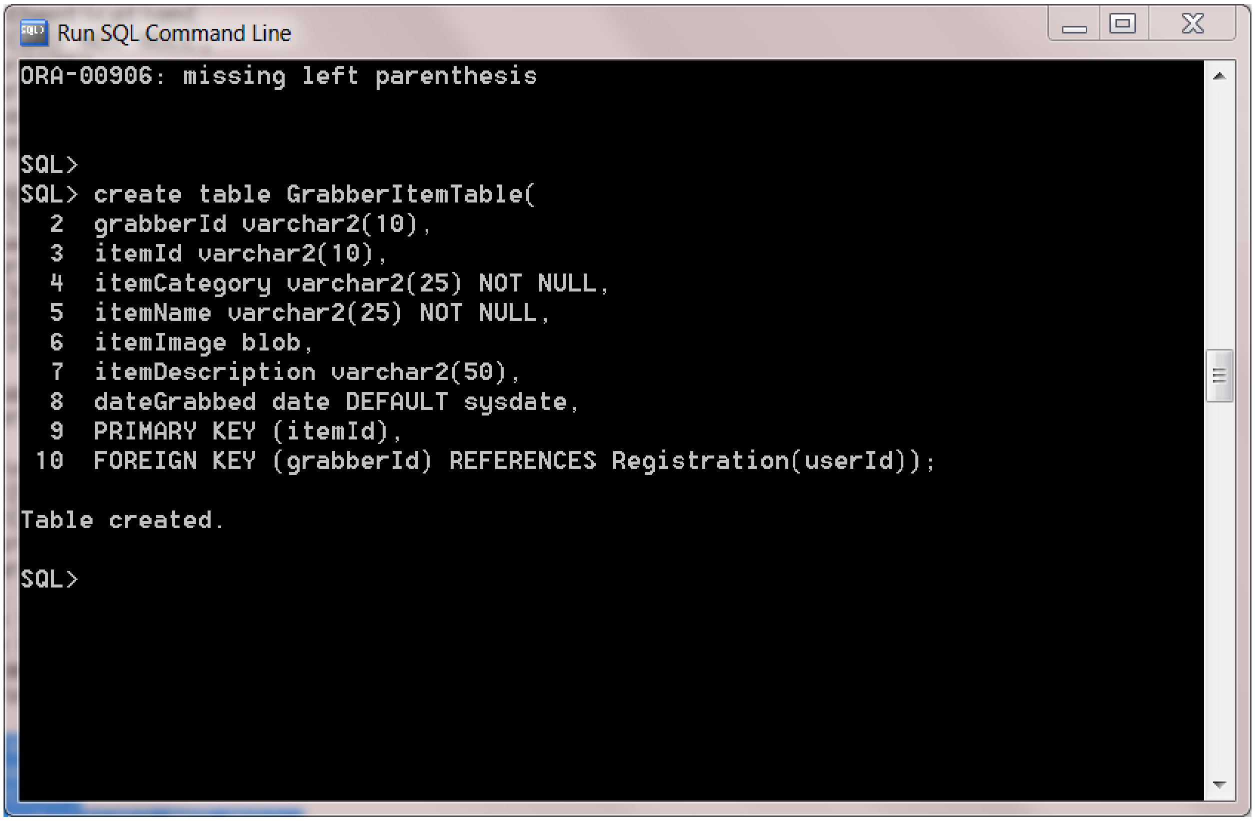
GiverItemTable: This table has giverId, itemId, itemCategory, itemName, itemImage, itemDescription, datePosted. The main purpose of this table is to store the details of the items that the people are willing to donate. Here, itemId is the primary key, and giverId refers to the userId of Registration Table.

GrabberItemTable: This table has grabberId, itemId, itemCategory, itemName, itemImage, itemDescription, dateGrabbed. The main purpose of this table is to store the details of the items that the students take. Here, itemId is the primary key and grabberId refers to the userId of the Registration table.

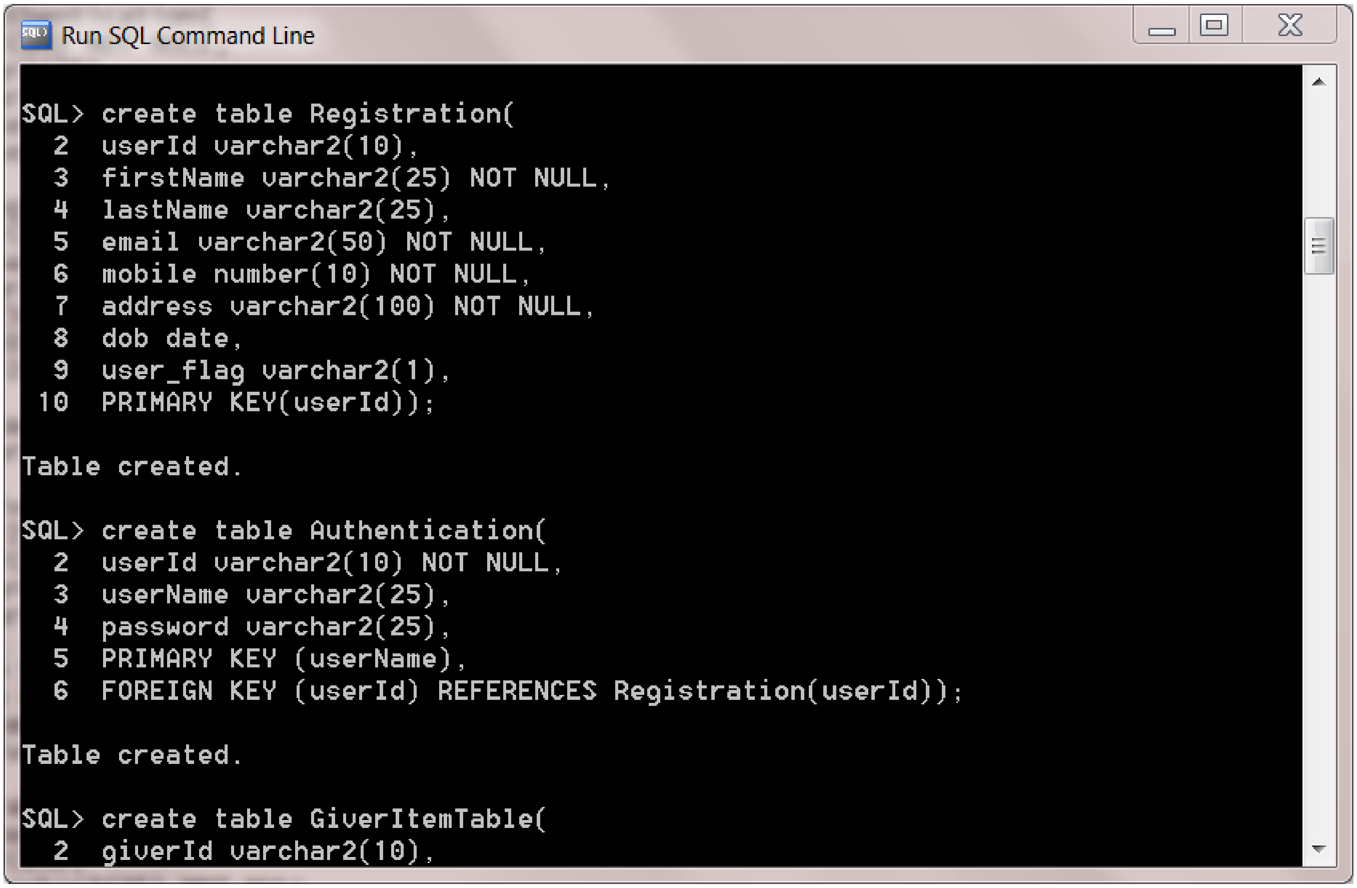
Screenshots:



```
SQL> create table Authentication(
  2  userId varchar2(10) NOT NULL,
  3  userName varchar2(25),
  4  password varchar2(25),
  5  PRIMARY KEY (userName),
  6  FOREIGN KEY (userId) REFERENCES Registration(userId));

Table created.

SQL> create table GiverItemTable(
  2  giverId varchar2(10),
  3  itemId varchar2(10),
  4  itemCategory varchar2(25) NOT NULL,
  5  itemName varchar2(25) NOT NULL,
  6  itemImage blob,
  7  itemDescription varchar2(50),
  8  datePosted date DEFAULT sysdate,
  9  PRIMARY KEY (itemId),
 10  FOREIGN KEY (giverId) REFERENCES Registration(userId));

Table created.

SQL> create table GrabberItemTable(
  2  grabberId varchar2(10),
```



```
ORA-00906: missing left parenthesis


SQL>
SQL> create table GrabberItemTable(
  2  grabberId varchar2(10),
  3  itemId varchar2(10),
  4  itemCategory varchar2(25) NOT NULL,
  5  itemName varchar2(25) NOT NULL,
  6  itemImage blob,
  7  itemDescription varchar2(50),
  8  dateGrabbed date DEFAULT sysdate,
  9  PRIMARY KEY (itemId),
 10  FOREIGN KEY (grabberId) REFERENCES Registration(userId));

Table created.

SQL>
```

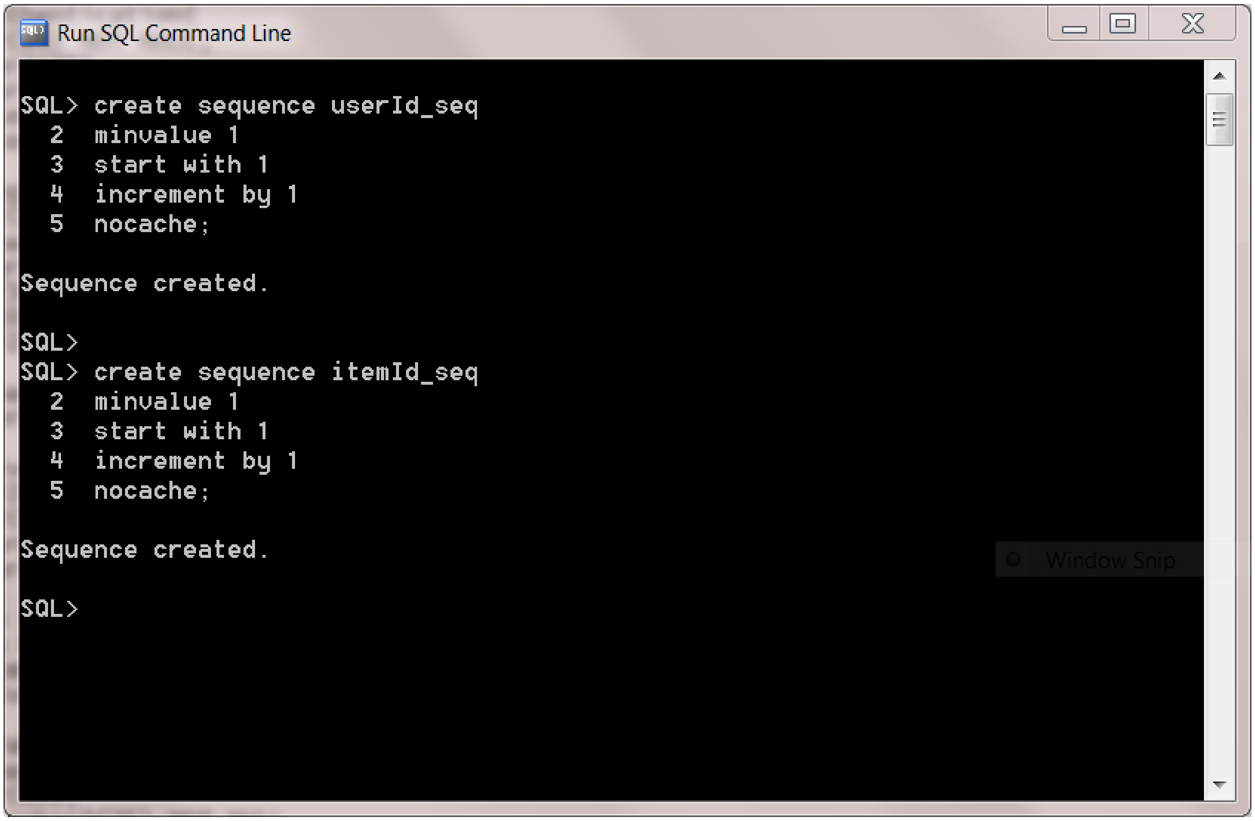


```

SQL> create table Registration(
  2  userId varchar2(10),
  3  firstName varchar2(25) NOT NULL,
  4  lastName varchar2(25),
  5  email varchar2(50) NOT NULL,
  6  mobile number(10) NOT NULL,
  7  address varchar2(100) NOT NULL,
  8  dob date,
  9  user_flag varchar2(1),
 10  PRIMARY KEY(userId));

Table created.

SQL> create table Authentication(
  2  userId varchar2(10) NOT NULL,
  3  userName varchar2(25),
  4  password varchar2(25),
  5  PRIMARY KEY (userName),
  6  FOREIGN KEY (userId) REFERENCES Registration(userId));

Table created.

SQL> create table GiverItemTable(
  2  giverId varchar2(10),
```

- **Creating Sequences for Auto Increment:**

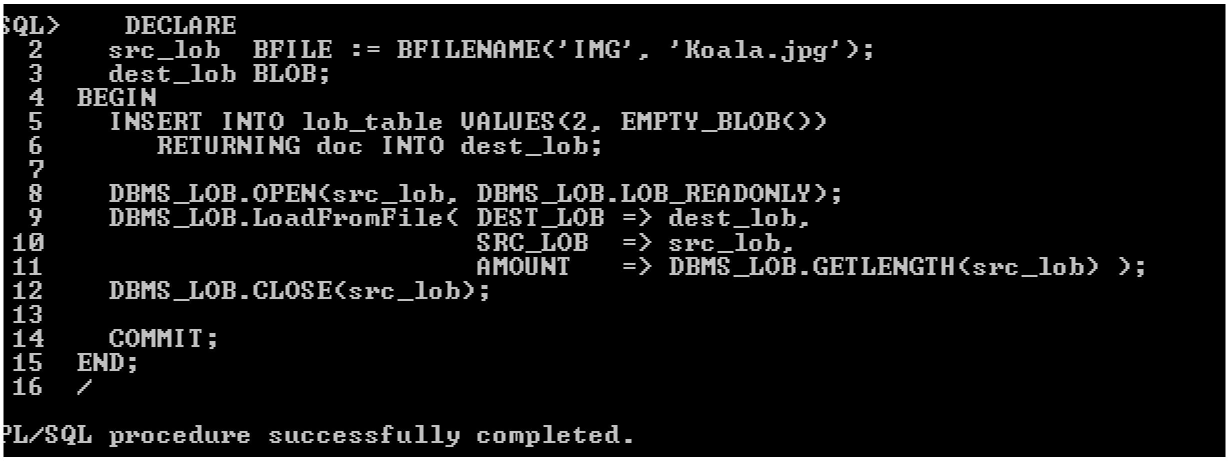
  For this application we have created two sequences for autogeneration logic for itemId and userId namely, userId_seq and itemId_seq.

  Screenshots:



```
SQL> create sequence userId_seq
  2  minvalue 1
  3  start with 1
  4  increment by 1
  5  nocache;

Sequence created.

SQL>
SQL> create sequence itemId_seq
  2  minvalue 1
  3  start with 1
  4  increment by 1
  5  nocache;

Sequence created.

SQL>
```

- **Converting Blob files to binary files and Inserting into Database:** When dealing with blob files, they need to be converted into binary files and should be stored in certain location with a filename and then we can insert the generated binary file into the required table. Here we have written a stored procedure for creating a binary file for each Blob file and inserting it into the database.



```
SQL>     DECLARE
       src_lob  BFILE := BFILENAME('IMG', 'Koala.jpg');
       dest_lob BLOB;
     BEGIN
       INSERT INTO lob_table VALUES(2, EMPTY_BLOB())
          RETURNING doc INTO dest_lob;

       DBMS_LOB.OPEN(src_lob, DBMS_LOB.LOB_READONLY);
       DBMS_LOB.LoadFromFile( DEST_LOB => dest_lob,
                              SRC_LOB  => src_lob,
                              AMOUNT   => DBMS_LOB.GETLENGTH(src_lob) );
       DBMS_LOB.CLOSE(src_lob);

       COMMIT;
     END;
     /

PL/SQL procedure successfully completed.
```

- **Retrieving the Blob files with Java Code**: Once a binary file is created and inserted into a table we need to access the file using a java code so that we can display the required image on HTML page. So here using Java we established a connection with Oracle Database and using the user credentials we have logged in and retrieved the inserted images and displayed them using MS paint. This is just a start up code for trying successful retrieval of inserted images into Database.

```
package hackathon;

import java.io.ByteArrayOutputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.Blob;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class Hackathon {
public static void main(String args[])
{

        Connection conn = null;
        String url = "jdbc:oracle:thin:@Shashi-PC:1521:XE";
        String driver = "oracle.jdbc.OracleDriver";
        String userName = "scott";
        String password = "tiger";
        ResultSet rs = null;
```

```
    try {
    Class.forName(driver).newInstance();
    conn = DriverManager.getConnection(url,userName,password);

    Statement stmt = conn.createStatement();
     rs =stmt.executeQuery("select doc from lob_table");
     Blob lob = null;
     while (rs.next()) {
     lob=rs.getBlob("doc");
}

    InputStream in1 = lob.getBinaryStream();
    ByteArrayOutputStream out1 = new ByteArrayOutputStream();
    OutputStream outputStream1 = new FileOutputStream("blobImage.png");

    int bufferSize = 1024;
    int length = (int) lob.length();

    byte[] buffer = new byte[bufferSize];

    while((length = in1.read(buffer)) != -1)
    {
     out1.write(buffer,0,length);
    }
    out1.writeTo(outputStream1);
    in1.close();
    conn.close();

     Process p1 =Runtime.getRuntime().exec("mspaint blobImage.png");

    }catch (Exception e) {
       e.printStackTrace();
       }

}
}
```

Implementation Status Report:

Work completed: UI Design for some of the screens and Database design.

- Description: In the UI part screens like user login page, user registration page, description page, registration success page have designed and with certain CSS styles and required JavaScript. In Database part the design of storage tables with required constraints have been done along with stored procedures to convert a blob image file into binary file and store them into table.

- Responsibility (Task, Person):
  Sashidhar Reddy Gowra-developed java code and stored procedure required for inserting and retrieving blob image from Database.
  Yashwanth Palisetty: UI Screen Design for Login and Success Pages.
  Ravikanth Devaboina: Creating tables and generating auto increment sequences to generate some unique ids.
  Anudeep Reddy Gujjula: UI Screen Design for Registration and Description pages.
- Time taken (#hours): UI Design: 12 hrs Database Design: 14 hrs.
- Contributions (members/percentage): Sashidhar Reddy Gowra 25%, Yashwanth Palisetty 25%, Ravikanth Devaboina 25%, Anudeep Reddy Gujjula 25%.

Work to be completed

- Description: Populating the UI screens with data from Database, Creating web services to get the data from Database, Using the already existing Google maps API for sharing the address location of a user, Unit Testing, System Testing, Testing the application on Android Devices
- Responsibility (Task, Person):
  Creating Web services, Hash Functions: Sashidhar Reddy Gowra, Ravikanth Devaboina.
  Populating screens, UI alignments: Yashwanth Palisetty, Anudeep Reddy Gujjula.
- Time to be taken (estimated #hours) 150 hrs.
- Issues/Concerns: Creating RESTfull Web services, Creating hash functions for password hiding, Android devices screen alignments.

Links:

GitHub Link:

https://github.com/sashi987/ASE/tree/master/Increment1

ScrumDo Link:

http://www.scrumdo.com/projects/project/giveaway/iteration/121735

Project Increment 2

Objectives:

The prime objective of this iteration is to design few web pages, provide necessary validations for the web pages, write necessary java code using struts framework for the designed web pages and provide navigation between the User Registration Screen and User Login Screen.

Import Existing Services/API:

Google Maps API:

We have planned to use Google maps API for locating the location of user who wish to grab items from donor. This service is yet to be used in the mobile version as we are currently working on the desktop version of the application.

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

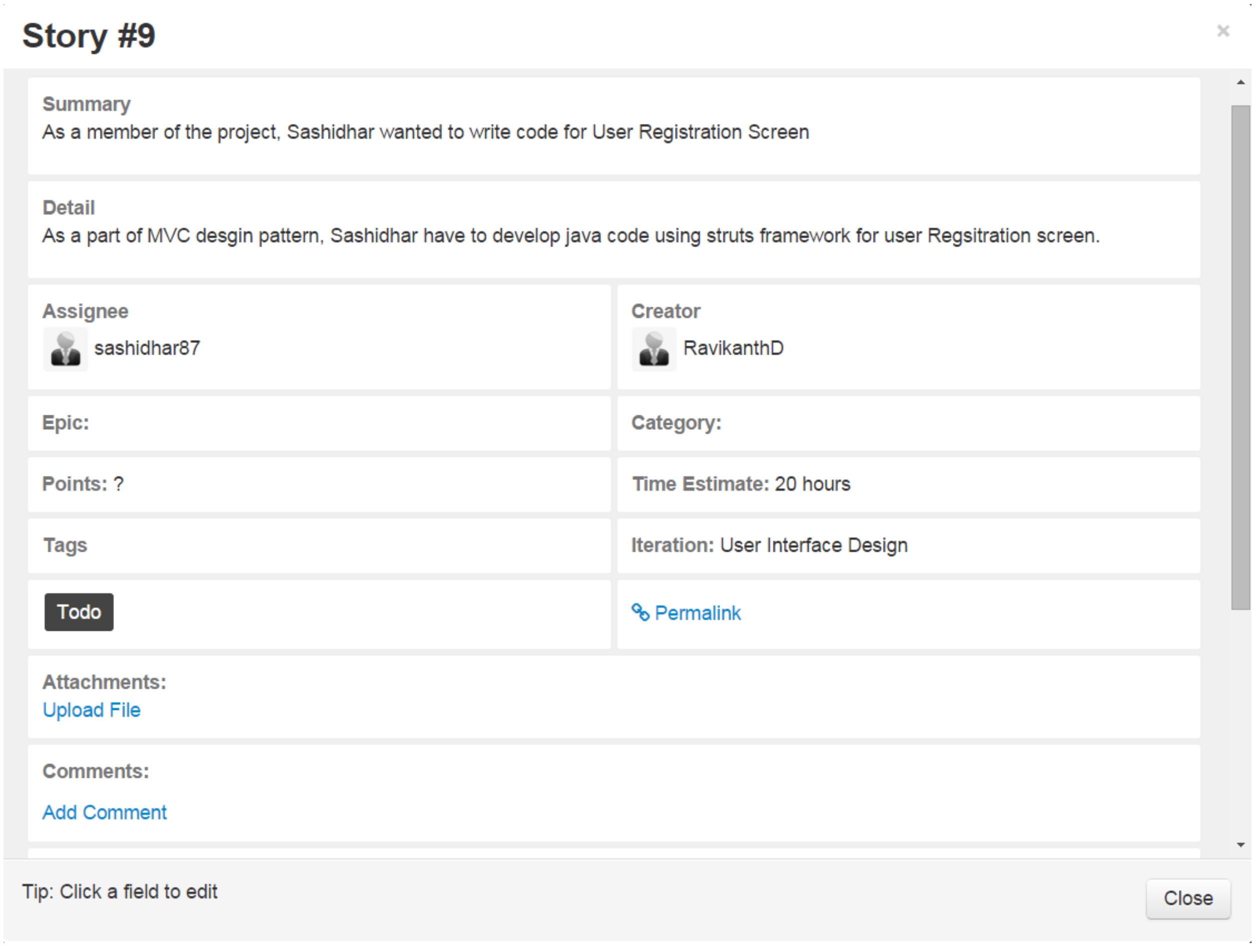
Amazon Product Search API:

We are planning to use amazon item search API for searching for items whenever a student wants to look for items and want to search for it using the application.

http://docs.aws.amazon.com/AWSECommerceService/latest/DG/ItemSearch.html

Detail Design of Services:

- **User Stories(Scrum Do Screen Shots):**
  The following are the stories we have created in the scrum do:



Story #9

Summary
As a member of the project, Sashidhar wanted to write code for User Registration Screen

Detail
As a part of MVC desgin pattern, Sashidhar have to develop java code using struts framework for user Regsitration screen.

Assignee
sashidhar87

Creator
RavikanthD

Epic:

Category:

Points: ?

Time Estimate: 20 hours

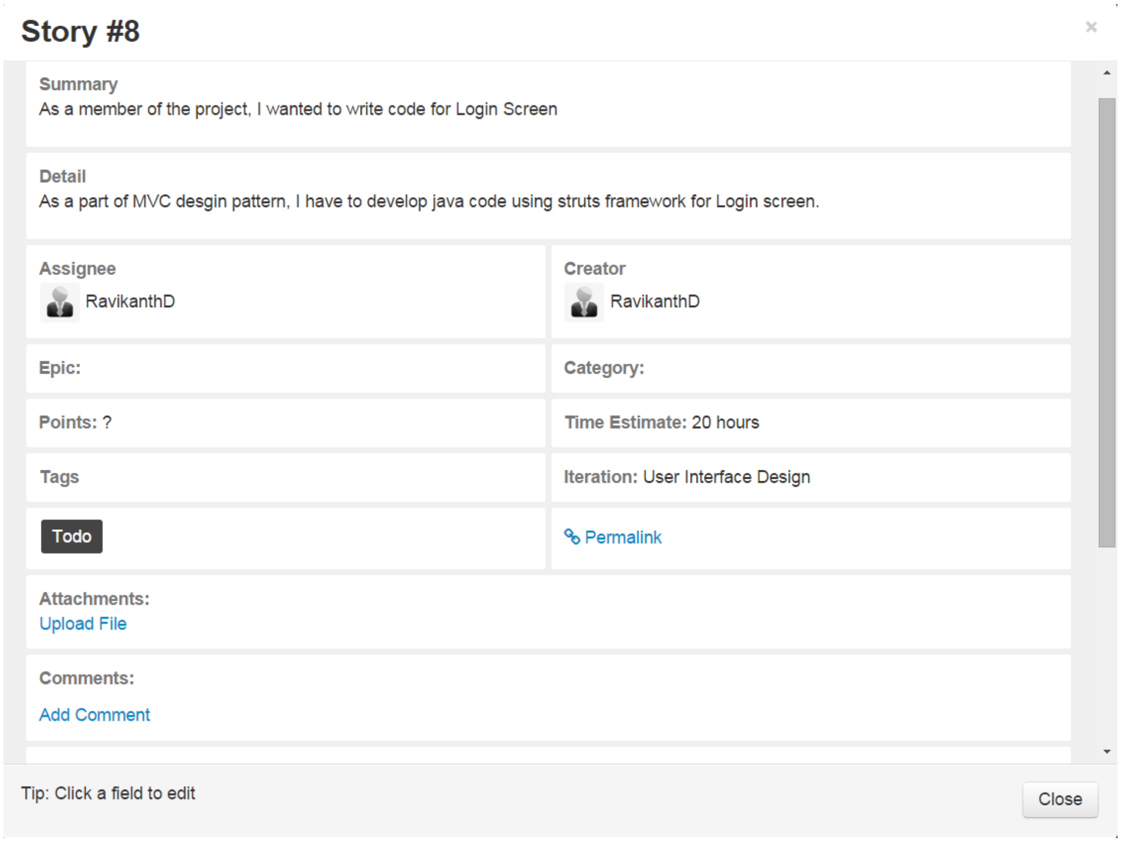
Tags

Iteration: User Interface Design

Attachments:

Comments:

Tip: Click a field to edit



Story #8

Summary
As a member of the project, I wanted to write code for Login Screen

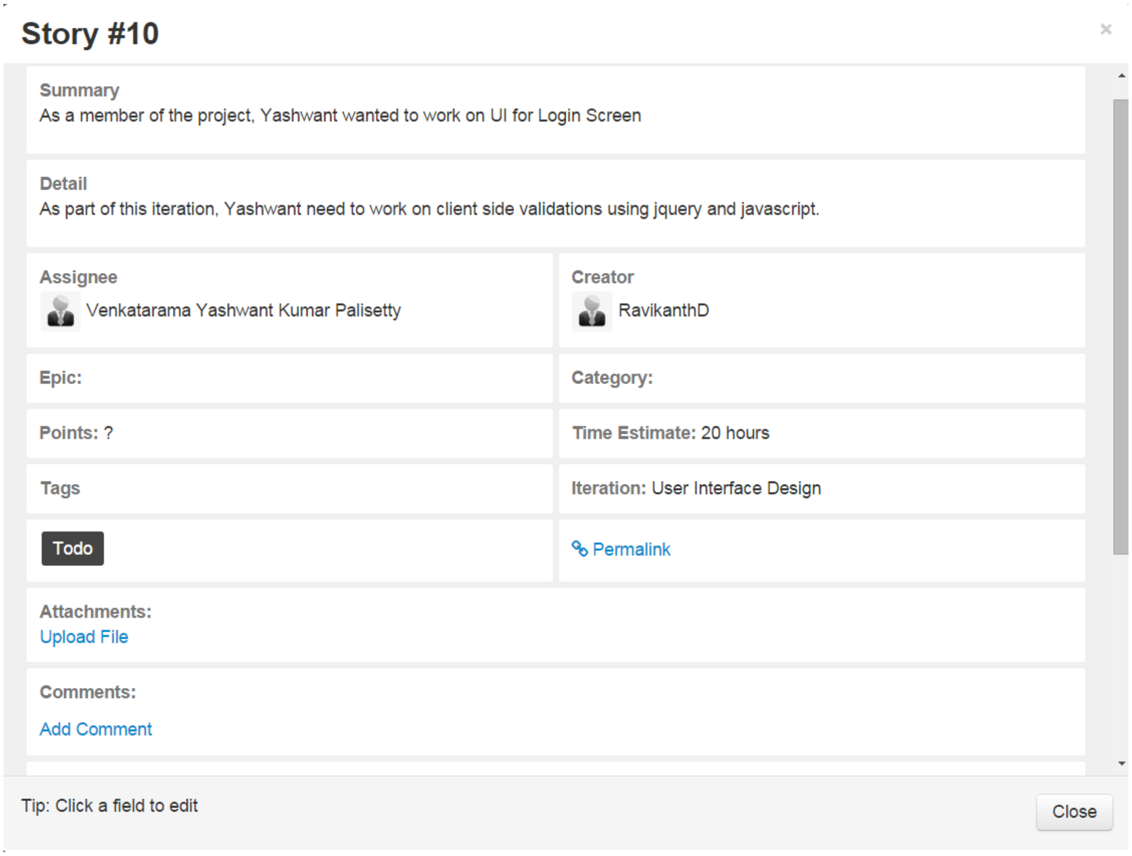
Detail
As a part of MVC desgin pattern, I have to develop java code using struts framework for Login screen.

Assignee RavikanthD	Creator RavikanthD
Epic:	Category:
Points: ?	Time Estimate: 20 hours
Tags	Iteration: User Interface Design
Todo	Permalink

Attachments:
Upload File

Comments:
Add Comment

Tip: Click a field to edit



Story #10

Summary
As a member of the project, Yashwant wanted to work on UI for Login Screen

Detail
As part of this iteration, Yashwant need to work on client side validations using jquery and javascript.

Assignee
Venkatarama Yashwant Kumar Palisetty

Creator
RavikanthD

Epic:

Category:

Points: ?

Time Estimate: 20 hours

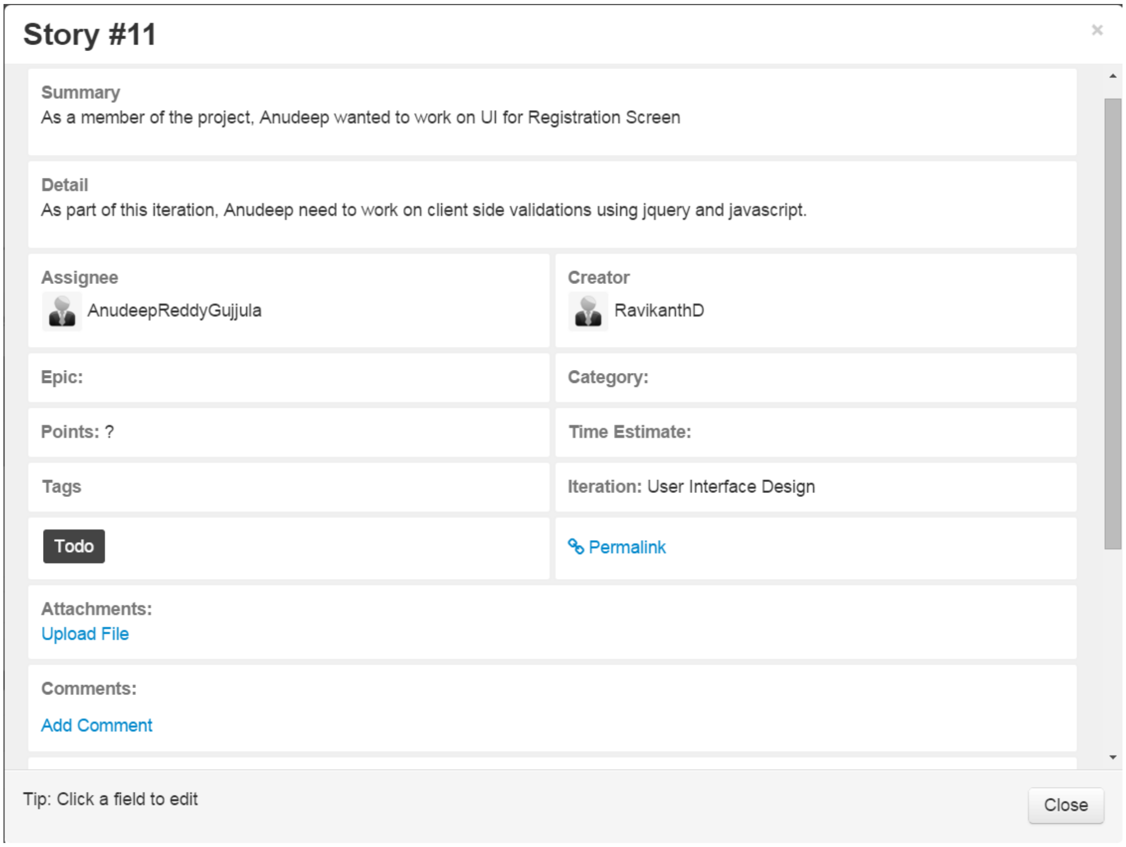
Tags

Iteration: User Interface Design

Attachments:

Comments:

Tip: Click a field to edit



Story #11

Summary
As a member of the project, Anudeep wanted to work on UI for Registration Screen

Detail
As part of this iteration, Anudeep need to work on client side validations using jquery and javascript.

Assignee
AnudeepReddyGujjula

Creator
RavikanthD

Epic:

Category:

Points: ?

Time Estimate:

Tags

Iteration: User Interface Design

Todo

Permalink

Attachments:
Upload File

Comments:
Add Comment

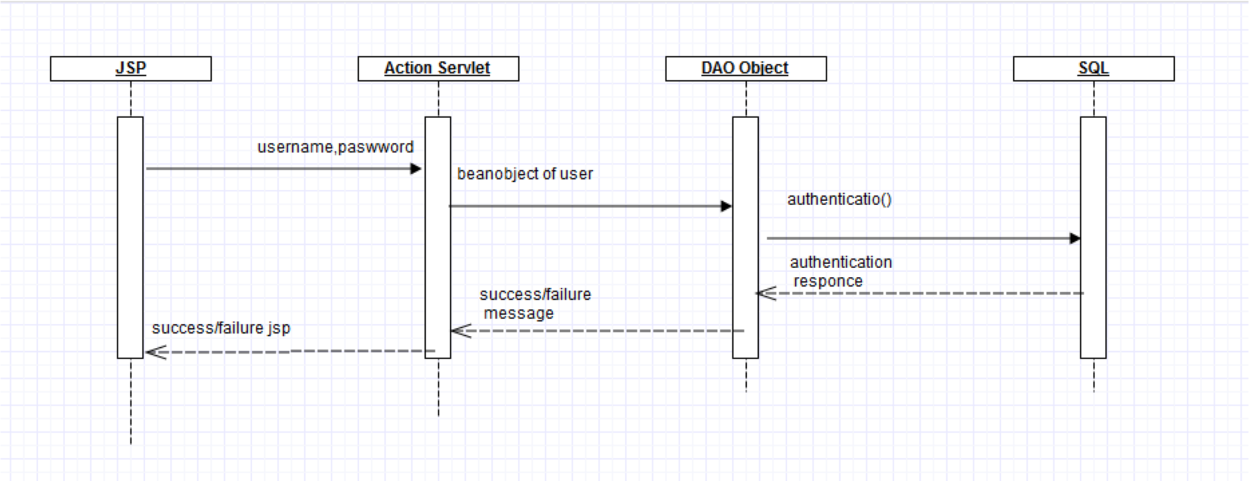
Tip: Click a field to edit

Close

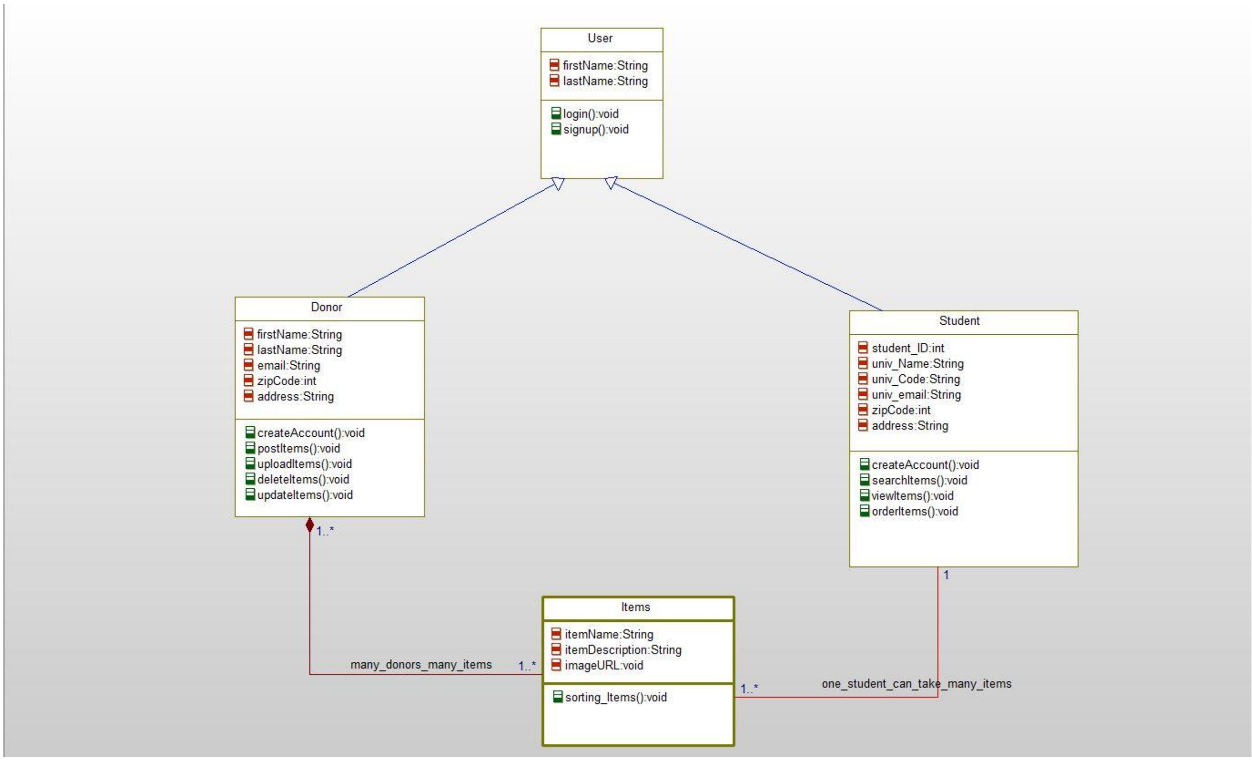
Service description:

We are have developed Registration and Login modules as part of second increment. We are using Oracle database to make the data persistent. We are planning to build a Restful service on top of it to store username and passwords and also to authenticate the users.

Sequence diagram:



Class diagram:



Design of User Interface:

- **Features:** The client will have the required UI pages to make a user login to the application look for items or even post some items for giveaway. Separate login for people who look donate items and people who look to grab items and also student verification for people who would like to grab items.
- **Styles:** As the application has many images of items which are posted in the application to donate, we need to use effective CSS styles to separate text and images and to make them fit to the screen without overlapping. Bootstrap gives an effective way CSS styling where we can maintain a single style sheet file and include all the style elements in that sheet with references.
- **Technologies:** For the UI screen design we would go with HTML5, JQuery as they provide some advanced features for UI design and scripting.

Design of Unit test cases:

Test Case Id	**Module**	**Description**	**Expected Result**	**Status**
1	Registration	Click signup button without entering user details.	Messaging showing mandatory fields required	**Pass**
2	Registration	Numeric in First name ,Last name	Message showing numeric are not accepted.	**Pass**
3	Registration	Entering different passwords in password and retype password fields.	Message showing passwords should be same	**Pass**
4	Login	Click on submit with blank username and passwords.	Message showing mandatory fields required	**Pass**
5	Login	Entering unregistered username and password.	Application should not allow to login .	**Pass**
6	Login	Entering a valid username and Password	System should allow the user to login and a welcome page should be displayed.	**Pass**

Implementation:

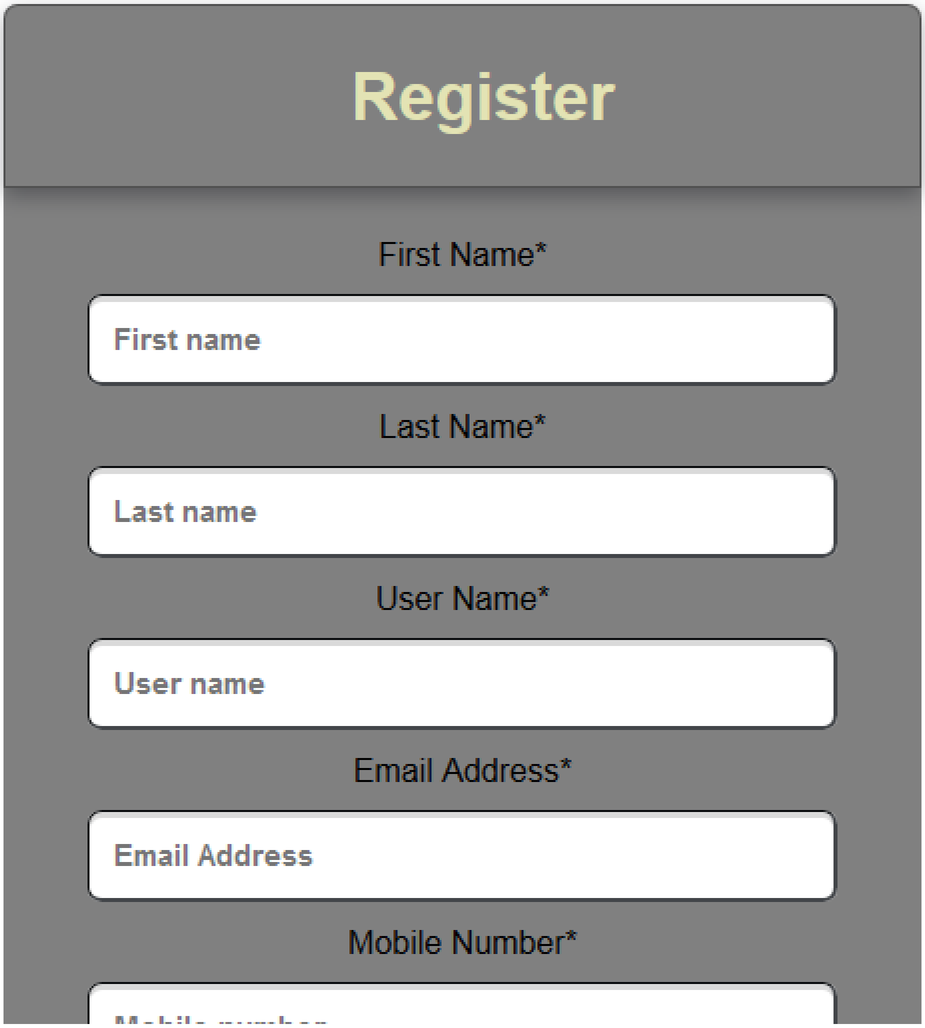
Implementation of Rest Services:

Right now we are working on Web Application and Rest services are yet to be implemented in the next increments.

Implementation of User Interface:

User Registration Screen:

Give Away Signup



Register

First Name*

First name

Last Name*

Last name

User Name*

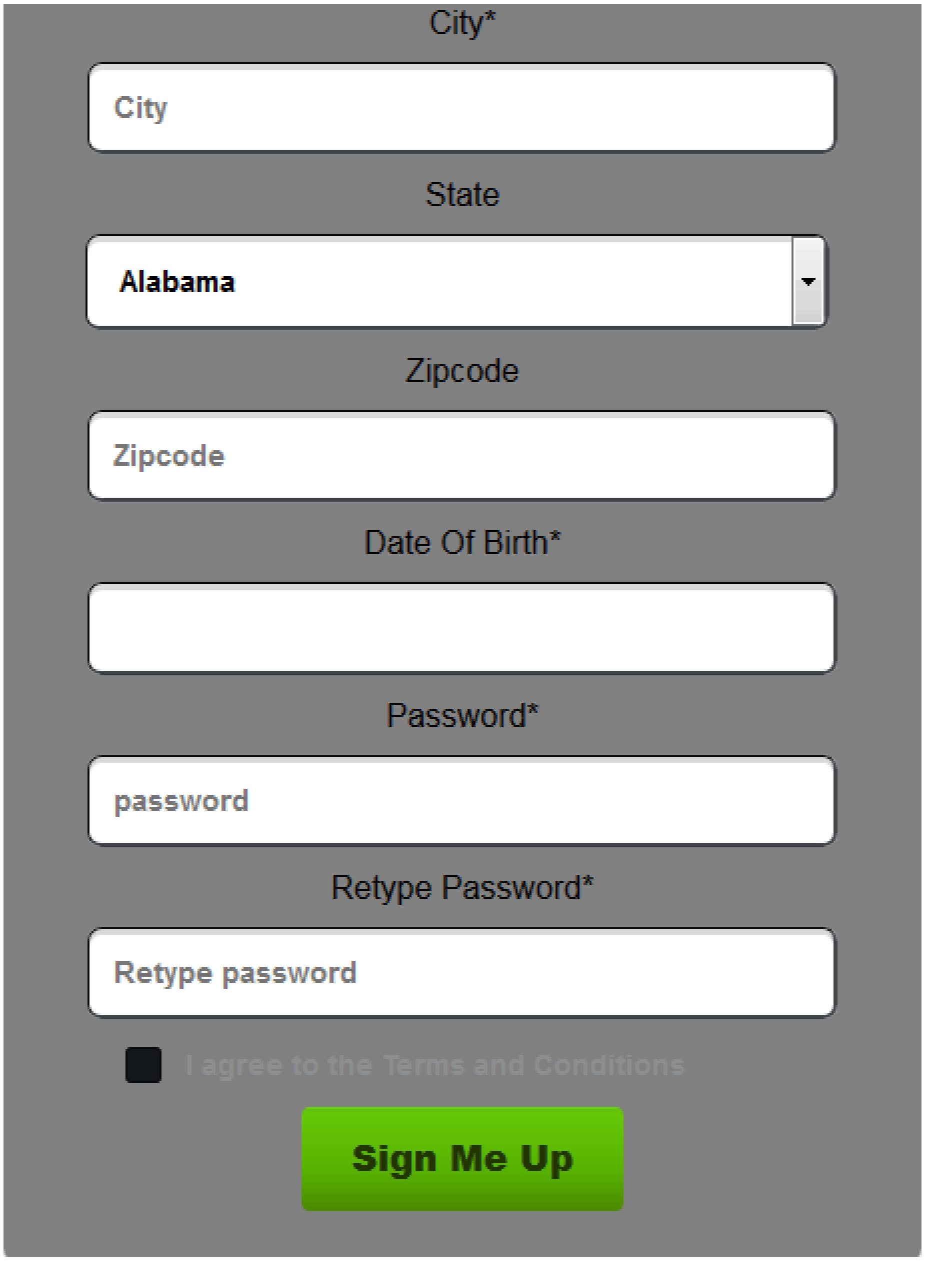
User name

Email Address*

Email Address

Mobile Number*

Mobile number



City*

City

State

Alabama

Zipcode

Zipcode

Date Of Birth*

Password*

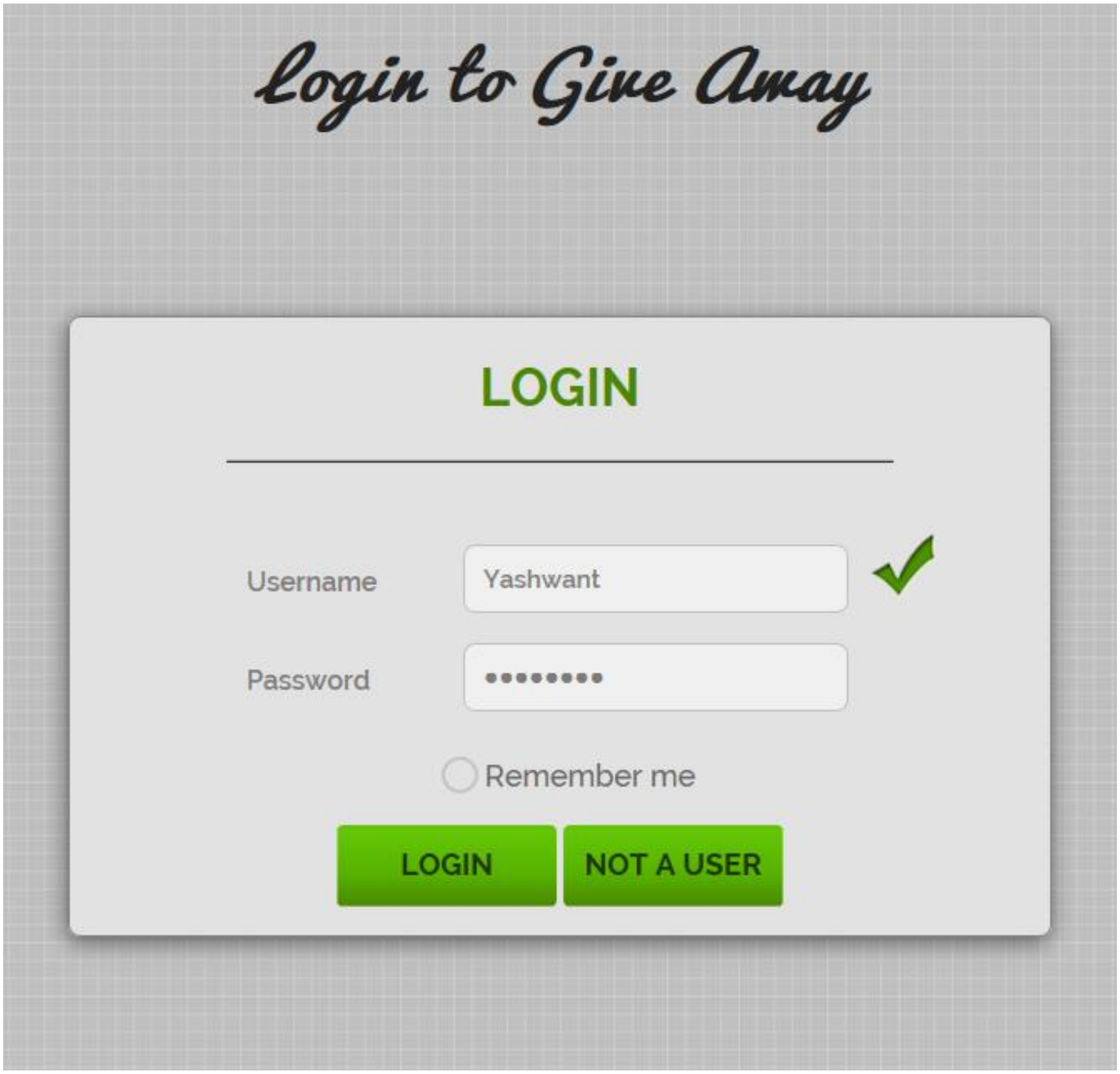
password

Retype Password*

Retype password

■ I agree to the Terms and Conditions

Login Page:



Login to Give Away

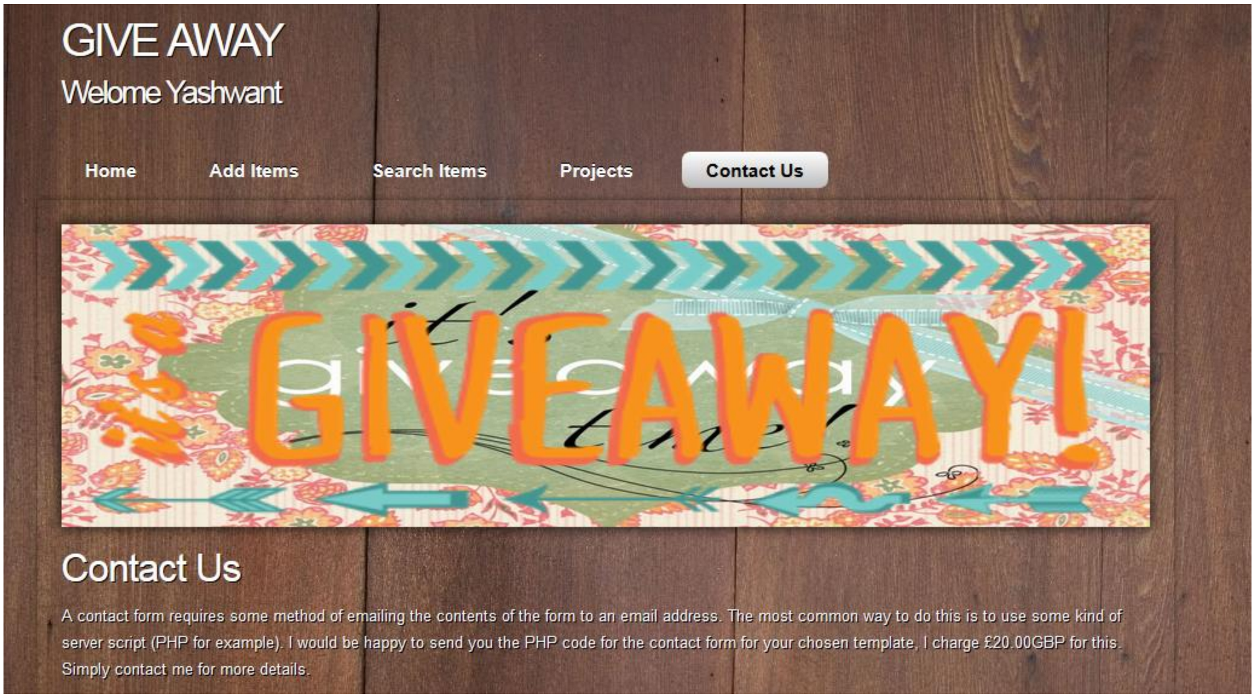
LOGIN

Username: Yashwant ✓

Password: •••••••••

○ Remember me

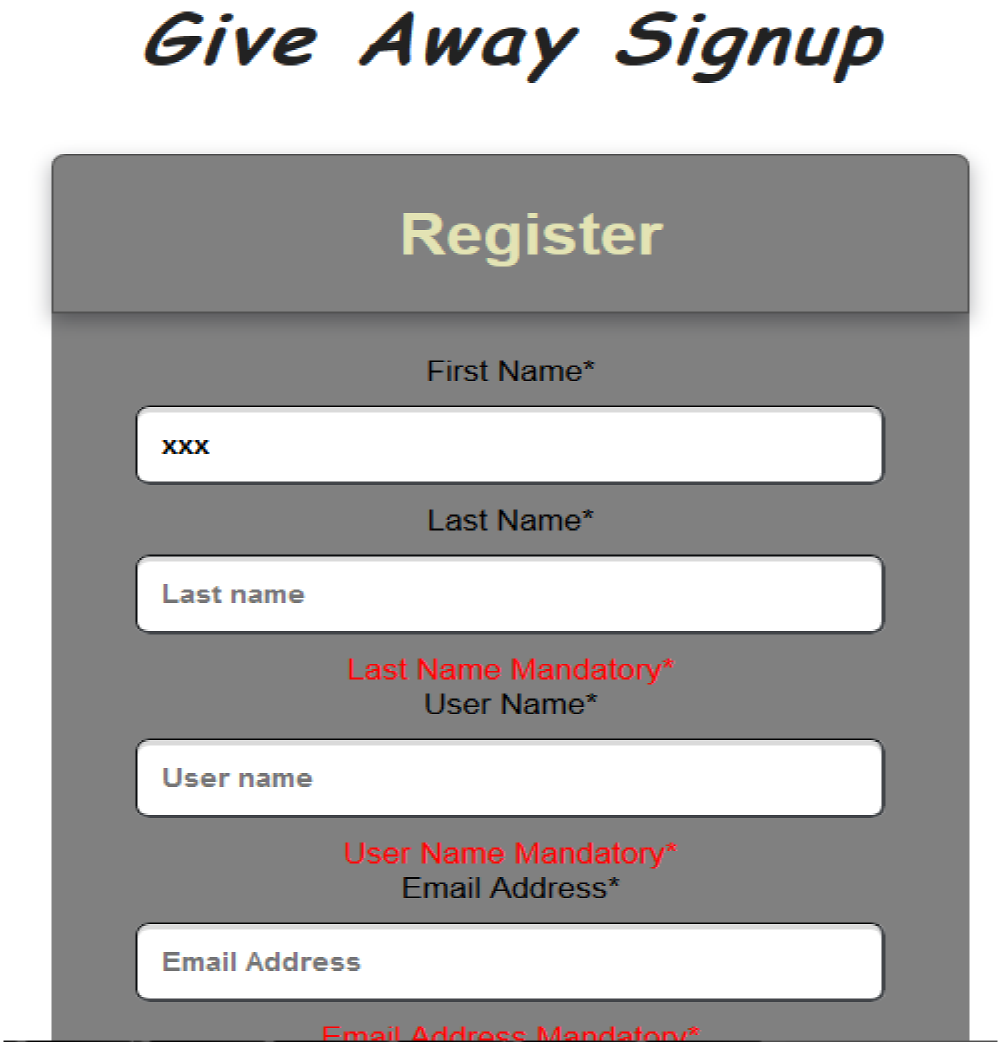
Home Page:



Contact Us

A contact form requires some method of emailing the contents of the form to an email address. The most common way to do this is to use some kind of server script (PHP for example). I would be happy to send you the PHP code for the contact form for your chosen template, I charge £20.00GBP for this. Simply contact me for more details.

Implementation of Test Cases: Screenshots of client side validation Implementation.



First Name*

xxx

Last Name*

Last name

Last Name Mandatory*

User Name*

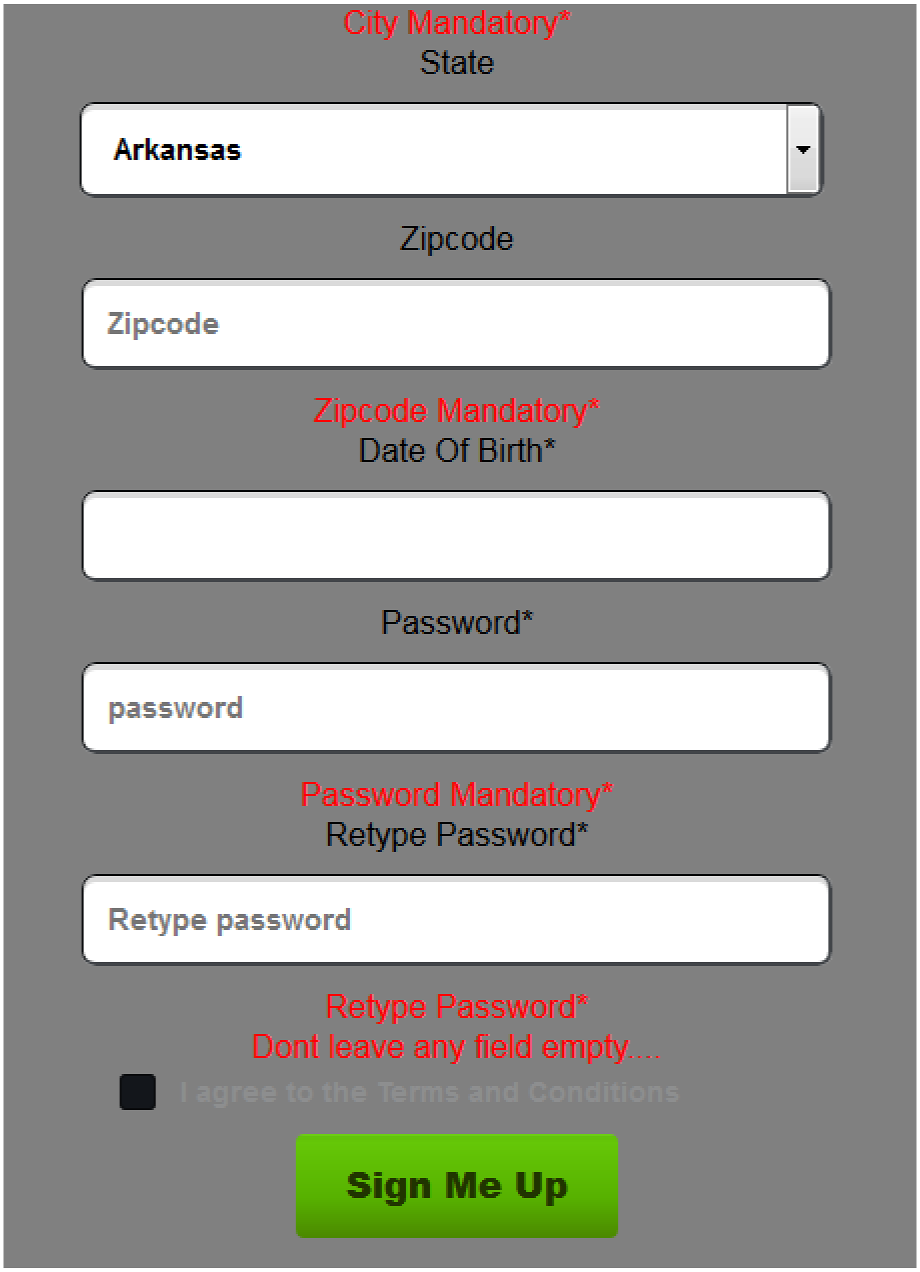
User name

User Name Mandatory*

Email Address*

Email Address

Email Address Mandatory*



City Mandatory*

State

Arkansas

Zipcode

Zipcode

Zipcode Mandatory*

Date Of Birth*

Password*

password

Password Mandatory*

Retype Password*

Retype password

Retype Password*

Dont leave any field empty....

□ I agree to the Terms and Conditions

Deployment:

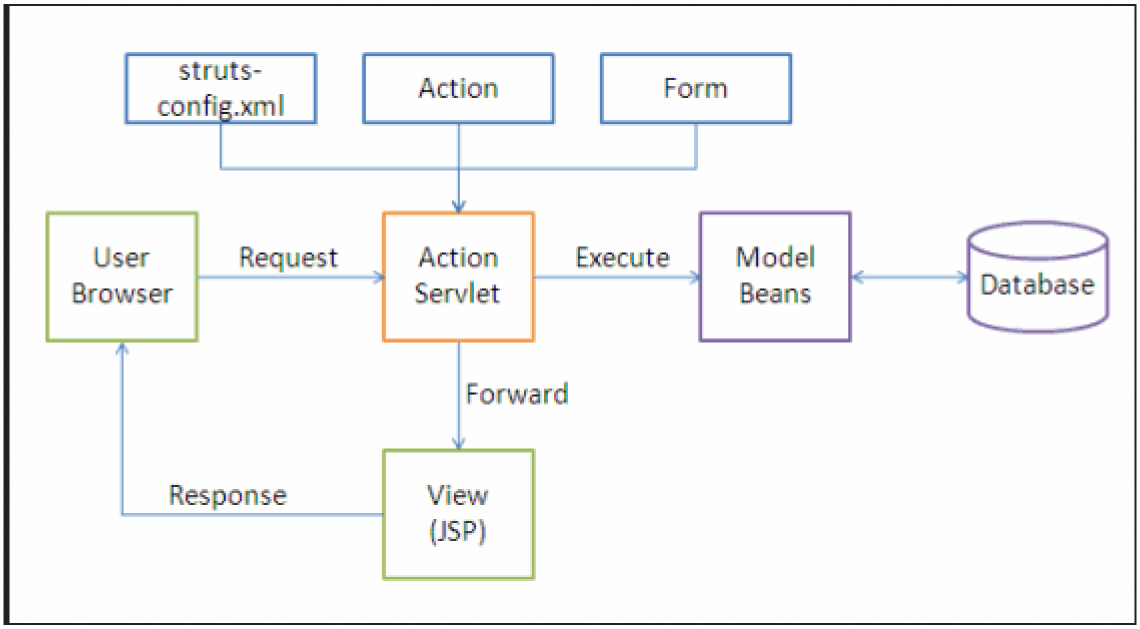
Project Scrum Do Link: **http://www.scrumdo.com/organization/giveaway/dashboard**

GitHub Link: https://github.com/sashi987/ASE/tree/master/Increment2

Report:

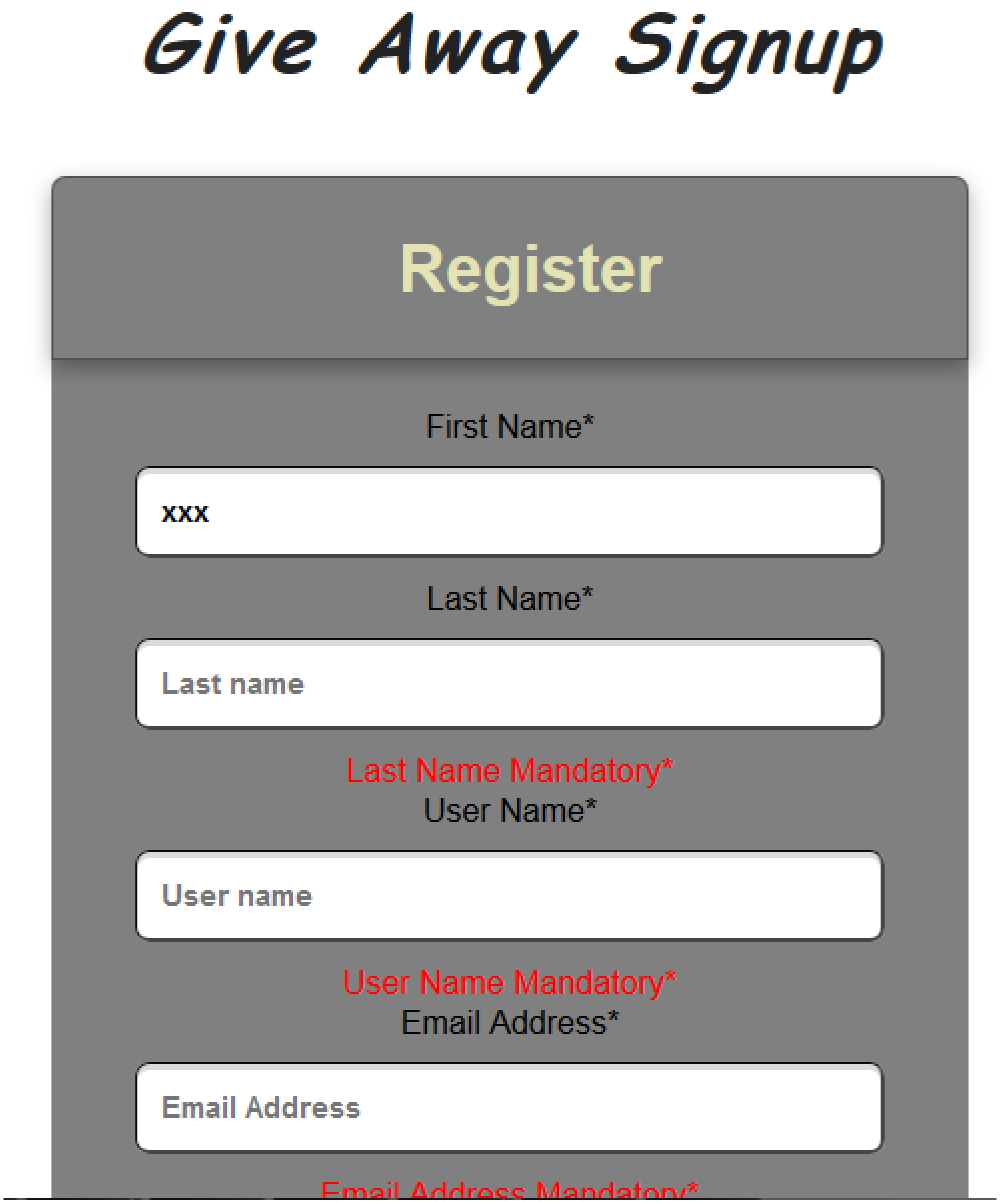
The Give Away application is being developed as web application using Struts2 framework on Java and Oracle data base for data storage and Jsp and javascipting and JQuery for User interface.

Struts Architecture:



The following are the screens that we are using as part of our project,

Registration Screen: Here in the registration page a user have to fill in the details like First name , Last name, Username . Email address, Mobile number. Date of birth, address, state. Zip code and a password for creating an account. All the fields are mandatory fields for user registration. Failure to fill any details will not allow the user to login.



Give Away Signup

Register

First Name*

xxx

Last Name*

Last name

Last Name Mandatory*

User Name*

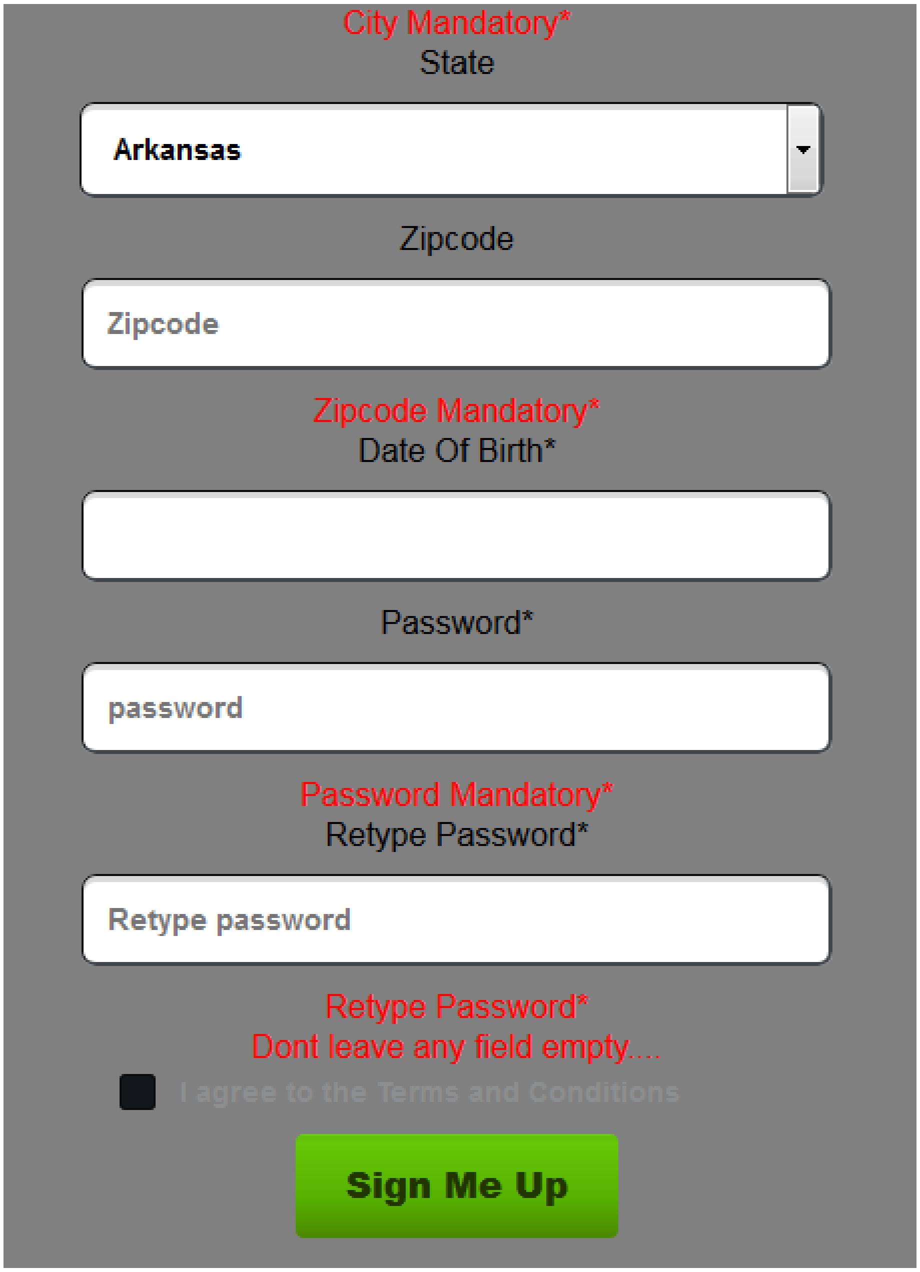
User name

User Name Mandatory*

Email Address*

Email Address

Email Address Mandatory*



City Mandatory*

State

Arkansas

Zipcode

Zipcode

Zipcode Mandatory*

Date Of Birth*

Password*

password

Password Mandatory*

Retype Password*

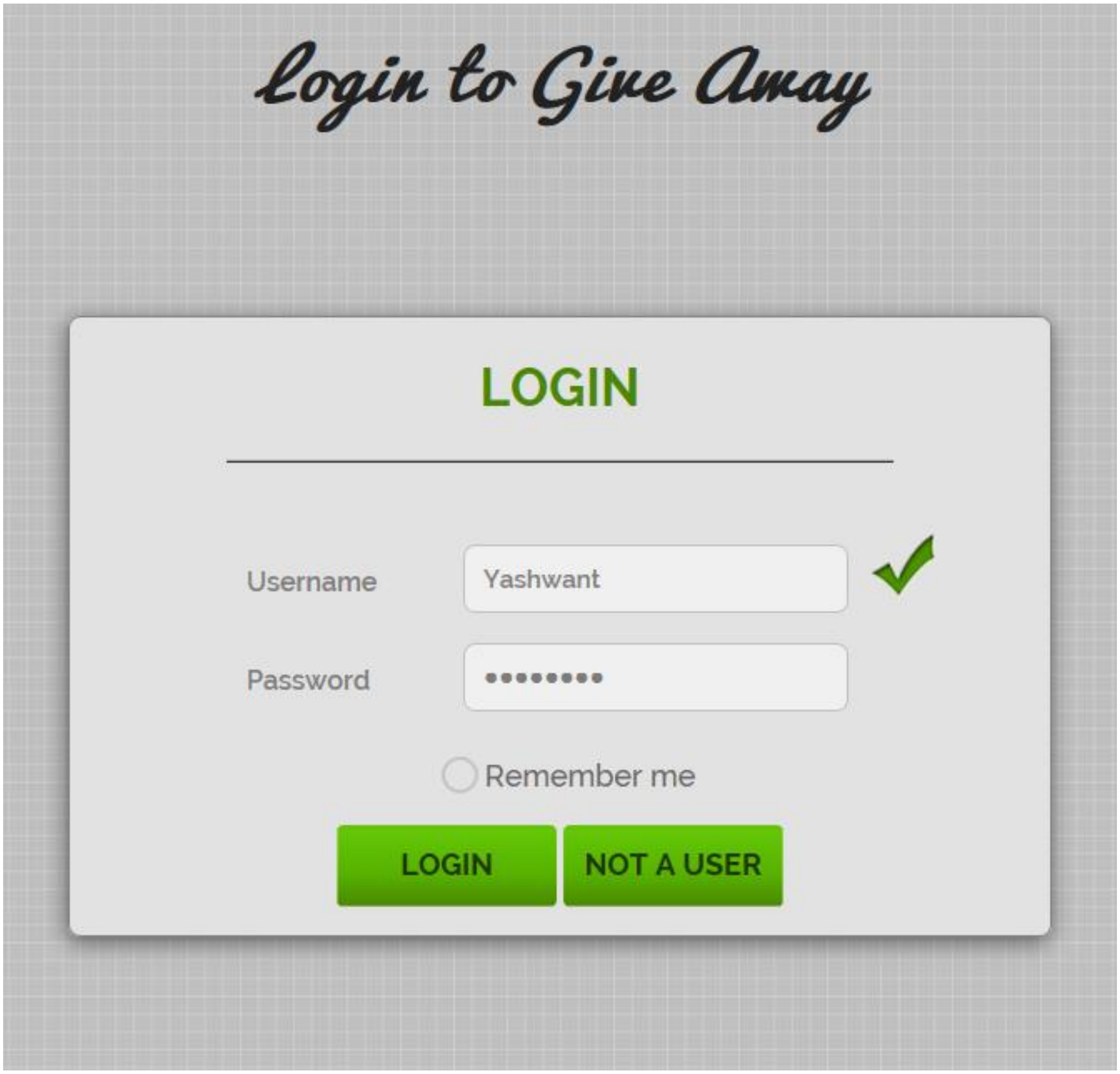
Retype password

Retype Password*

Dont leave any field empty....

□ I agree to the Terms and Conditions

Login Screen: This is the Login page where the user can login to the application if he is registered user of the application. Here there will be two fields, username and password.

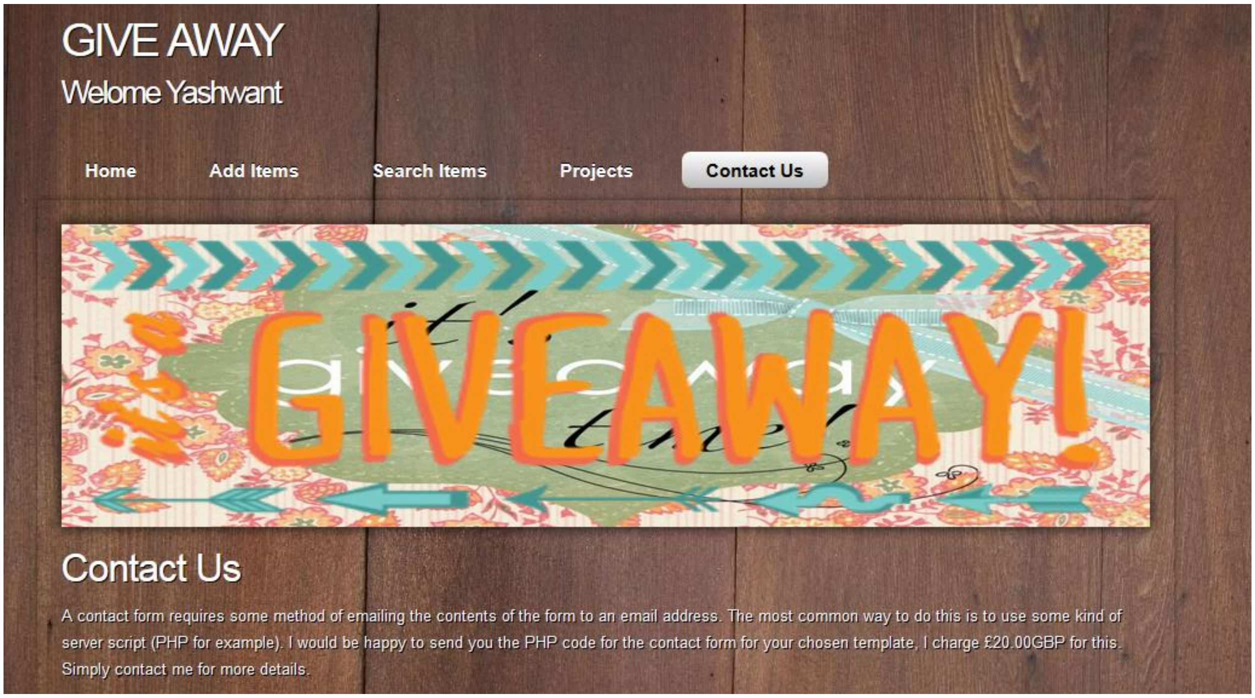


Username: Yashwant

Password: •••••••••

○ Remember me

Home Page: This is the Landing Page for the application from where the user can navigate to various other screens depending on his requirement. The various tabs include Add Items, Search Items, and Contact Us.



GIVE AWAY

Welome Yashwant

Contact Us

A contact form requires some method of emailing the contents of the form to an email address. The most common way to do this is to use some kind of server script (PHP for example). I would be happy to send you the PHP code for the contact form for your chosen template, I charge £20.00GBP for this. Simply contact me for more details.

Project Management:

https://www.scrumdo.com/projects/project/giveaway/iteration/121738/board

Implementation Status Report:

Work completed: Registration and Login Module for Users.

- Description: In the UI part screens like user login page, user registration page, home page,page have designed and with certain CSS styles and required JavaScript functions for client side validation. Connecting the pages with struts framework and successfully registering the users and assigning them username and passwords for login.
- Responsibility (Task, Person):

  Sashidhar Reddy Gowra-Developed java code on struts framework to connect to Oracle database and store registration details of the users, registering with the application.

  Yashwanth Palisetty: UI Screen Design and client side javascript validations for registration and home Pages.

  Ravi kanth Devanaboyina: Developed java code for login module by connecting to Oracle database with the DAO layer.

  Anudeep Reddy Gujjula: UI Screen Design and client side validations for Login page.
- Time taken (#hours): UI Design and Client Validations: 20 hrs Java code on struts: 24 hrs.
- Contributions (members/percentage): Sashidhar Reddy Gowra 25%, Yashwanth Palisetty 25%, Ravikanth Devaboyina 25%, Anudeep Reddy Gujjula 25%.

Work to be completed:

- Description: Creating web services to get the data from Database, Using the already existing Google maps API for sharing the address location of a user, Unit Testing, System Testing, Testing the application on Android Devices
- Responsibility (Task, Person):
  Creating Web services, Hash Functions: Sashidhar Reddy Gowra, Ravikanth Devaboina.
  Populating screens, UI alignments: Yashwanth Palisetty, Anudeep Reddy Gujjula.
- Time to be taken (estimated #hours) 150 hrs.
- Issues/Concerns: Creating RESTfull Web services, Creating hash functions for password hiding, Android devices screen alignments.

Project Increment 3

Objective:

The prime objective of this iteration is to create the web services to populate the data UI screens for the modules like Registration Module, Login Module, Item Addition Module, Item Updating Module, Item Deletion Module, Item Selection Module and Subscription Modules.

Import Existing API or Services:

Google Maps API:

We have planned to use Google maps API for locating the location of user who wish to grab items from donor. This service is yet to be used in the mobile version as we are currently working on the desktop version of the application.

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

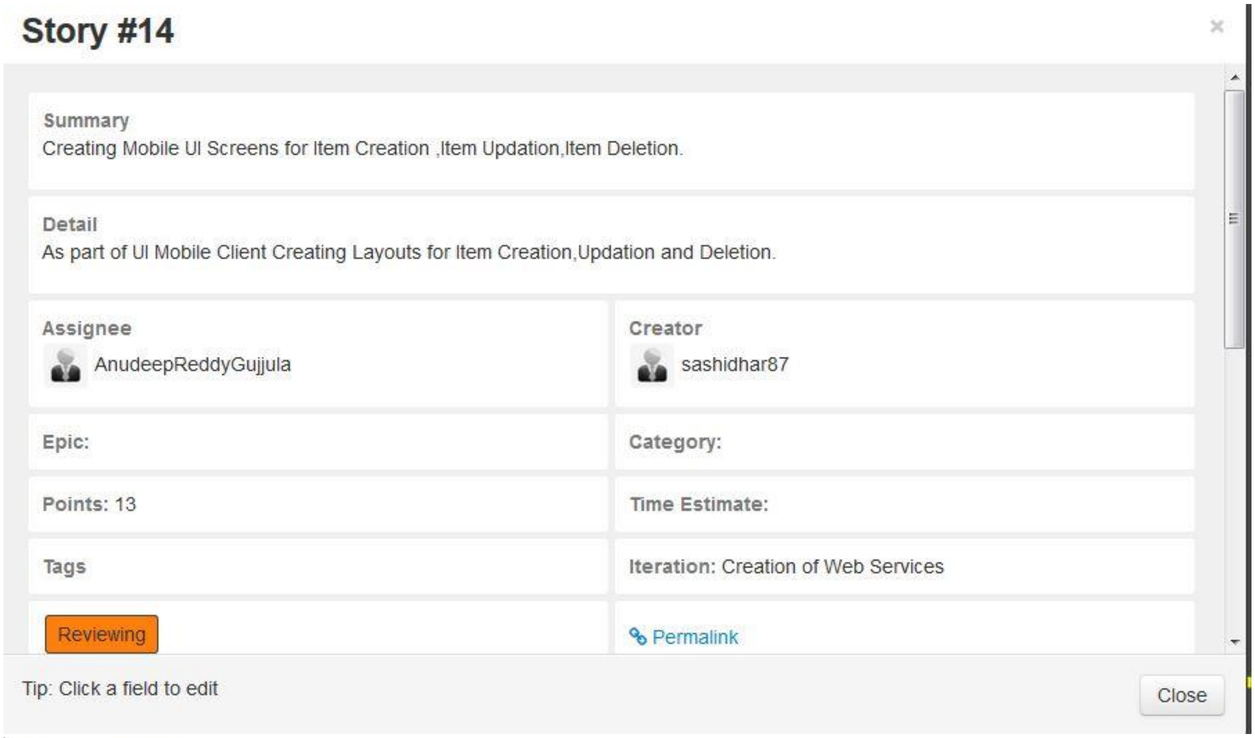
Amazon Product Search API:

We are planning to use amazon item search API for searching for items whenever a student wants to look for items and want to search for it using the application.

http://docs.aws.amazon.com/AWSECommerceService/latest/DG/ItemSearch.html

Detail Design of Services:

- **User stories using ScrumDo:**
  The following are the user stories we have created in the ScrumDo:



Story #14

Summary
Creating Mobile UI Screens for Item Creation ,Item Updation,Item Deletion.

Detail
As part of UI Mobile Client Creating Layouts for Item Creation,Updation and Deletion.

Assignee
AnudeepReddyGujjula

Creator
sashidhar87

Epic:

Category:

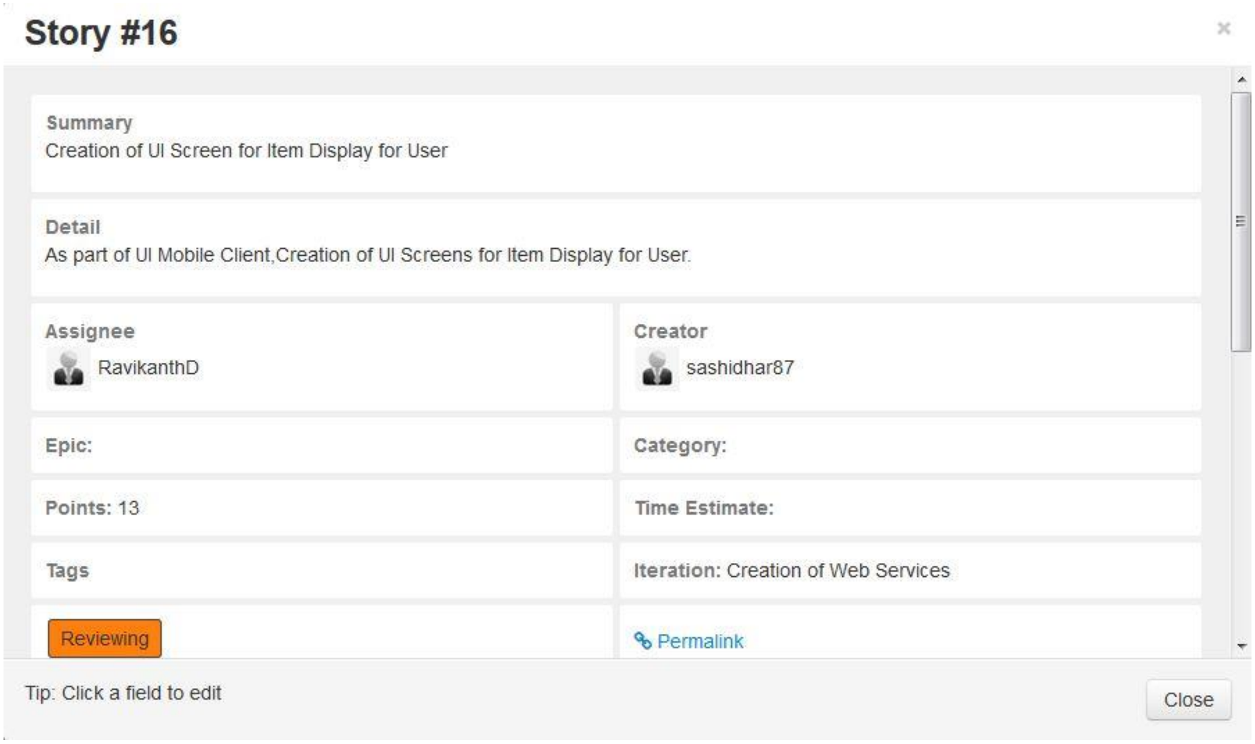
Points: 13

Time Estimate:

Tags

Iteration: Creation of Web Services

Tip: Click a field to edit



Story #16

Summary
Creation of UI Screen for Item Display for User

Detail
As part of UI Mobile Client,Creation of UI Screens for Item Display for User.

Assignee
RavikanthD

Creator
sashidhar87

Epic:

Category:

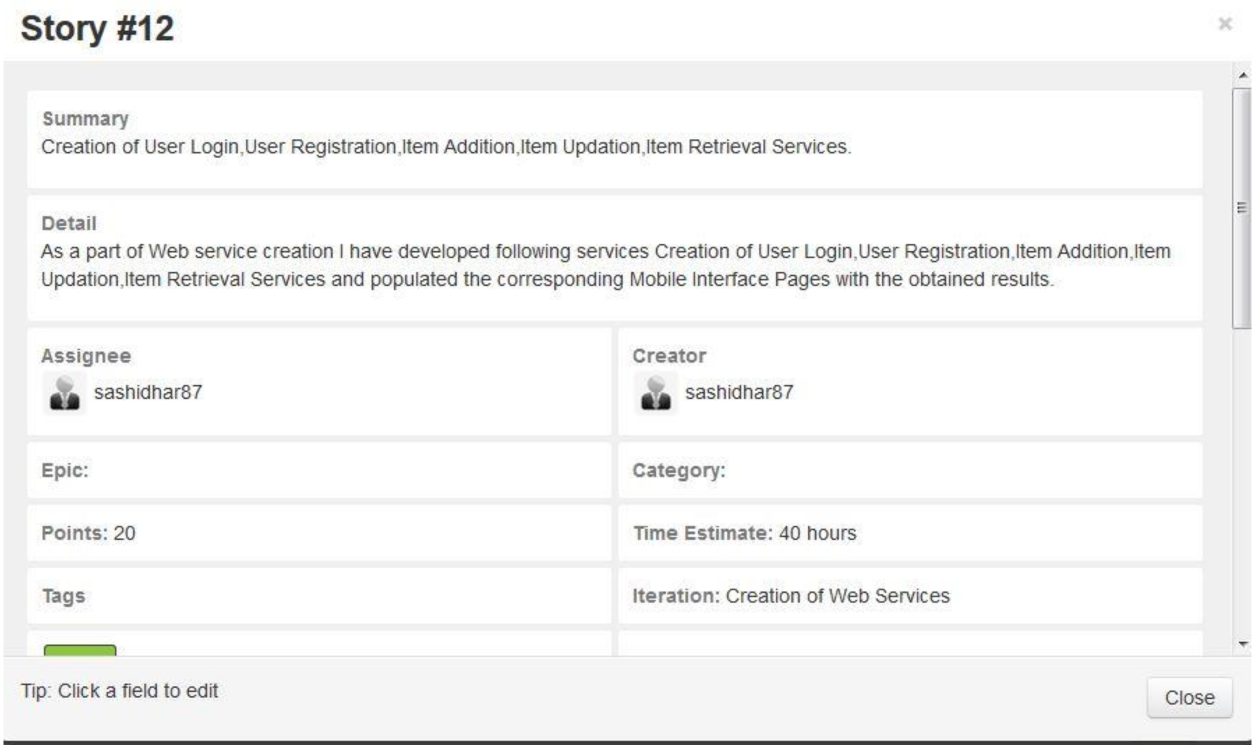
Points: 13

Time Estimate:

Tags

Iteration: Creation of Web Services

Tip: Click a field to edit



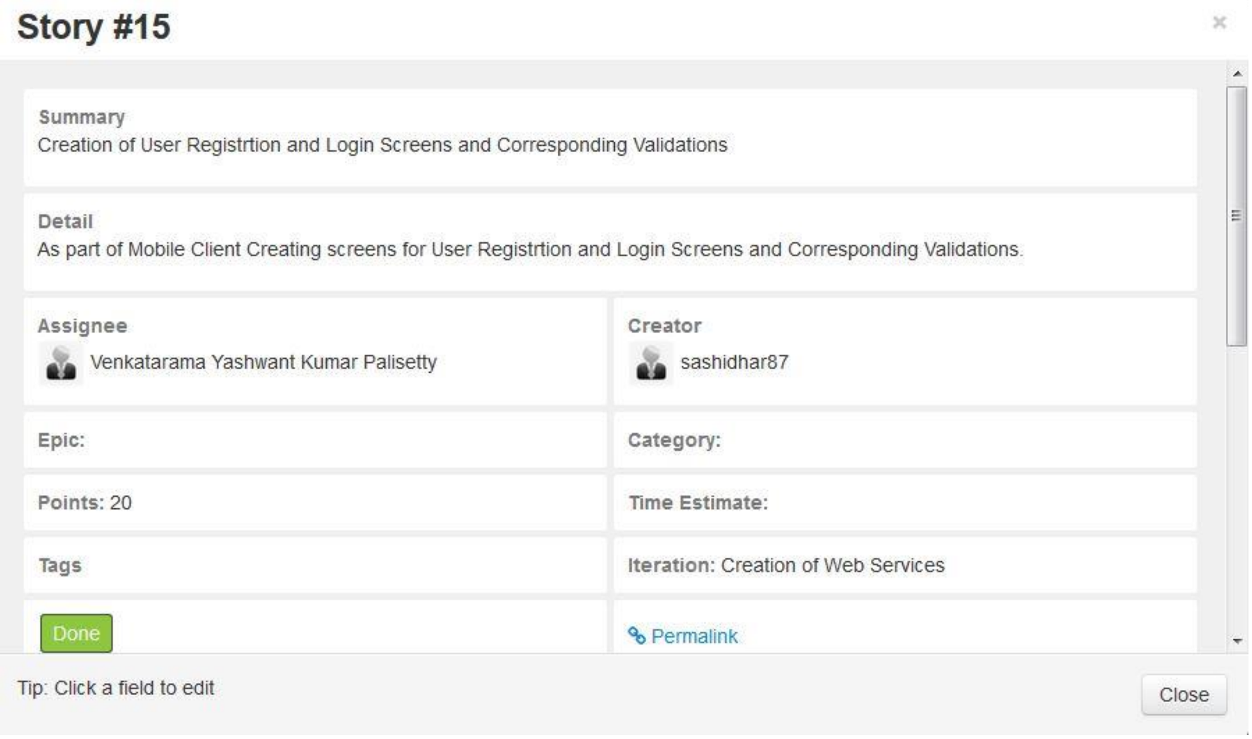
Story #12

Summary
Creation of User Login,User Registration,Item Addition,Item Updation,Item Retrieval Services.

Detail
As a part of Web service creation I have developed following services Creation of User Login,User Registration,Item Addition,Item Updation,Item Retrieval Services and populated the corresponding Mobile Interface Pages with the obtained results.

Assignee sashidhar87	**Creator** sashidhar87
Epic:	**Category:**
Points: 20	**Time Estimate:** 40 hours
Tags	**Iteration:** Creation of Web Services

Tip: Click a field to edit



Story #15

Summary
Creation of User Registrtion and Login Screens and Corresponding Validations

Detail
As part of Mobile Client Creating screens for User Registrtion and Login Screens and Corresponding Validations.

Assignee Venkatarama Yashwant Kumar Palisetty	**Creator** sashidhar87
Epic:	**Category:**
Points: 20	**Time Estimate:**
Tags	**Iteration:** Creation of Web Services
Done	Permalink

Tip: Click a field to edit

- **Service Description:**

I. **User Login Service:**
In this service, we are implementing User Login as a service, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

II. **User Registration Service:**
We are implementing the User Registration Service with the fields like First Name, Last Name, User Name, Password, Email ID, Mobile Number, Date of Birth, Address and Zip Code, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

III. **Item Addition Service:**
This service is used to add an item into the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

IV. **Item Updating Service:**

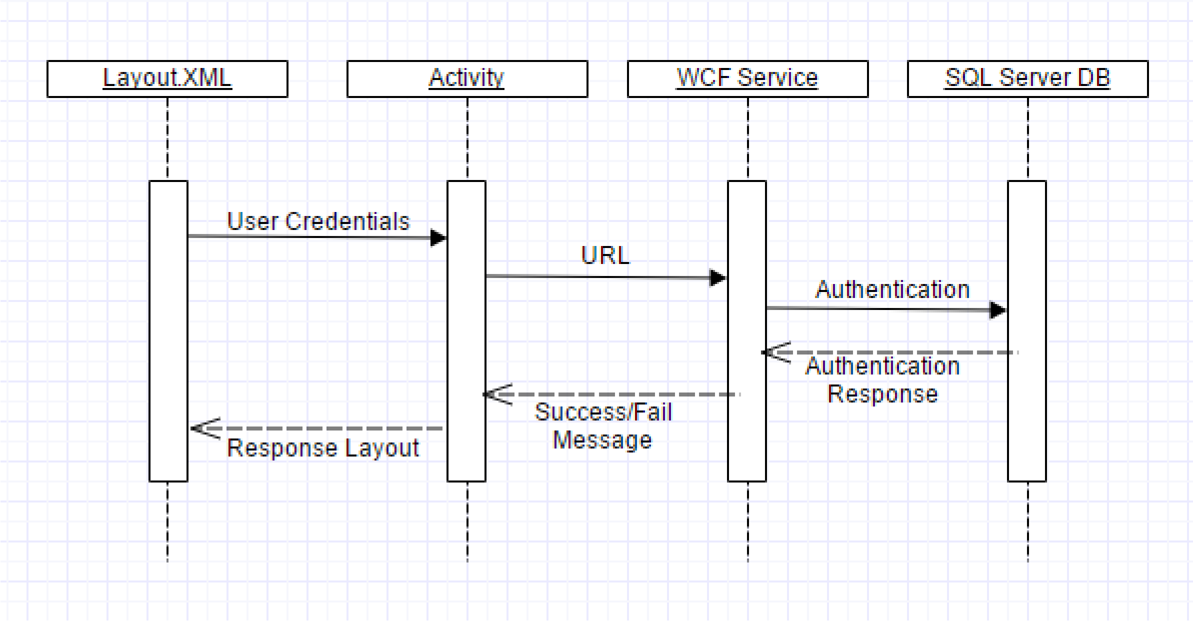
This service is used to update the details of an item in the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

V. **Item Retrieval Service:**

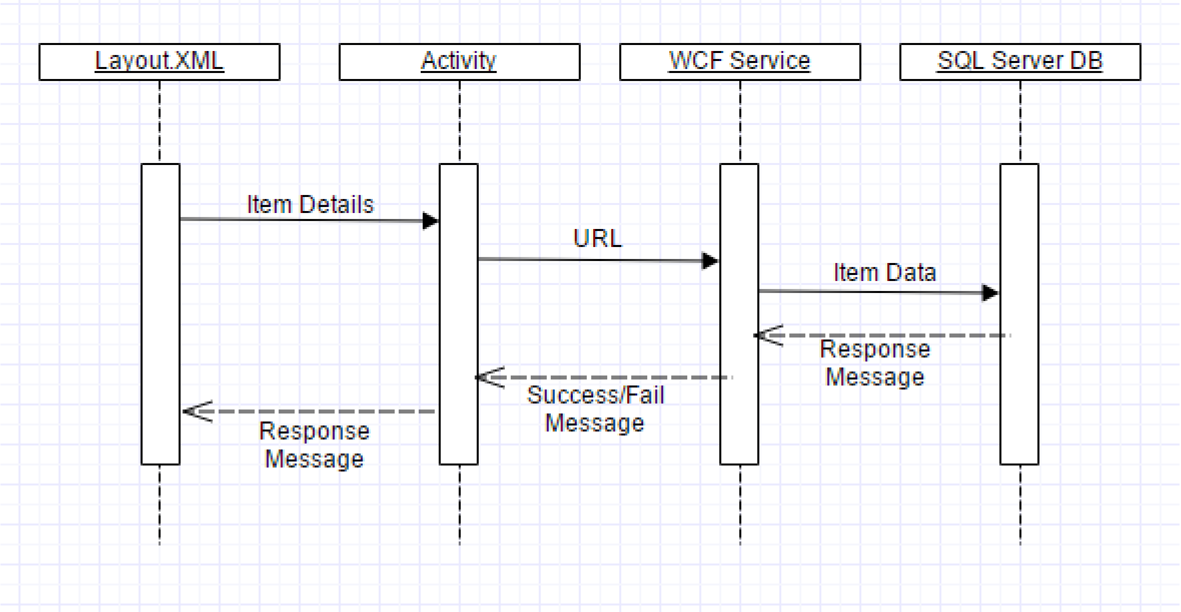
This service is used to retrieve the list of items from the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, and Method Name and get the required details.

- **Sequence Diagram:**

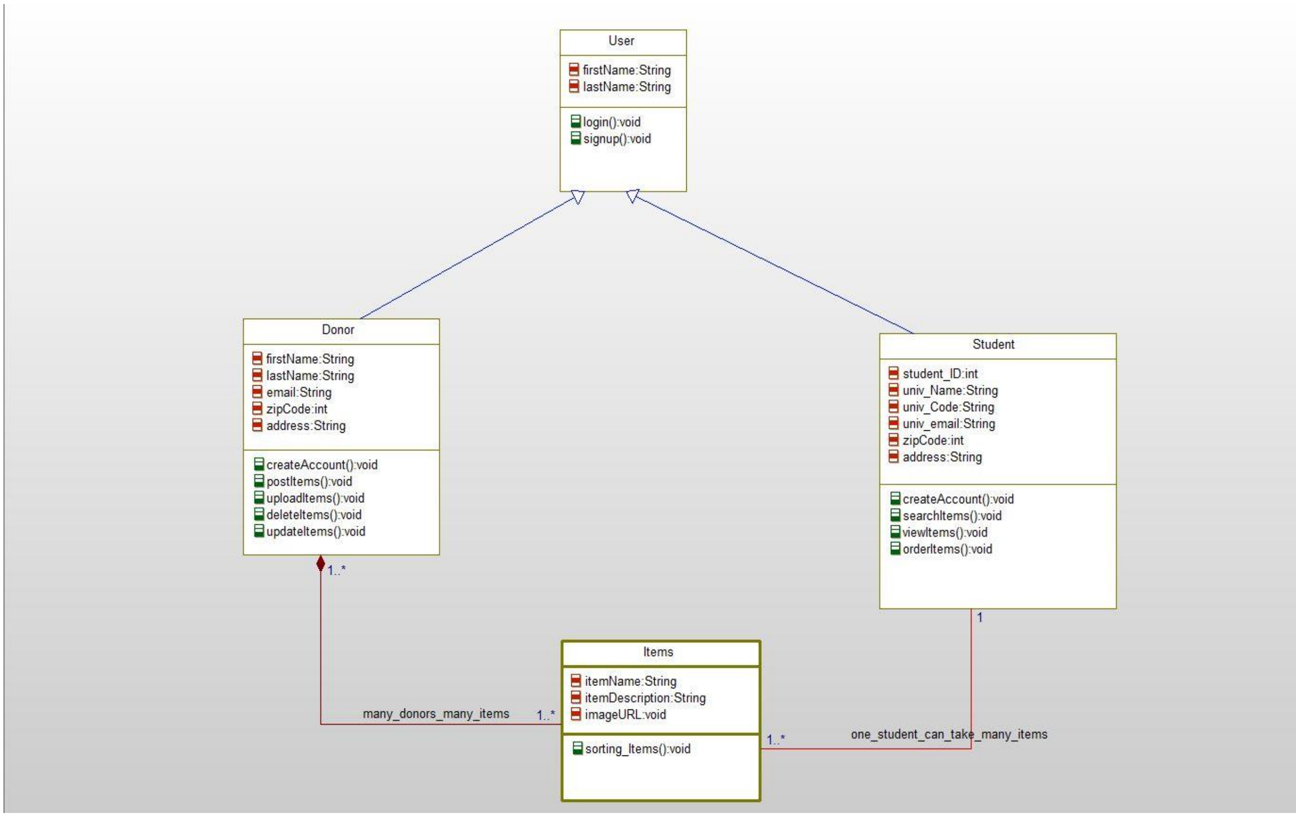
I. Sequence Diagram for User Login/Registration:



II. Sequence Diagram for adding/updating/retrieval of an Item:



- **Class Diagram:**



- **Design of Mobile Client Interface:**

a) Hardware Requirements:
   I. 1GHz processor
   II. SD card – 512 MB
   III. RAM – 512 MB
   IV. LED screen with touch enabled.

b) Software Requirements:
   I. Operating system: Android
   II. Version: Gingerbread (2.3) or advanced.

- **Design of Unit Test Cases:**

Test Case Id	**Module**	**Description**	**Expected Result**	**Status**
1	Registration	Click signup button without entering user details.	Message showing mandatory fields required	**Pass**
2	Registration	Numeric in First name ,Last name	Message showing numeric are not accepted.	**Pass**
3	Registration	Entering different passwords in password and retype password fields.	Message showing passwords should be same	**Pass**
4	Login	Click on submit with blank username and passwords.	Message showing mandatory fields required	**Pass**
5	Login	Entering unregistered username and password.	Application should not allow to login.	**Pass**
6	Login	Entering a valid username and Password	System should allow the user to login and a welcome page should be displayed.	**Pass**
7	Adding Items	Click Add Item button without entering the item details.	Message showing mandatory fields required	**Pass**
8	Adding Items	Numeric in item name	Message showing numeric are not accepted.	**Pass**
9	Adding Items	Alphabets and special characters in Quantity and Years used fields.	Message showing alphabets and special characters are not accepted.	**Pass**
10	Adding Items	Enter valid details.	System should allow the user to add the item.	**Pass**

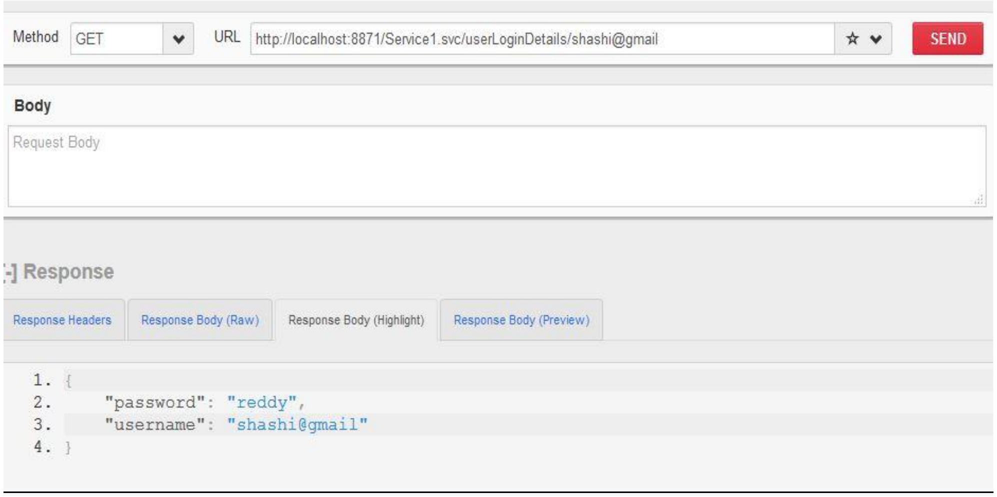
Implementation:

- **Implementation of Rest Services:**

I. User Login Service:

In this service, we are implementing User Login as a service, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

URL: http://10.0.2.2:8871/Service1.svc/userLoginDetails/shashi@gmail



Method GET

URL http://localhost:8871/Service1.svc/userLoginDetails/shashi@gmail

Body

Request Body

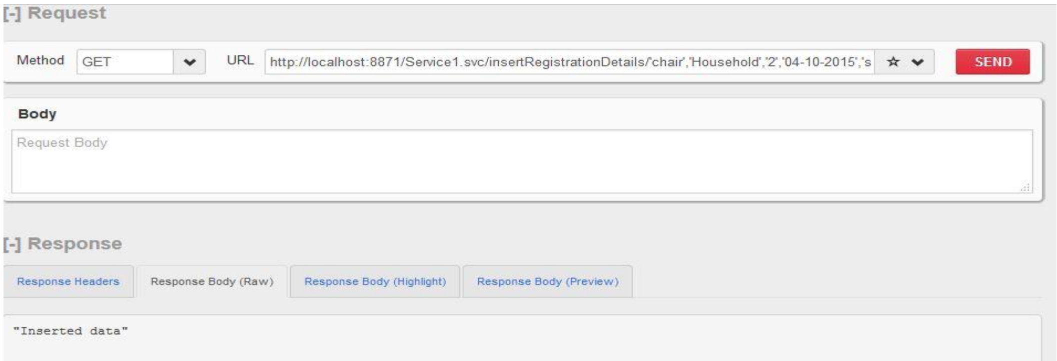
[-] Response

```
{
    "password": "reddy",
    "username": "shashi@gmail"
}
```

II. User Registration Service:

We are implementing the User Registration Service with the fields like First Name, Last Name, User Name, Password, Email ID, Mobile Number, Date of Birth, Address and Zip Code, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

URL:
http://10.0.2.2:8871/Service1.svc/insertRegistrationDetails/'shashi','reddy','sashi','reddy','com','123','08-20-1990','kansas','64112'



[-] Request

Method GET

URL http://localhost:8871/Service1.svc/insertRegistrationDetails/'chair','Household','2','04-10-2015','s

Body

Request Body

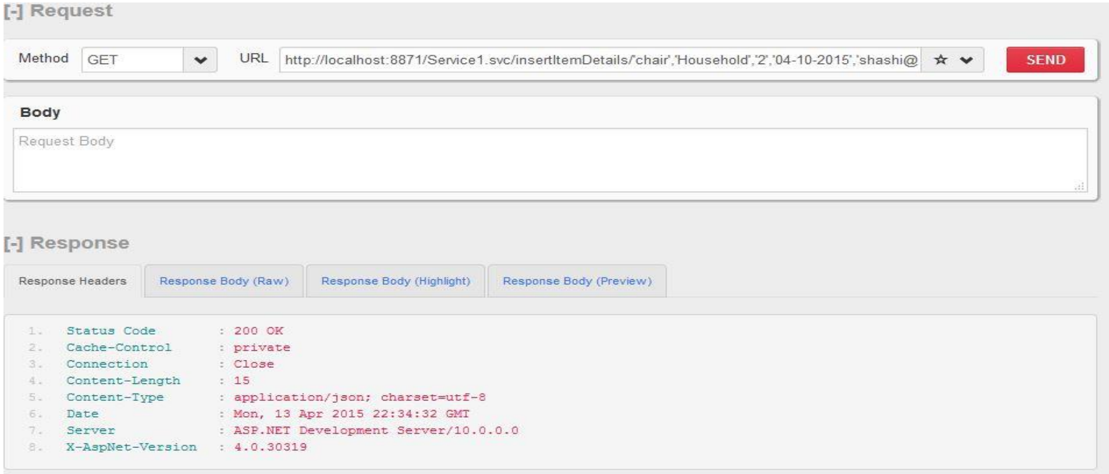
[-] Response

```
"Inserted data"
```

III. Item Addition Service:

This service is used to add an item into the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

URL: http://10.0.2.2:8871/Service1.svc/insertItemDetails/'chair','Household','2','04-10-2015','shashi@gmail','4'



Method: GET

URL: http://localhost:8871/Service1.svc/insertItemDetails/'chair','Household','2','04-10-2015','shashi@

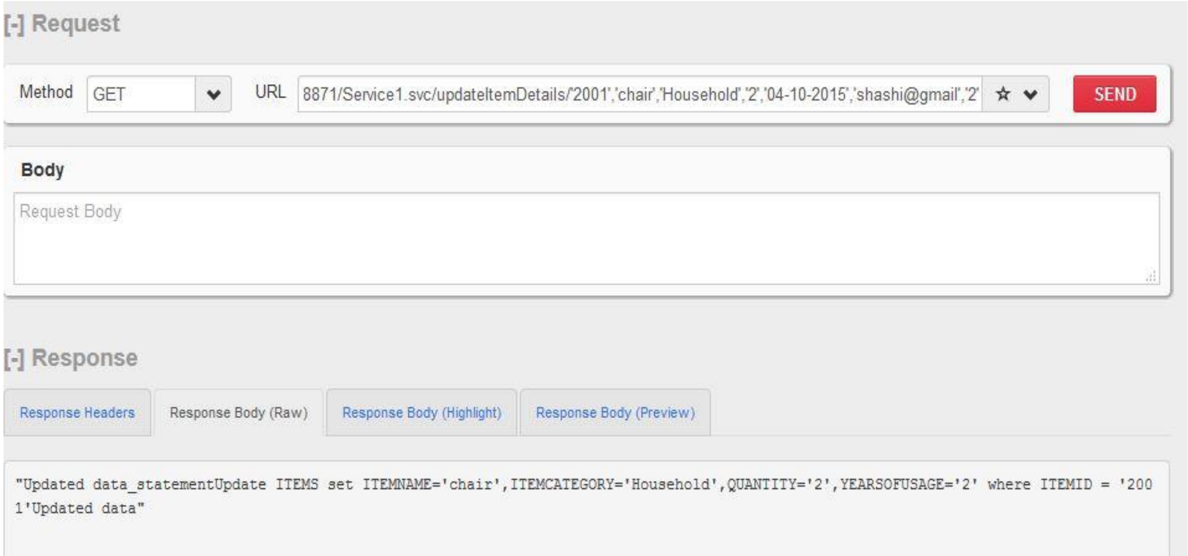
Request Body

```
Status Code       : 200 OK
Cache-Control     : private
Connection        : Close
Content-Length    : 15
Content-Type      : application/json; charset=utf-8
Date              : Mon, 13 Apr 2015 22:34:32 GMT
Server            : ASP.NET Development Server/10.0.0.0
X-AspNet-Version  : 4.0.30319
```

IV. Item Updating Service:

This service is used to update the details of an item in the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

URL: http://10.0.2.2:8871/Service1.svc/extractItemDetails/shashi@gmail



Method: GET

URL: 8871/Service1.svc/updateItemDetails/'2001','chair','Household','2','04-10-2015','shashi@gmail','2'

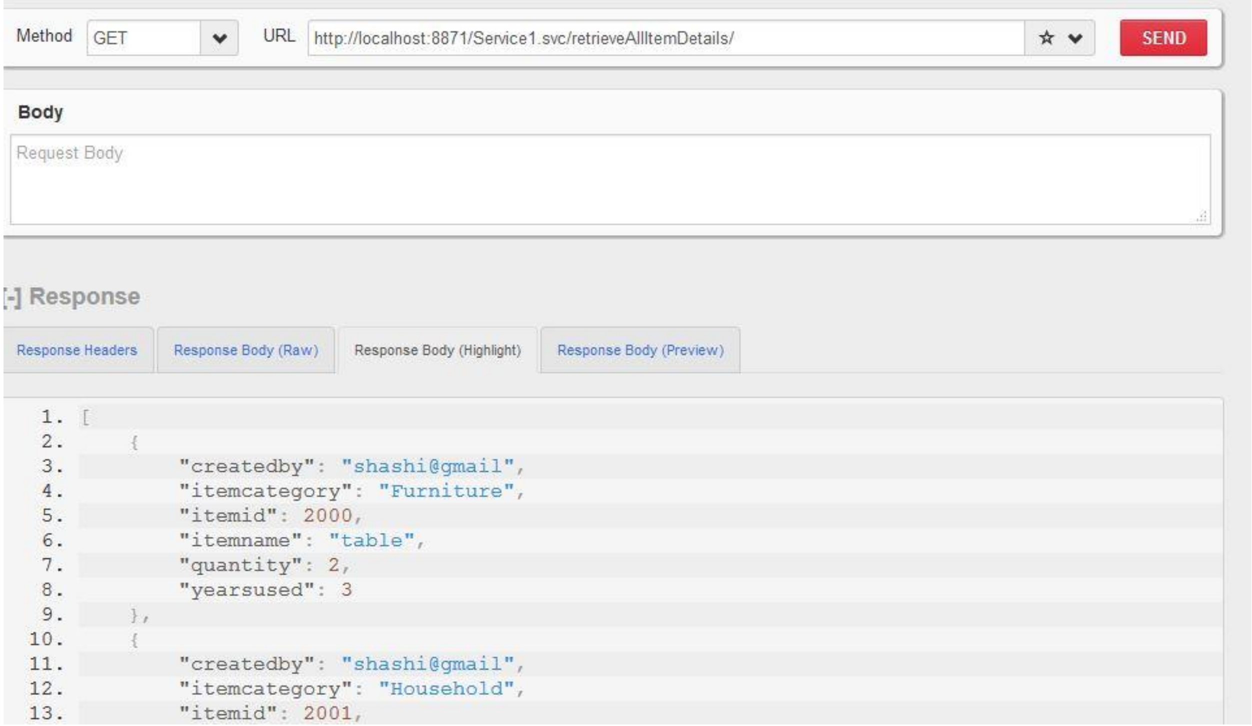
Request Body

```
"Updated data_statementUpdate ITEMS set ITEMNAME='chair',ITEMCATEGORY='Household',QUANTITY='2',YEARSOFUSAGE='2' where ITEMID = '200
1'Updated data"
```

V. Item Retrieval Service:

This service is used to retrieve the list of items from the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and get the required details.

URL: http://10.0.2.2:8871/Service1.svc/retrieveAllItemDetails/

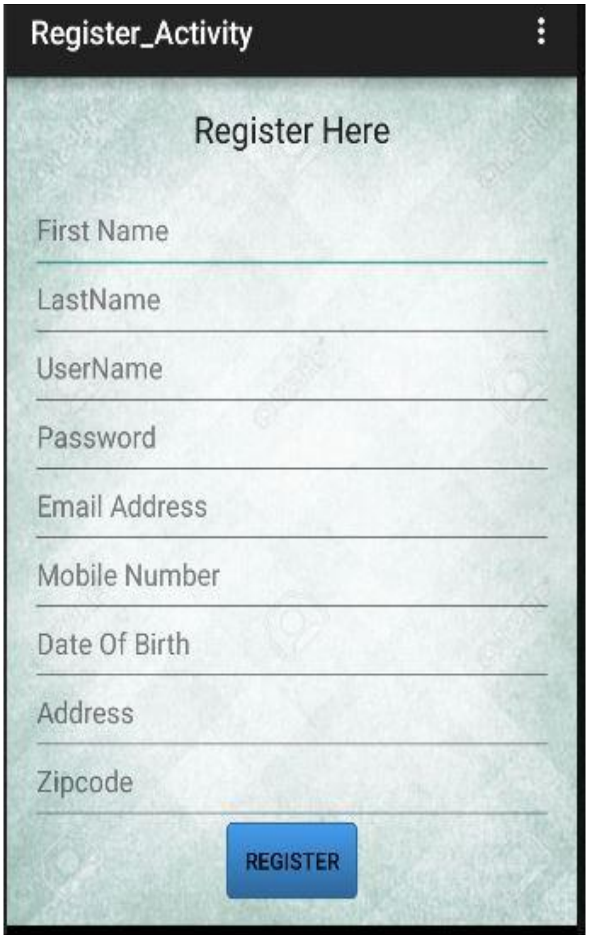


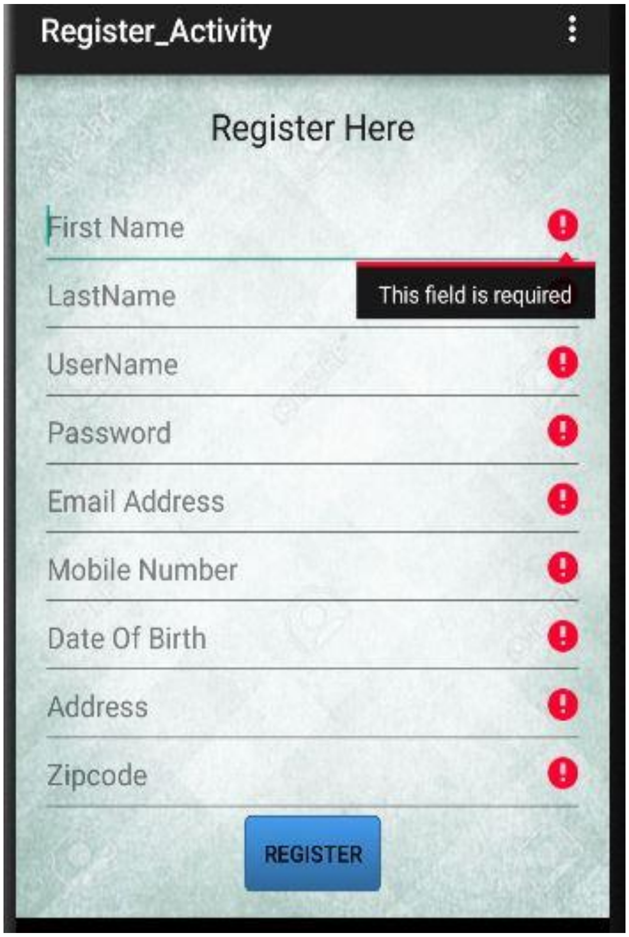
```
[
    {
        "createdby": "shashi@gmail",
        "itemcategory": "Furniture",
        "itemid": 2000,
        "itemname": "table",
        "quantity": 2,
        "yearsused": 3
    },
    {
        "createdby": "shashi@gmail",
        "itemcategory": "Household",
        "itemid": 2001,
```

- **Implementation of User Interface:**

I. User Registration Screen:

The user has to first register to the application by giving the required data like First Name, Last Name, User Name, Password, Email ID, Mobile Number, Date of Birth, Address and Zip Code.

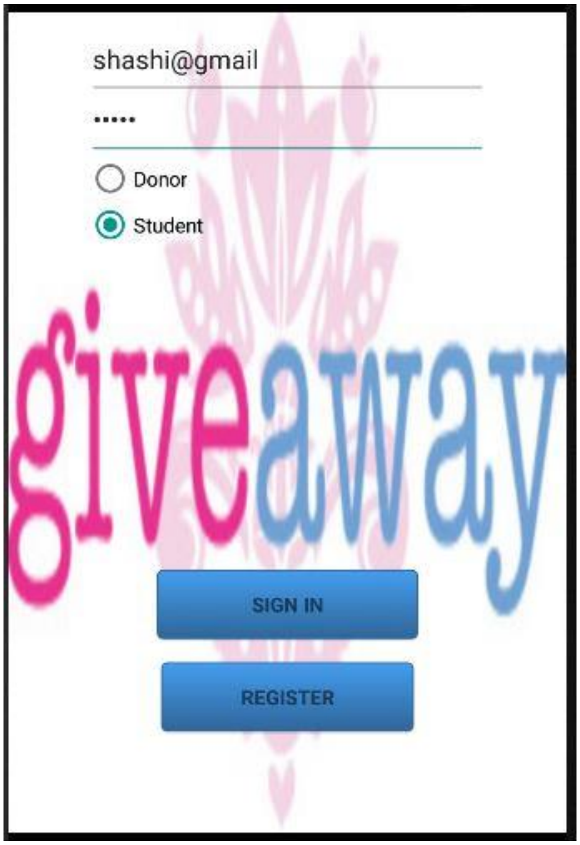




This field is required

II. Login Screen:

The user after the successful registration has to login to the application by using his/her credentials. He has to select one of the options between the donor and the student.

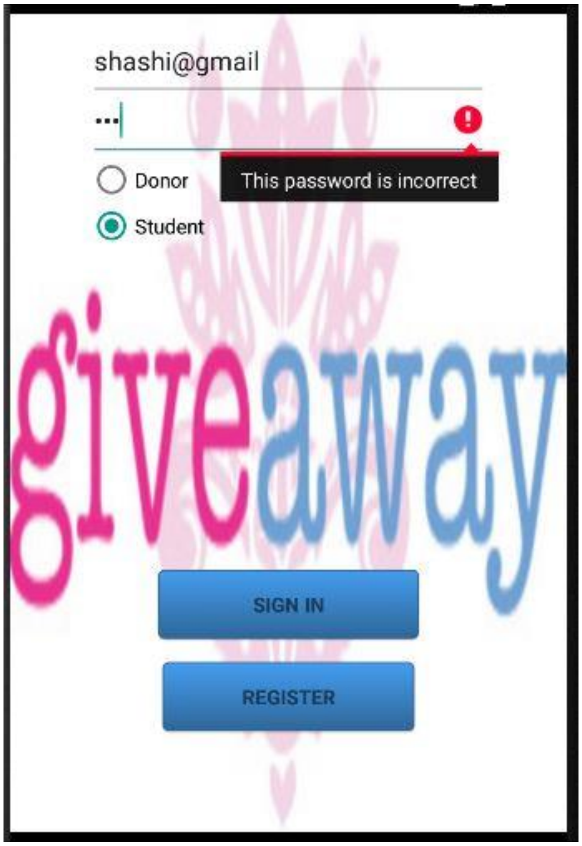


shashi@gmail

••••••

○ Donor

● Student



shashi@gmail

•••

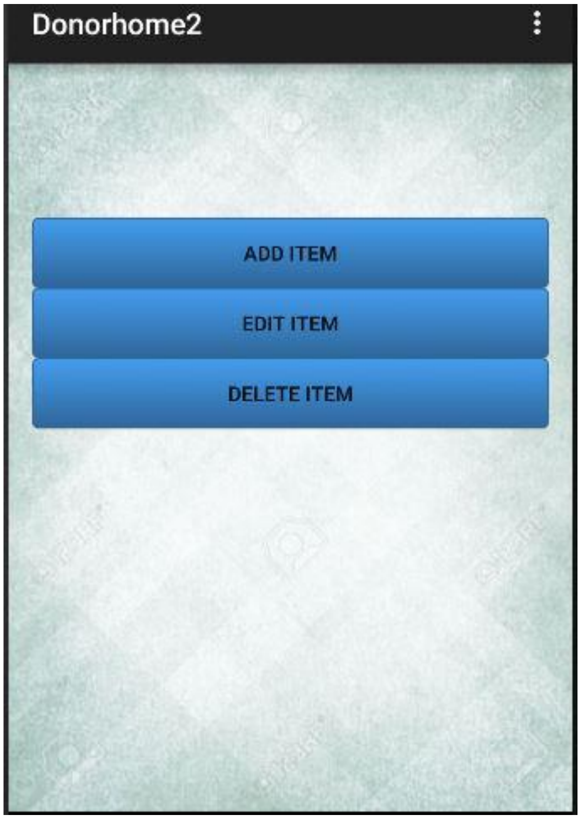
○ Donor

● Student

This password is incorrect

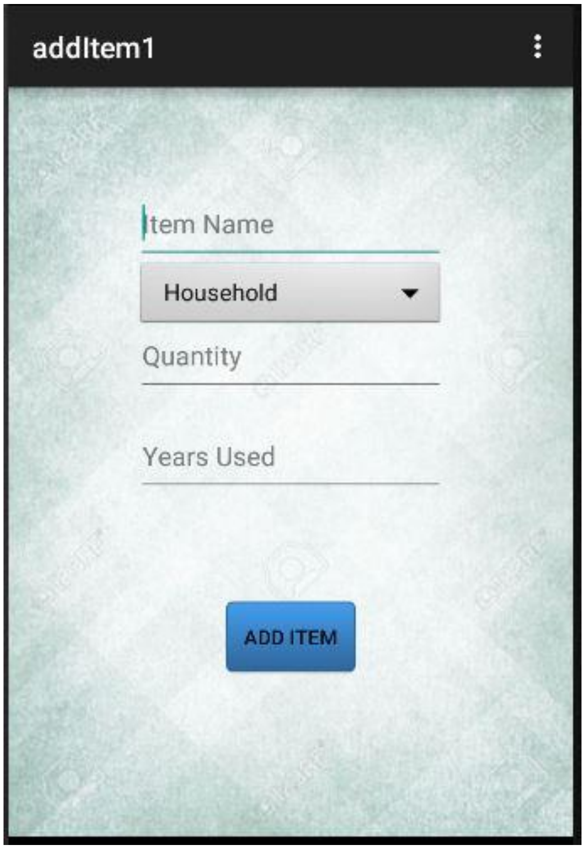
III. Donor Home Screen:

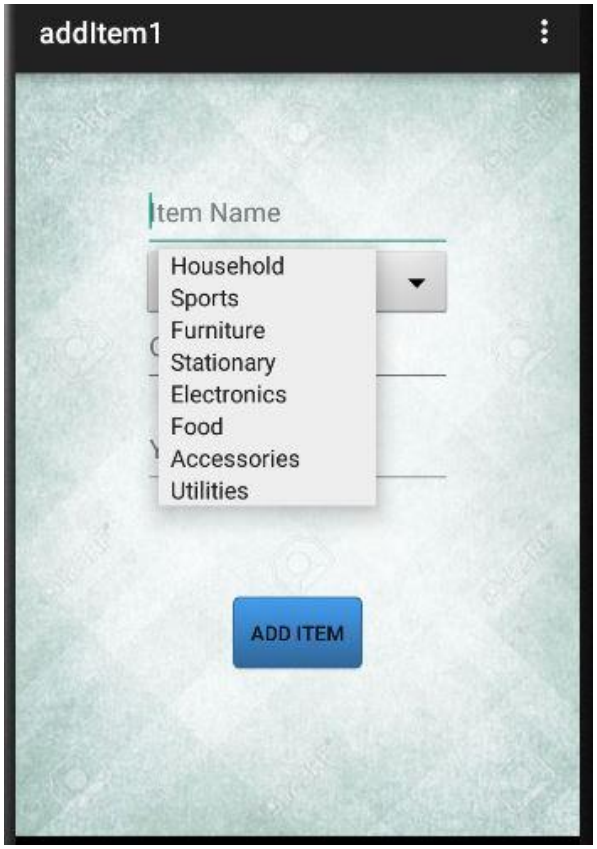
This screen contains the tabs for selecting the next operation. It has Add Item, Edit Item and Delete Item Tabs.



IV. Item Addition Screen:

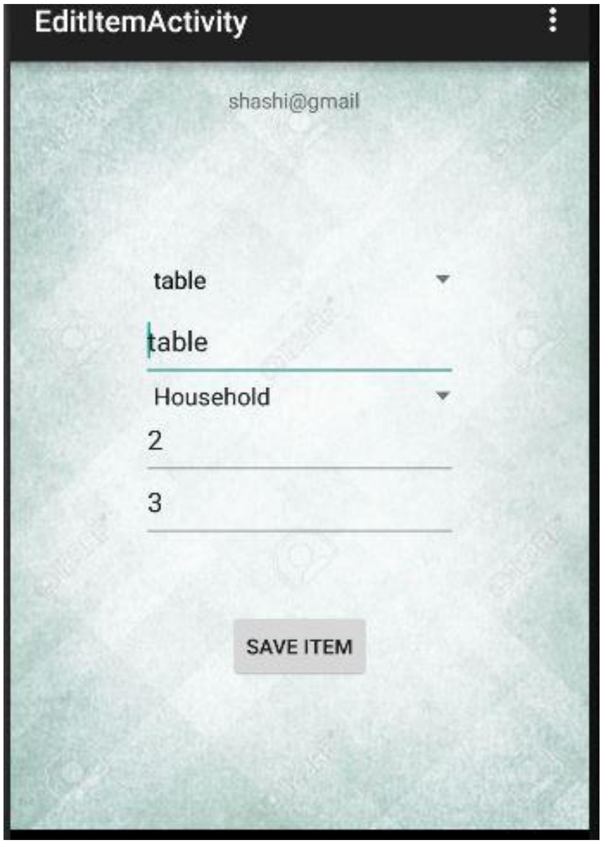
The user uses this screen for adding an item for Giveaway. He has to give the details like Item Name, Item Category, Quantity and the number of years used.





V. Item Updating Screen:

If the user wants to update the information of the items which he/she has posted, he can use this screen.



shashi@gmail

table

table

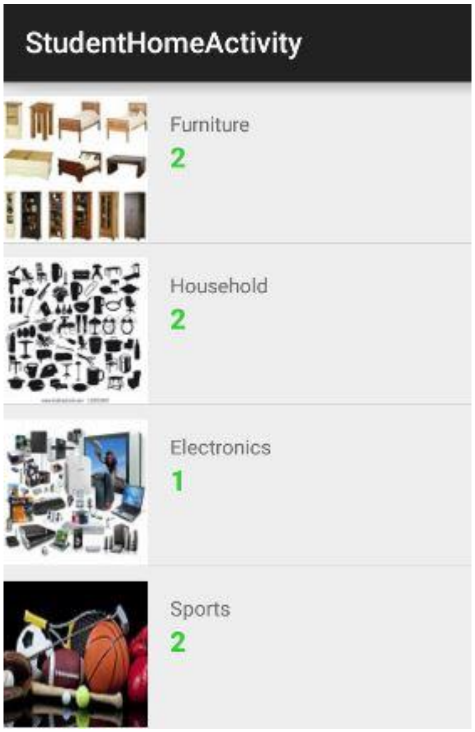
Household

2

3

VI. Student Home Screen:

After the successful login, the student will be redirected to this screen, where the list of all items will be populated.



VII. Item Selection Screen:

The student can select any item using this screen.

VIII. Subscription Screen:

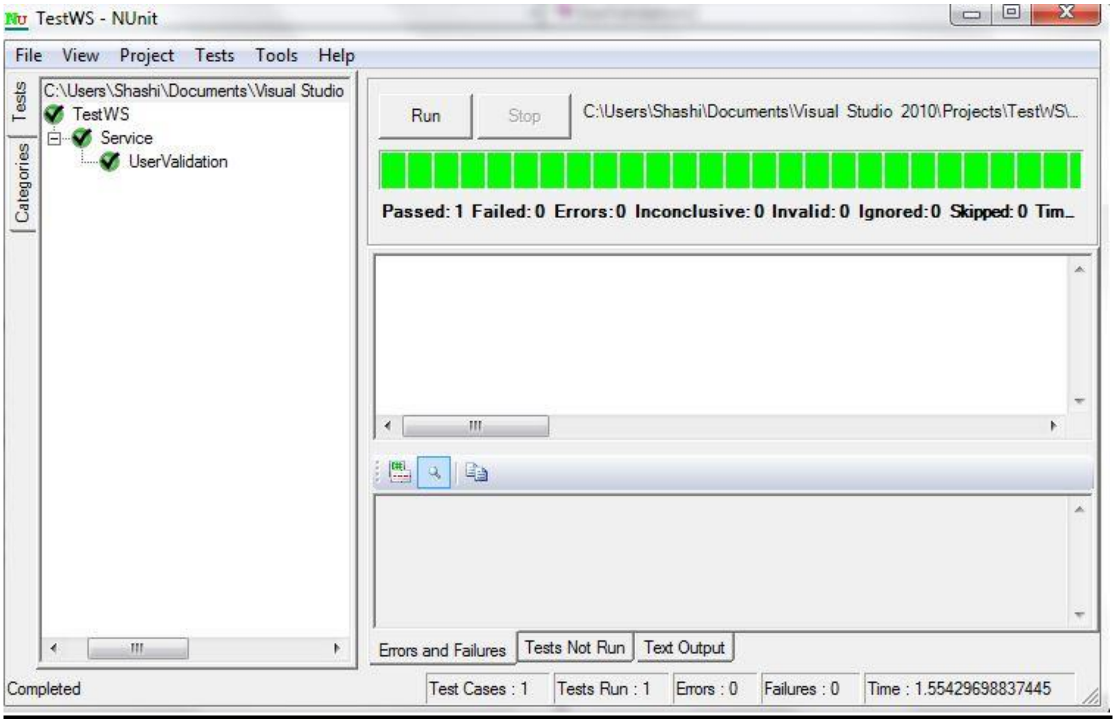
The Student can subscribe to any item which he likes using this screen.

Testing:

Functional Testing

NUnit Test Cases:

a. User Login:



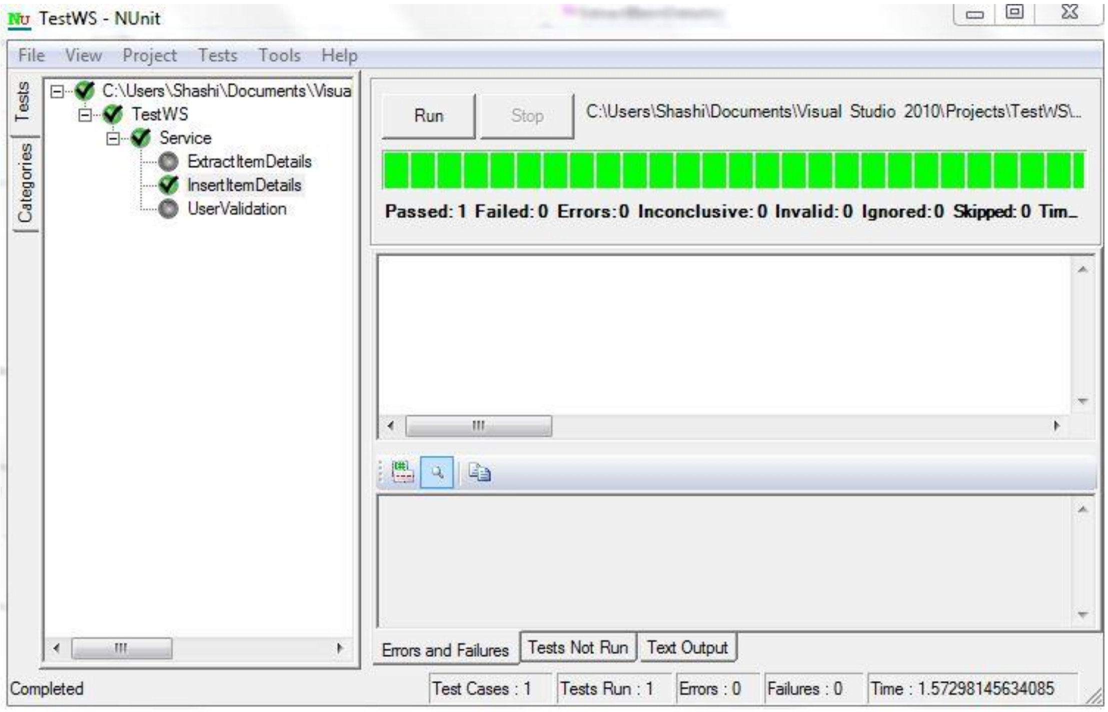
C:\Users\Shashi\Documents\Visual Studio
TestWS
Service
UserValidation

C:\Users\Shashi\Documents\Visual Studio 2010\Projects\TestWS\...

Passed: 1 Failed: 0 Errors: 0 Inconclusive: 0 Invalid: 0 Ignored: 0 Skipped: 0 Tim...

Completed | Test Cases : 1 | Tests Run : 1 | Errors : 0 | Failures : 0 | Time : 1.55429698837445

b. Item Addition:



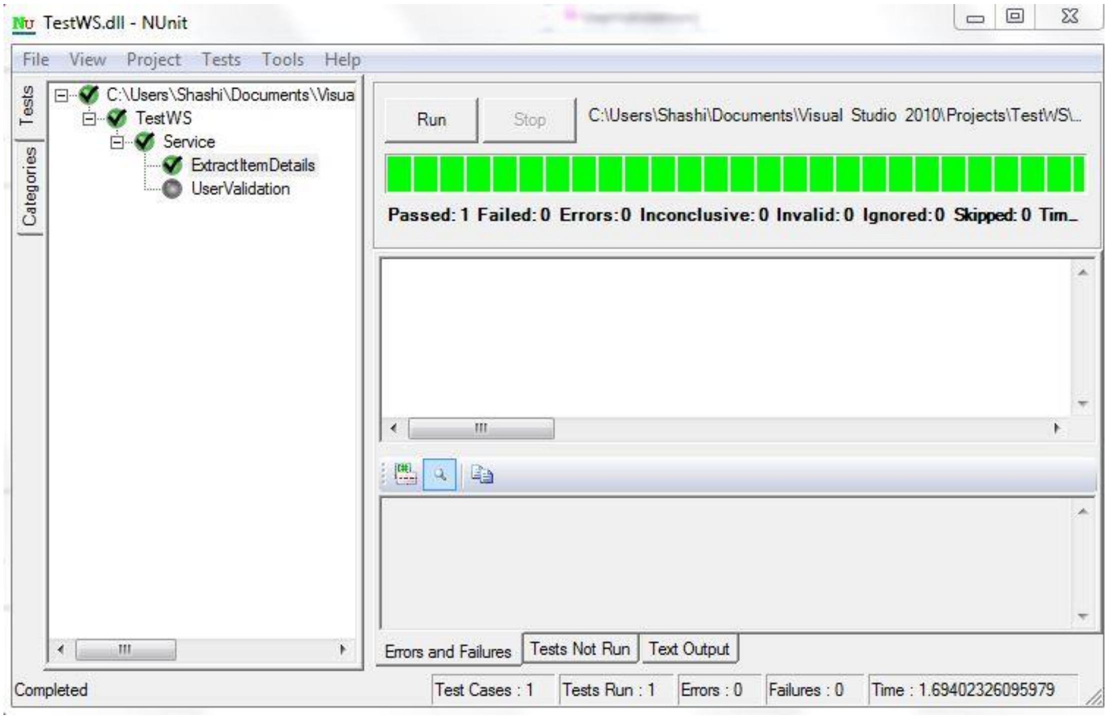
C:\Users\Shashi\Documents\Visua
TestWS
Service
ExtractItemDetails
InsertItemDetails
UserValidation

C:\Users\Shashi\Documents\Visual Studio 2010\Projects\TestWS\...

Passed: 1 Failed: 0 Errors: 0 Inconclusive: 0 Invalid: 0 Ignored: 0 Skipped: 0 Tim...

Completed | Test Cases : 1 | Tests Run : 1 | Errors : 0 | Failures : 0 | Time : 1.57298145634085

c. Item List Retrieval:



C:\Users\Shashi\Documents\Visua
TestWS
Service
ExtractItemDetails
UserValidation

C:\Users\Shashi\Documents\Visual Studio 2010\Projects\TestWS\...

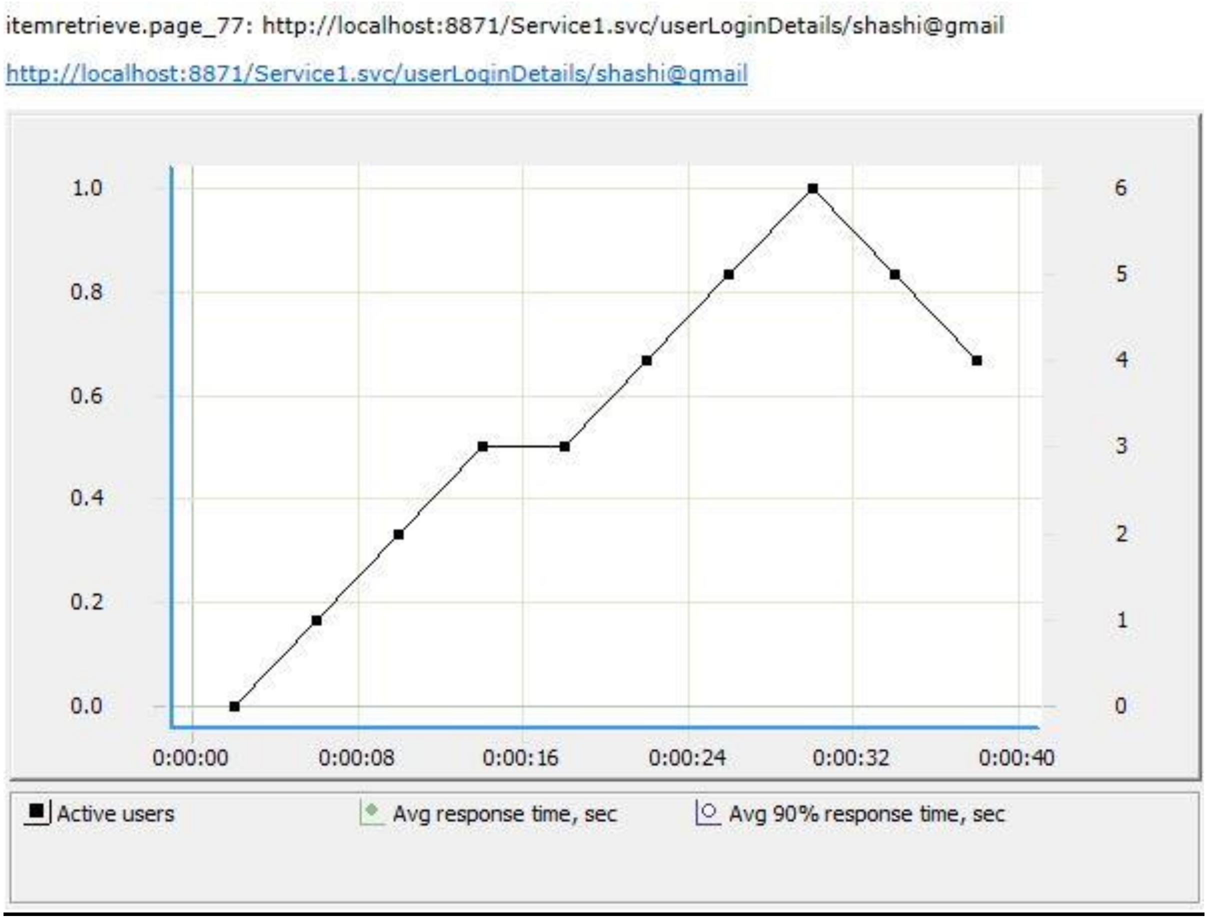
Passed: 1 Failed: 0 Errors: 0 Inconclusive: 0 Invalid: 0 Ignored: 0 Skipped: 0 Tim...

Completed | Test Cases : 1 | Tests Run : 1 | Errors : 0 | Failures : 0 | Time : 1.69402326095979

Deployment Testing:

Wapt Tool:

WAPT provides the most affordable and easy to configure load testing solution. It works as a single application that includes test design and load generation features. We can record tests using any desktop or mobile browser, design and optimize them with help of a convenient product GUI and run test scenarios with up to approximately 2,000 concurrent virtual users. Our target web site can run under any OS, including all UNIX and Windows platforms. It can be implemented with any web technology and have a distributed architecture consisting of multiple servers and databases. The only requirement is the availability of a web interface through which WAPT can simulate the activity of real site visitors. Here we have tested one of our service in the Wapt tool and the plotted graph shows the results.



itemretrieve.page_77: http://localhost:8871/Service1.svc/userLoginDetails/shashi@gmail

http://localhost:8871/Service1.svc/userLoginDetails/shashi@gmail

Deployment:

- **SrumDo:**

https://www.scrumdo.com/projects/project/giveaway/iteration/121737

- **GitHub:**

https://github.com/sashi987/ASE/tree/master/Increment3

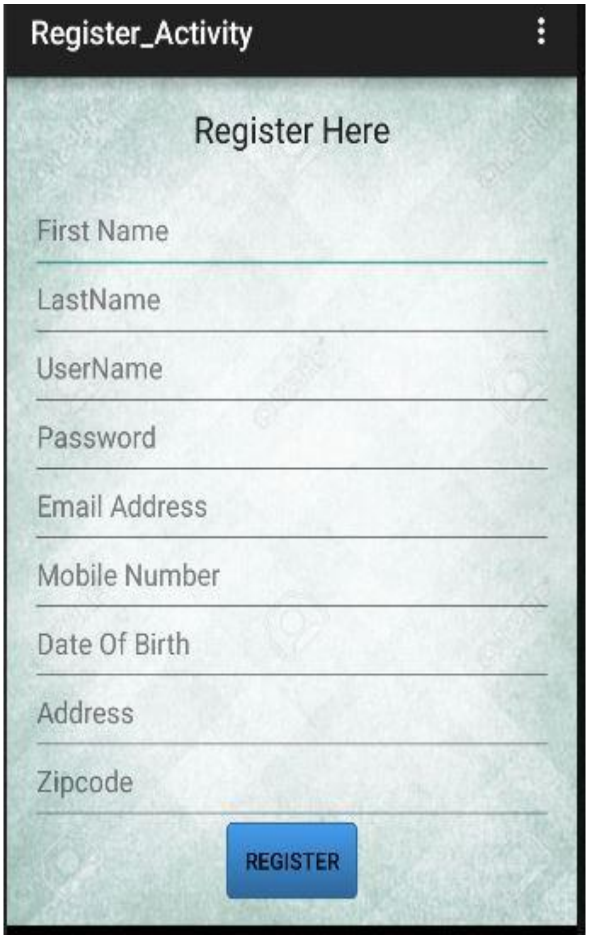
Report:

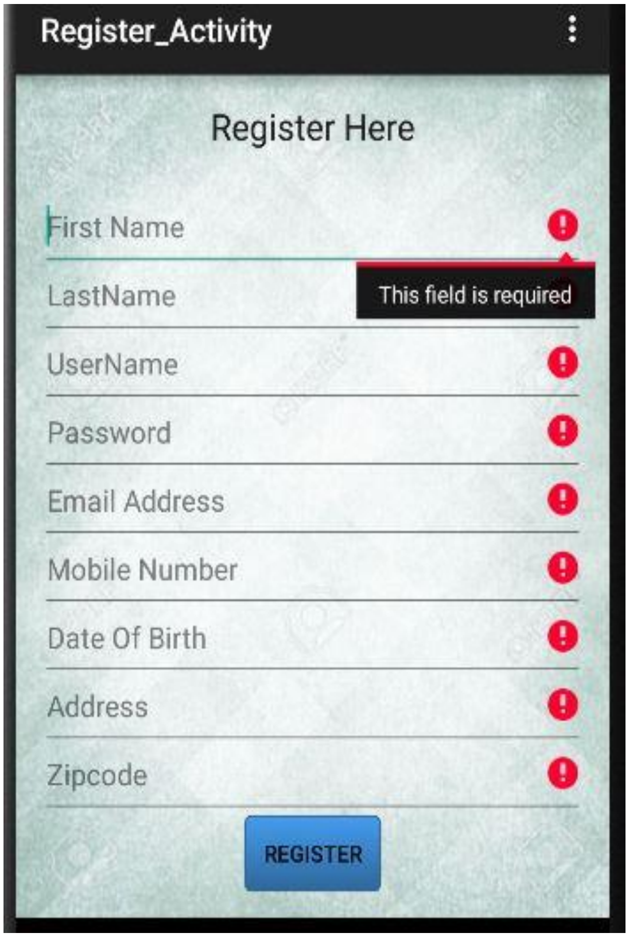
The Give Away application is being developed as an Android application using Android SDK framework and SQL Server Database for data.

The following are the screens that we are using as part of our project,

I. **User Registration Screen:**

The user has to first register to the application by giving the required data like First Name, Last Name, User Name, Password, Email ID, Mobile Number, Date of Birth, Address and Zip Code.

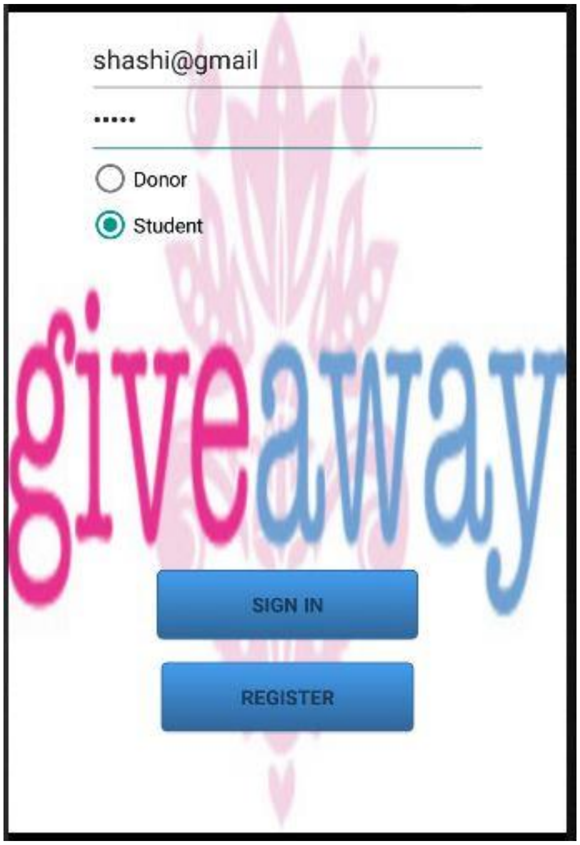




This field is required

II. Login Screen:

The user after the successful registration has to login to the application by using his/her credentials. He has to select one of the options between the donor and the student.

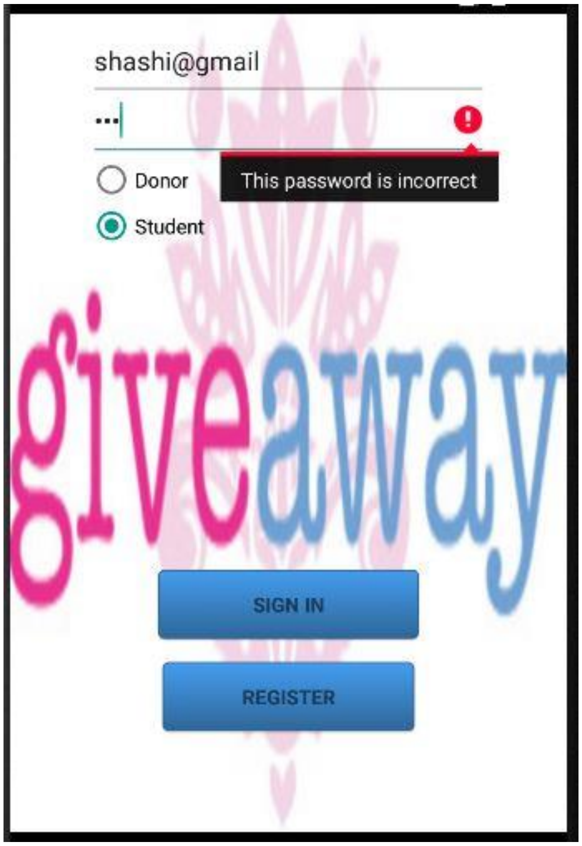


shashi@gmail

••••••

○ Donor

● Student



shashi@gmail

•••

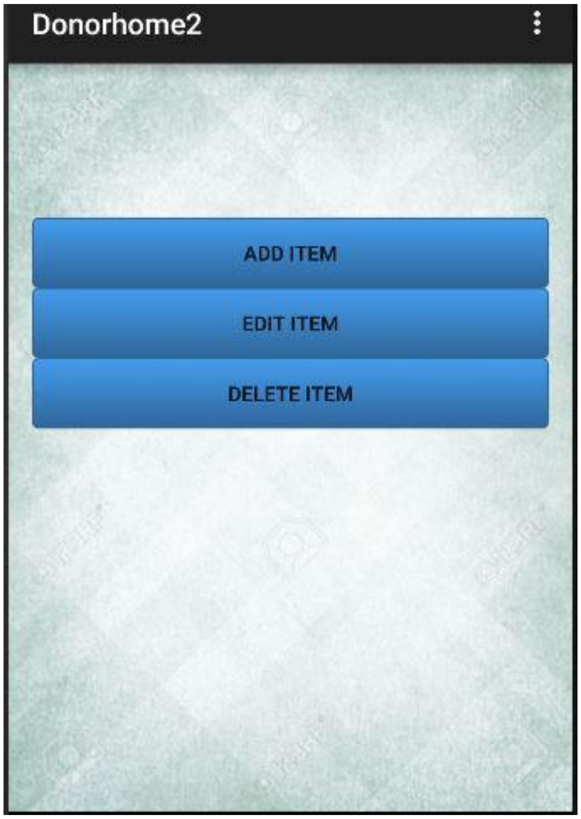
This password is incorrect

○ Donor

● Student

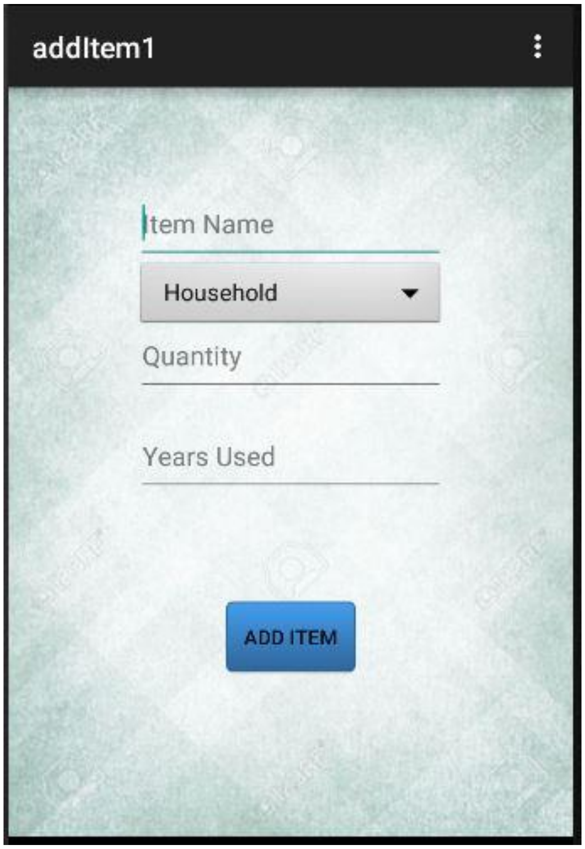
III. Donor Home Screen:

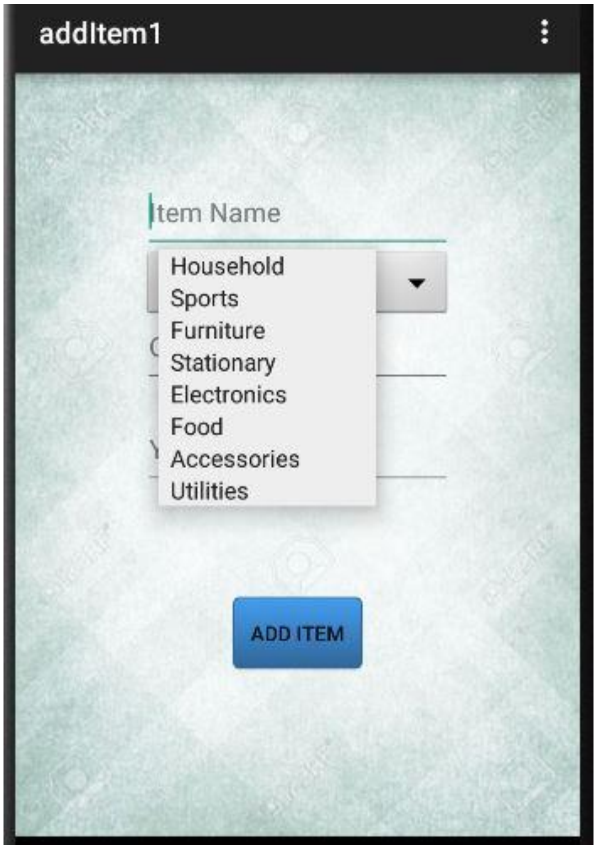
This screen contains the tabs for selecting the next operation. It has Add Item, Edit Item and Delete Item Tabs.



IV. Item Addition Screen:

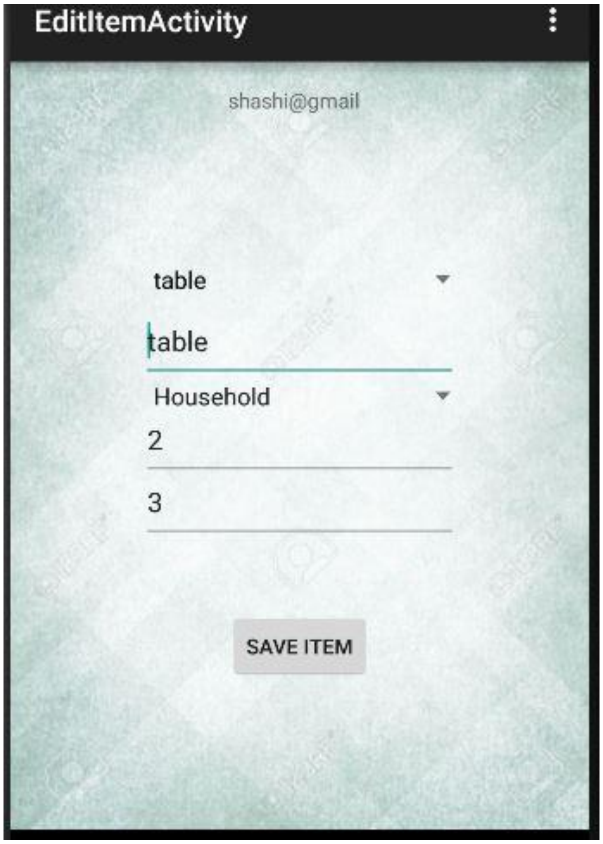
The user uses this screen for adding an item for Giveaway. He has to give the details like Item Name, Item Category, Quantity and the number of years used.





V. Item Updating Screen:

If the user wants to update the information of the items which he/she has posted, he can use this screen.



shashi@gmail

table

table

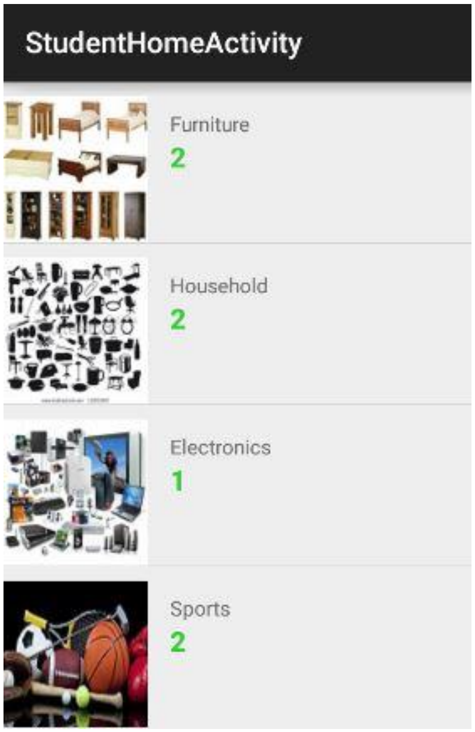
Household

2

3

VI. Student Home Screen:

After the successful login, the student will be redirected to this screen, where the list of all items will be populated.



StudentHomeActivity

Furniture
2

Household
2

Electronics
1

Sports
2

Project Management:

Work completed:

- Description:
  a. Registration and Login Services for Users
  b. Adding/Updating/Deleting an Item
  c. Retrieval of the list of Items

- Responsibility (Task, Person):

  a. **Sashidhar Reddy Gowra**: Creation of Services for User Login, User Registration, Item Addition, Item Updating and Item retrieval.

  b. **Venkataramana Yashwant Kumar Palisetty**: Creation of UI Screens for User Login and Registration and the corresponding validations.

  c. **Ravikanth Devanaboyina**: Creation of UI Screen for Displaying the Item list.

  d. **Anudeep Reddy Gujjula**: Creation of UI Screens for Item Addition, Item Updating and Item Deletion.

- Time taken (#hours):
  UI Design, Service Creation and Client Validations: 80 hrs.

- Contributions (members/percentage):
  a. Sashidhar Reddy Gowra - 50%
  b. Venkataramana Yashwant Kumar Palisetty - 20%
  c. Ravikanth Devanaboyina - 10%
  d. Anudeep Reddy Gujjula - 20%.

Work to be completed:

- Description: Using the already existing Google maps API for sharing the address location of a user, Unit Testing, System Testing, Testing the application on Android Devices
- Responsibility (Task, Person):
  Creating Web services, Hash Functions: Sashidhar Reddy Gowra, Ravikanth Devanaboyina.
  Populating screens, UI alignments: Venkataramana Yashwant Kumar Palisetty, Anudeep Reddy Gujjula.
- Time to be taken (estimated #hours) 100 hrs.

Issues/Concerns: Image insertion using service and using Hash functions to store user details in Database.

Project Increment 4

Objective:

The prime objective of this iteration is to create the web services to populate the data UI screens for the modules like Registration Module, Login Module, Item Addition Module, Item Updating Module, Item Deletion Module, Item Selection Module and Subscription Modules.

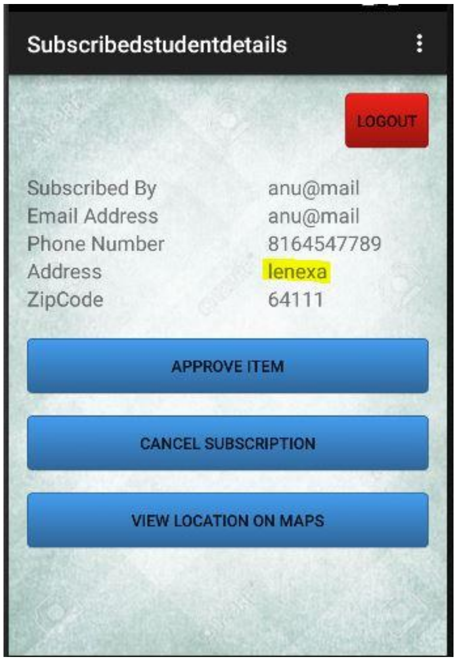
Import Existing API or Services:

Google Maps API:

We have implemented Google maps API for locating the location of user who wish to grab items from donor. This service can be used on android mobile to locate the user and to get the driving directions to the student's destination which will help donor in delivering the products.

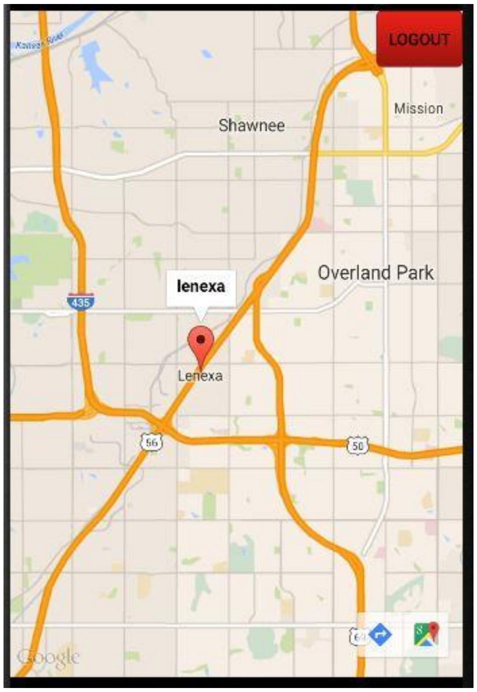
https://developers.google.com/maps/documentation/android/

Subscribed Student Details Screen: This screen has address details of students who subscribed for an item.

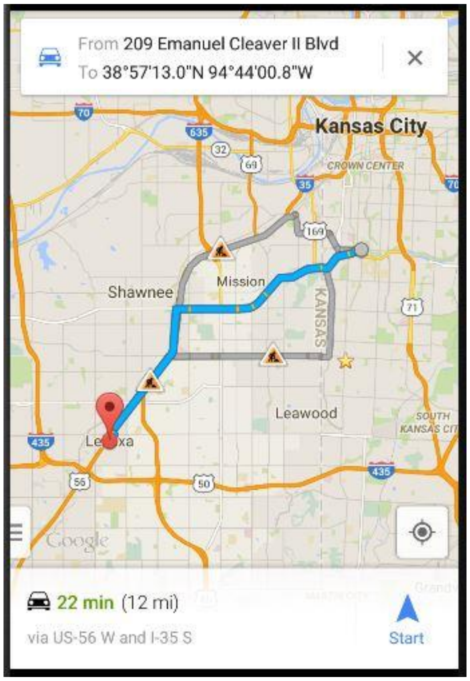


Subscribed By	anu@mail
Email Address	anu@mail
Phone Number	8164547789
Address	lenexa
ZipCode	64111

Google Maps Screen: Here student location is identified on the Google maps and is pointed with a marker.



Google maps screen: Displaying estimated distance and time of travel from donor's location to student's location



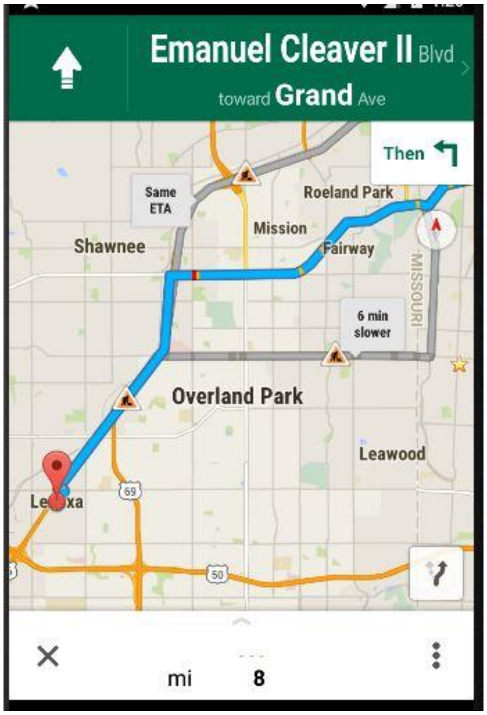
From 209 Emanuel Cleaver II Blvd

To 38°57'13.0"N 94°44'00.8"W

22 min (12 mi)

via US-56 W and I-35 S

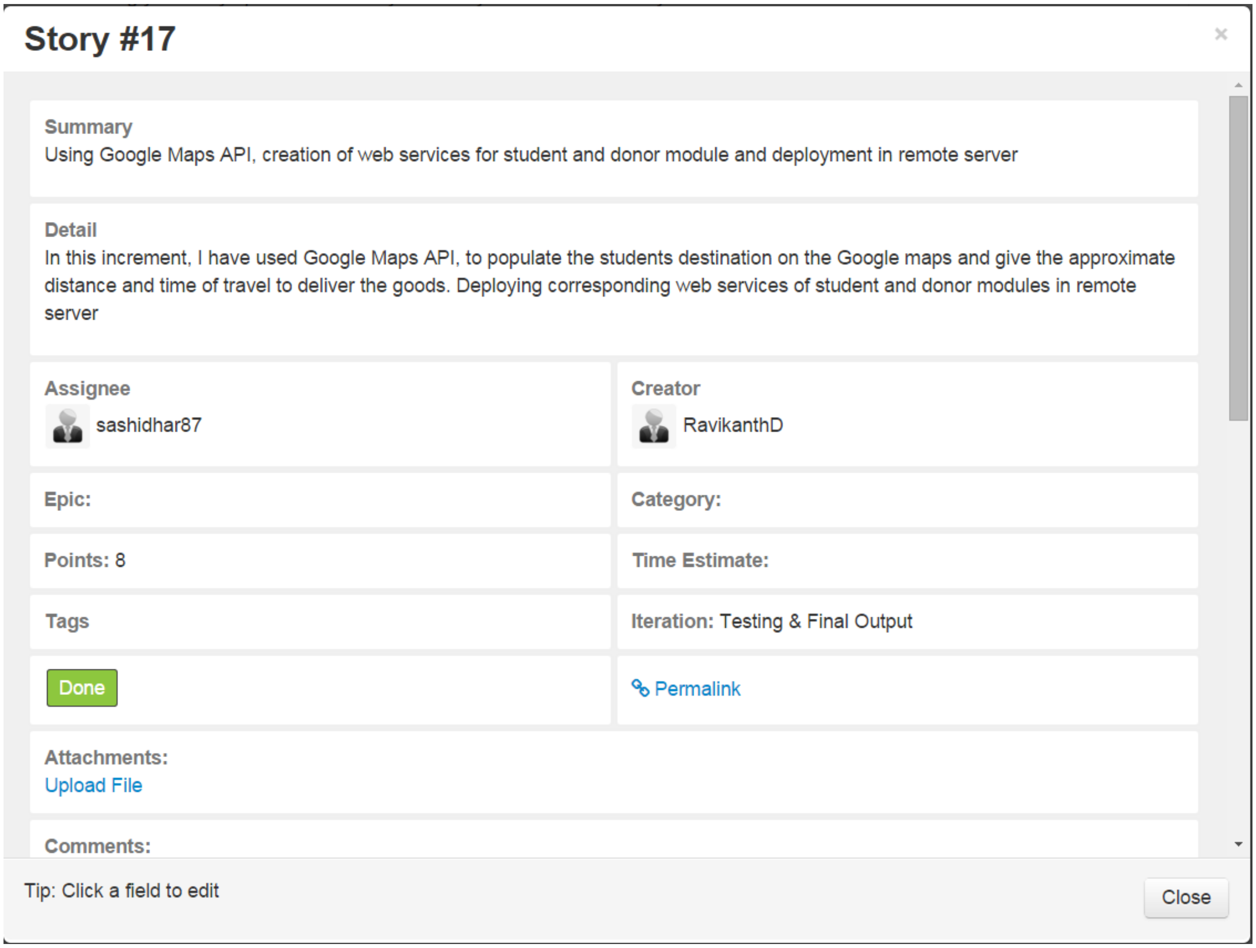
Driving directions to Students location:



Detail Design of Services:

- **User stories using ScrumDo:**

The following are the user stories we have created in the ScrumDo:



Story #17

Summary
Using Google Maps API, creation of web services for student and donor module and deployment in remote server

Detail
In this increment, I have used Google Maps API, to populate the students destination on the Google maps and give the approximate distance and time of travel to deliver the goods. Deploying corresponding web services of student and donor modules in remote server

Assignee: sashidhar87

Creator: RavikanthD

Epic:

Category:

Points: 8

Time Estimate:

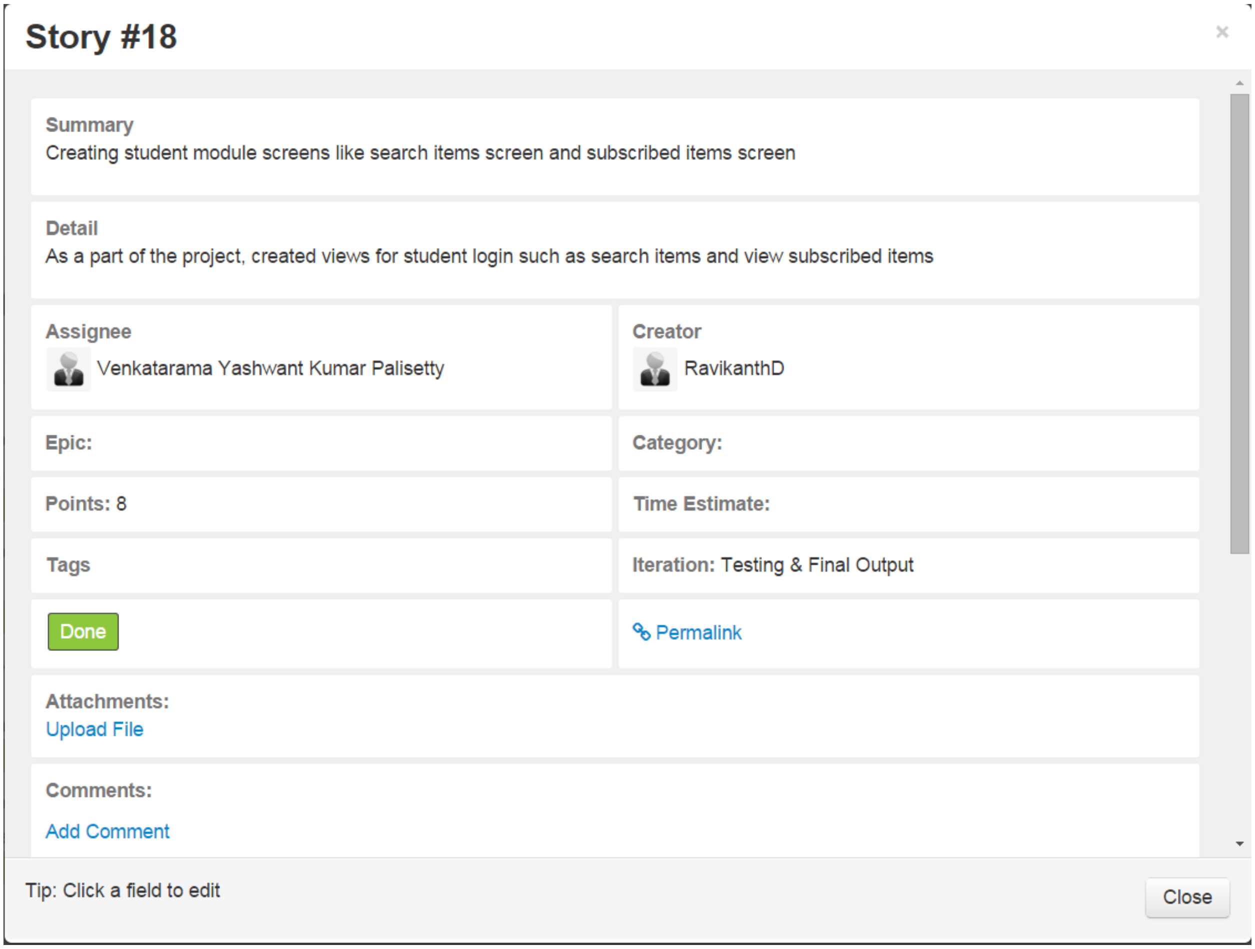
Tags

Iteration: Testing & Final Output

Attachments:

Comments:

Tip: Click a field to edit



Story #18

Summary
Creating student module screens like search items screen and subscribed items screen

Detail
As a part of the project, created views for student login such as search items and view subscribed items

Assignee
Venkatarama Yashwant Kumar Palisetty

Creator
RavikanthD

Epic:

Category:

Points: 8

Time Estimate:

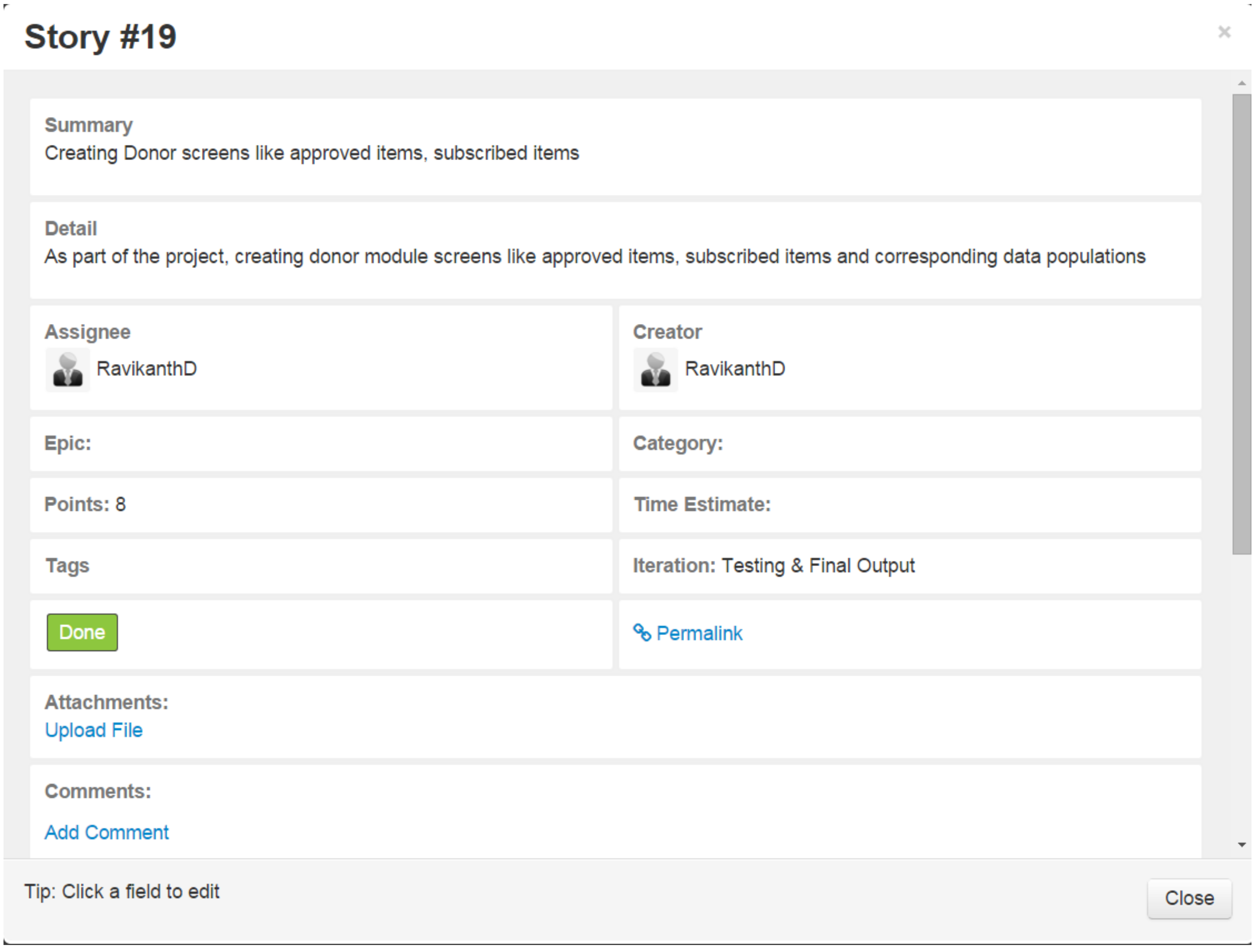
Tags

Iteration: Testing & Final Output

Attachments:

Comments:

Tip: Click a field to edit



Story #19

Summary
Creating Donor screens like approved items, subscribed items

Detail
As part of the project, creating donor module screens like approved items, subscribed items and corresponding data populations

Assignee: RavikanthD

Creator: RavikanthD

Epic:

Category:

Points: 8

Time Estimate:

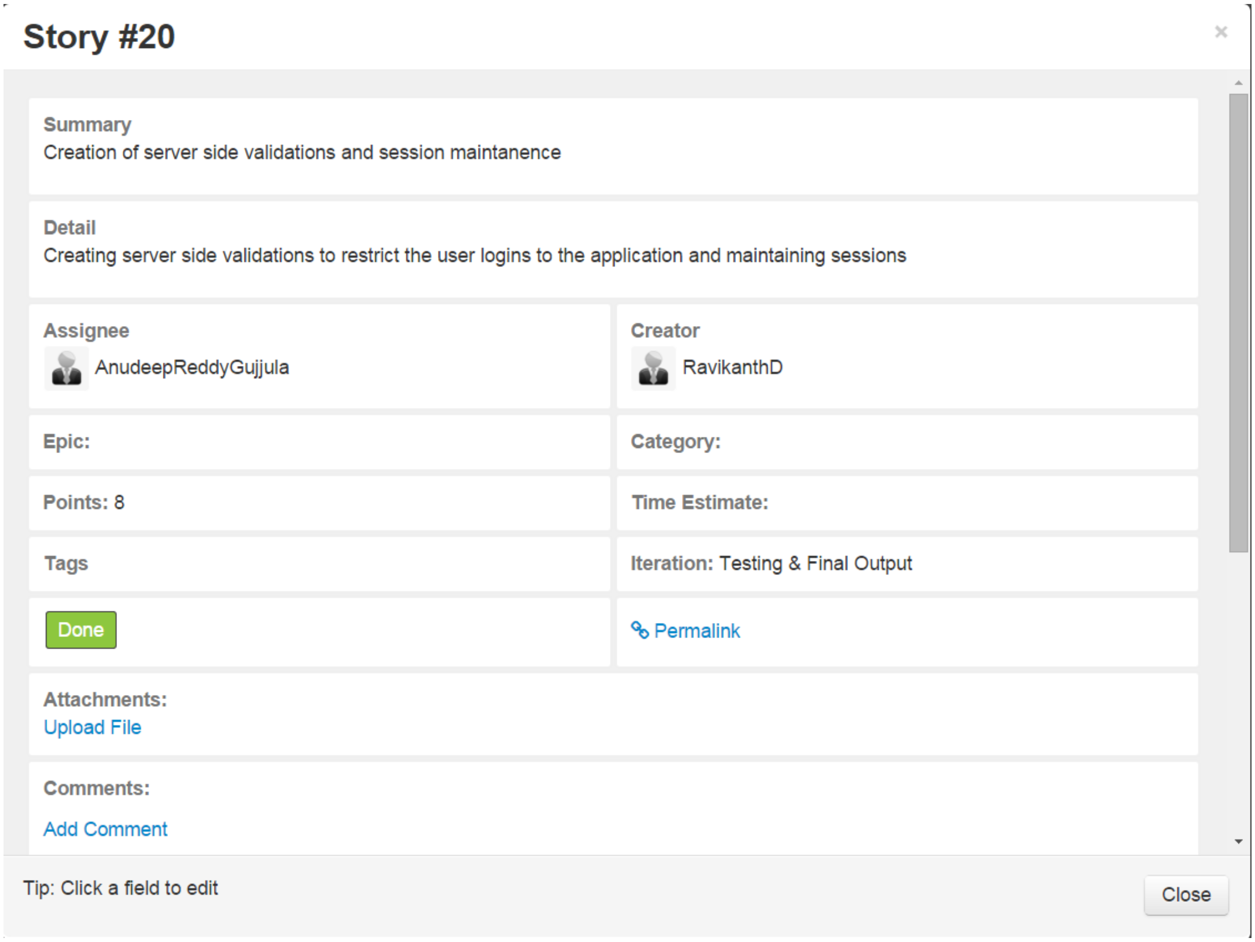
Tags

Iteration: Testing & Final Output

Attachments:
Upload File

Comments:
Add Comment

Tip: Click a field to edit



Story #20

Summary
Creation of server side validations and session maintanence

Detail
Creating server side validations to restrict the user logins to the application and maintaining sessions

Assignee: AnudeepReddyGujjula

Creator: RavikanthD

Epic:

Category:

Points: 8

Time Estimate:

Tags

Iteration: Testing & Final Output

Done

Permalink

Attachments:
Upload File

Comments:
Add Comment

Tip: Click a field to edit

Close

- **Service Description:**

I. **User Login Service:**
In this service, we are implementing User Login as a service, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

II. **User Registration Service:**
We are implementing the User Registration Service with the fields like First Name, Last Name, User Name, Password, Email ID, Mobile Number, Date of Birth, Address and Zip Code, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

III. **Subscribed Items Service (Student):** In this service, we have implemented a list view of subscribed items that are ready for donation by the donor and subscribed by the students.

IV. **Subscribed Student Details Service:** In this service, we have implemented a view to display all the students who have subscribed for the items.

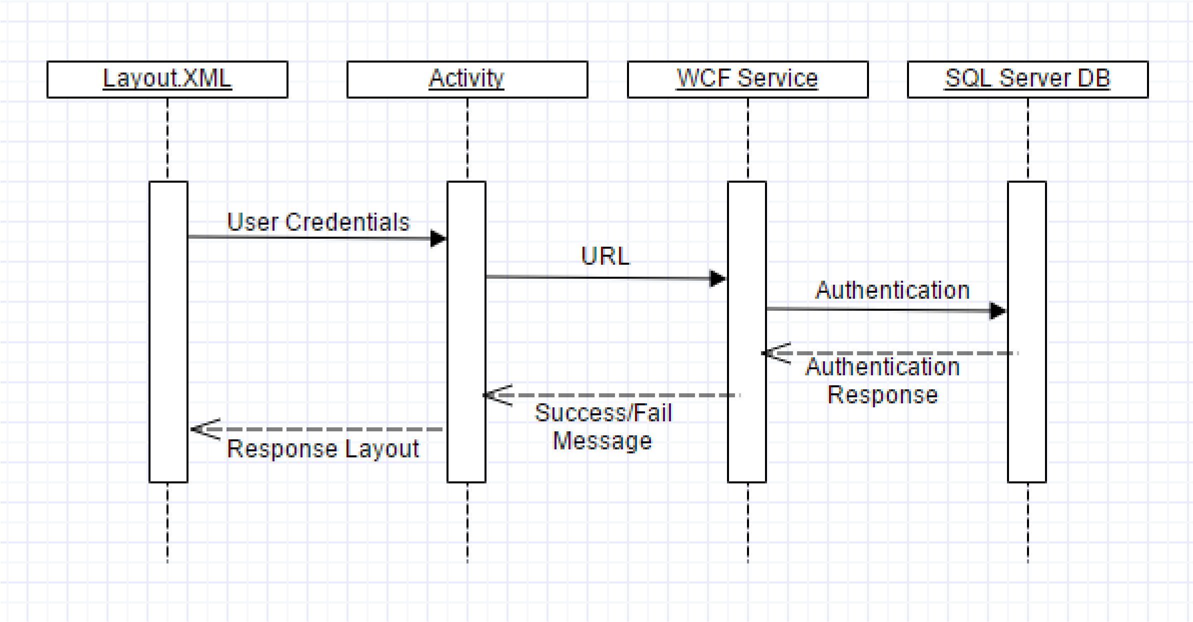
V. **Approve Item Service:** This service is used to approve the items to the students by the donor whenever the student subscribes for the item.

VI. **Search for item Service:** This service is used to search for an item and populating them on the corresponding screen.

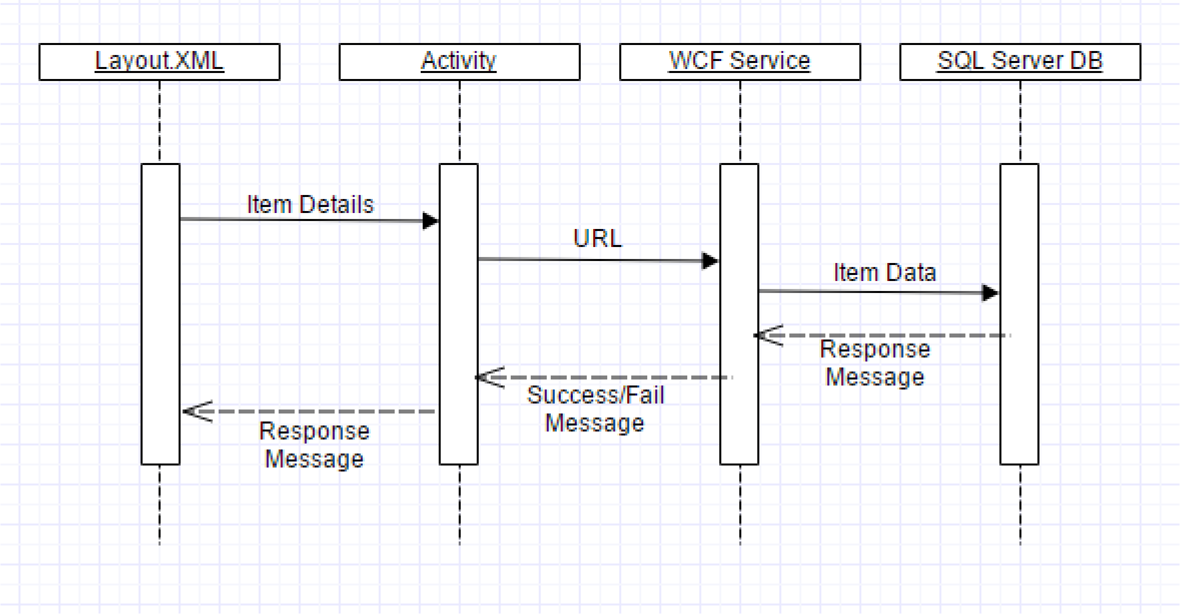
VII. **Subscribed Items Service (Donor):** In this service, we have implemented a list view of the items that the donor has donated and that are subscribed by the student.

- **Sequence Diagram:**

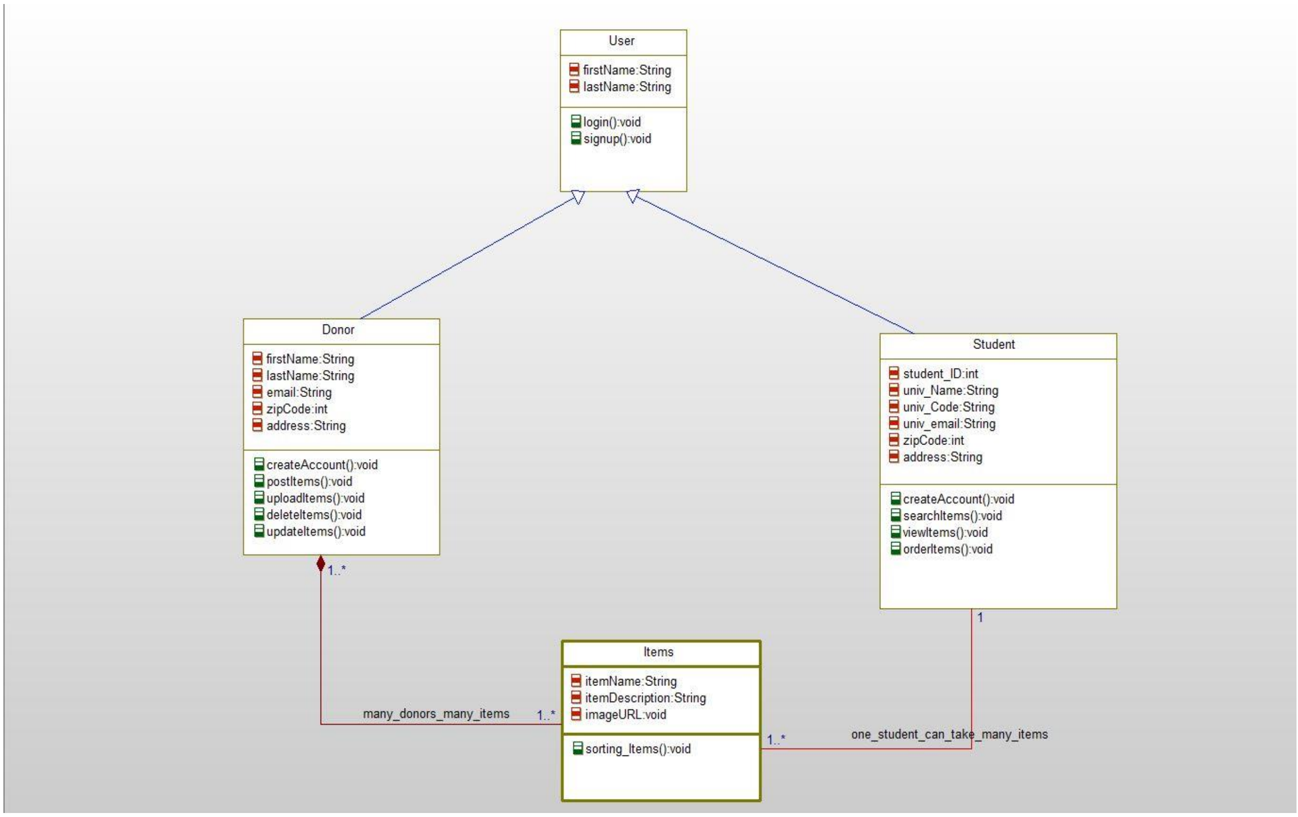
I. Sequence Diagram for User Login/Registration:



II. Sequence Diagram for adding/updating/retrieval of an Item:



- **Class Diagram:**



- **Design of Mobile Client Interface:**

a) Hardware Requirements:

I. 1GHz processor
II. SD card – 512 MB
III. RAM – 512 MB
IV. LED screen with touch enabled.

b) Software Requirements:

I. Operating system: Android
II. Version: Gingerbread (2.3) or advanced.

- **Design of Unit Test Cases:**

Test Case Id	**Module**	**Description**	**Expected Result**	**Status**
1	Registration	Click signup button without entering user details.	Message showing mandatory fields required	**Pass**
2	Registration	Numeric in First name ,Last name	Message showing numeric are not accepted.	**Pass**
3	Registration	Entering different passwords in password and retype password fields.	Message showing passwords should be same	**Pass**
4	Login	Click on submit with blank username and passwords.	Message showing mandatory fields required	**Pass**
5	Login	Entering unregistered username and password.	Application should not allow to login.	**Pass**
6	Login	Entering a valid username and Password	System should allow the user to login and a welcome page should be displayed.	**Pass**
7	Adding Items	Click Add Item button without entering the item details.	Message showing mandatory fields required	**Pass**
8	Adding Items	Numeric in item name	Message showing numeric are not accepted.	**Pass**
9	Adding Items	Alphabets and special characters in Quantity and Years used fields.	Message showing alphabets and special characters are not accepted.	**Pass**
10	Adding Items	Enter valid details.	System should allow the user to add the item.	**Pass**

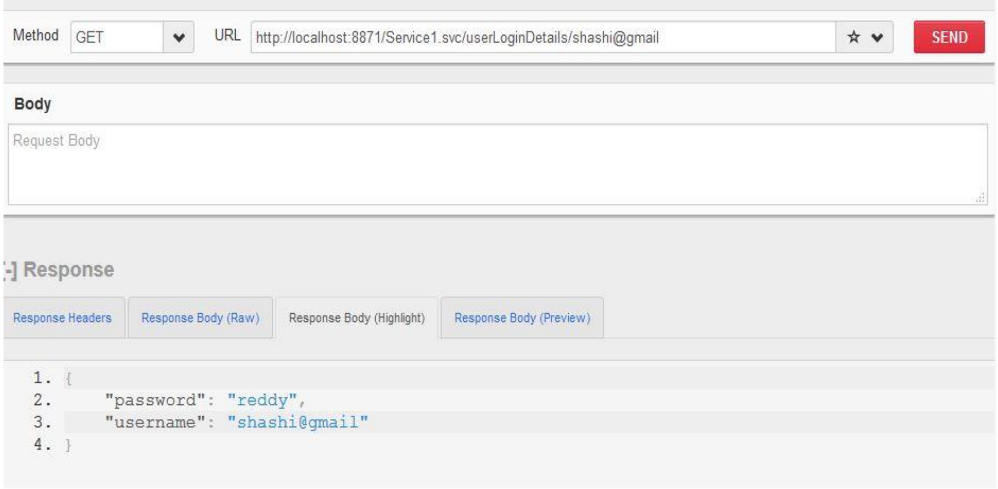
Implementation:

- **Implementation of Rest Services:**

I. User Login Service:

In this service, we are implementing User Login as a service, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

URL: http://10.0.2.2:8871/Service1.svc/userLoginDetails/shashi@gmail

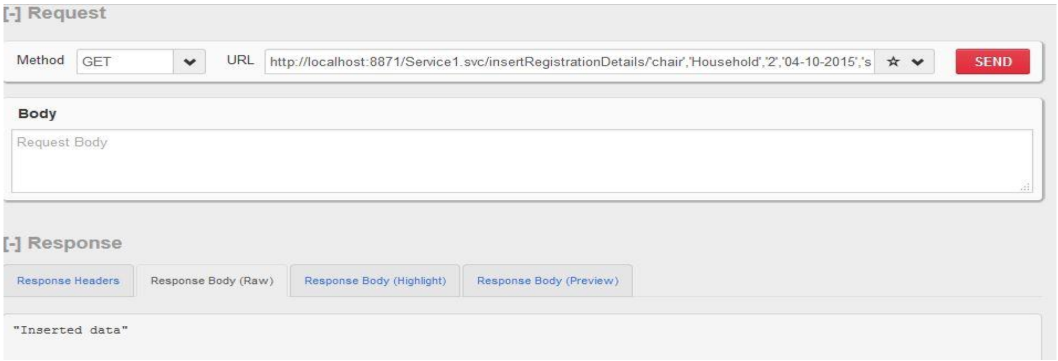


```
{
    "password": "reddy",
    "username": "shashi@gmail"
}
```

II. User Registration Service:

We are implementing the User Registration Service with the fields like First Name, Last Name, User Name, Password, Email ID, Mobile Number, Date of Birth, Address and Zip Code, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

URL: http://10.0.2.2:8871/Service1.svc/insertRegistrationDetails/'shashi','reddy','sashi','reddy','com','123','08-20-1990','kansas','64112'

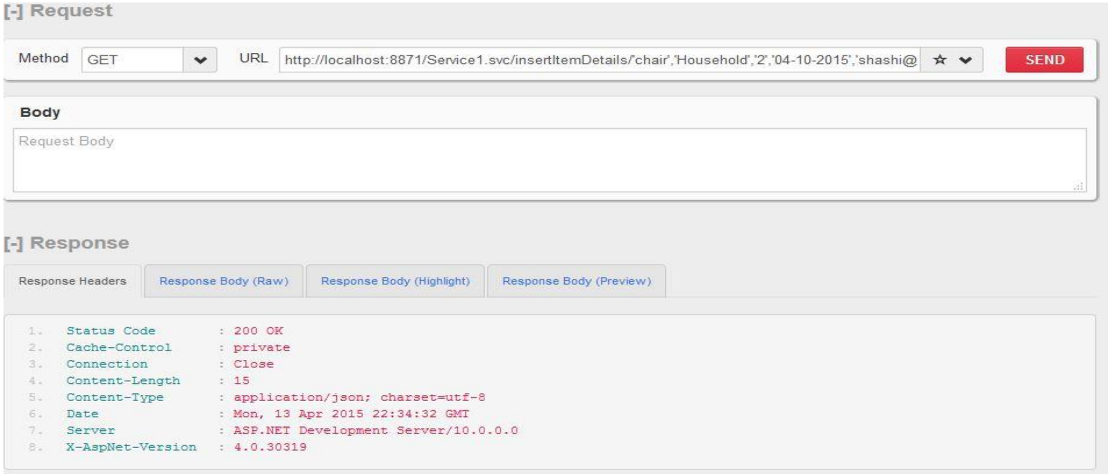

[-] Request
Method
GET
URL
http://localhost:8871/Service1.svc/insertRegistrationDetails/'chair','Household','2','04-10-2015','s
SEND
Body
Request Body
[-] Response
Response Headers
Response Body (Raw)
Response Body (Highlight)
Response Body (Preview)

```
"Inserted data"
```

III. Item Addition Service:

This service is used to add an item into the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

URL: http://10.0.2.2:8871/Service1.svc/insertItemDetails/'chair','Household','2','04-10-2015','shashi@gmail','4'



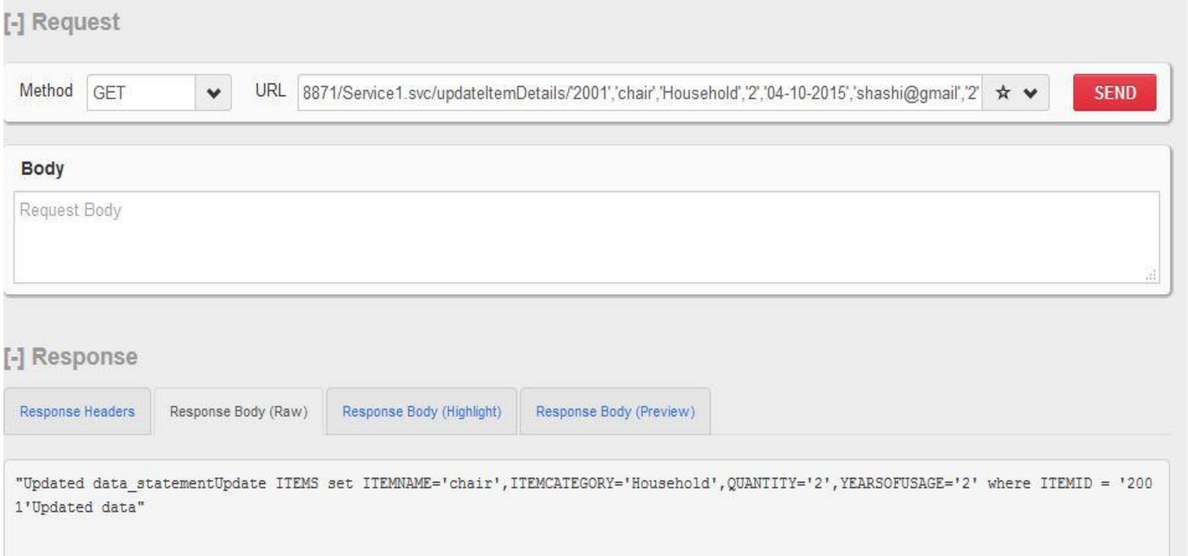
http://localhost:8871/Service1.svc/insertItemDetails/'chair','Household','2','04-10-2015','shashi@

```
Status Code        : 200 OK
Cache-Control      : private
Connection         : Close
Content-Length     : 15
Content-Type       : application/json; charset=utf-8
Date               : Mon, 13 Apr 2015 22:34:32 GMT
Server             : ASP.NET Development Server/10.0.0.0
X-AspNet-Version   : 4.0.30319
```

IV. Item Updating Service:

This service is used to update the details of an item in the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

URL: http://10.0.2.2:8871/Service1.svc/extractItemDetails/shashi@gmail



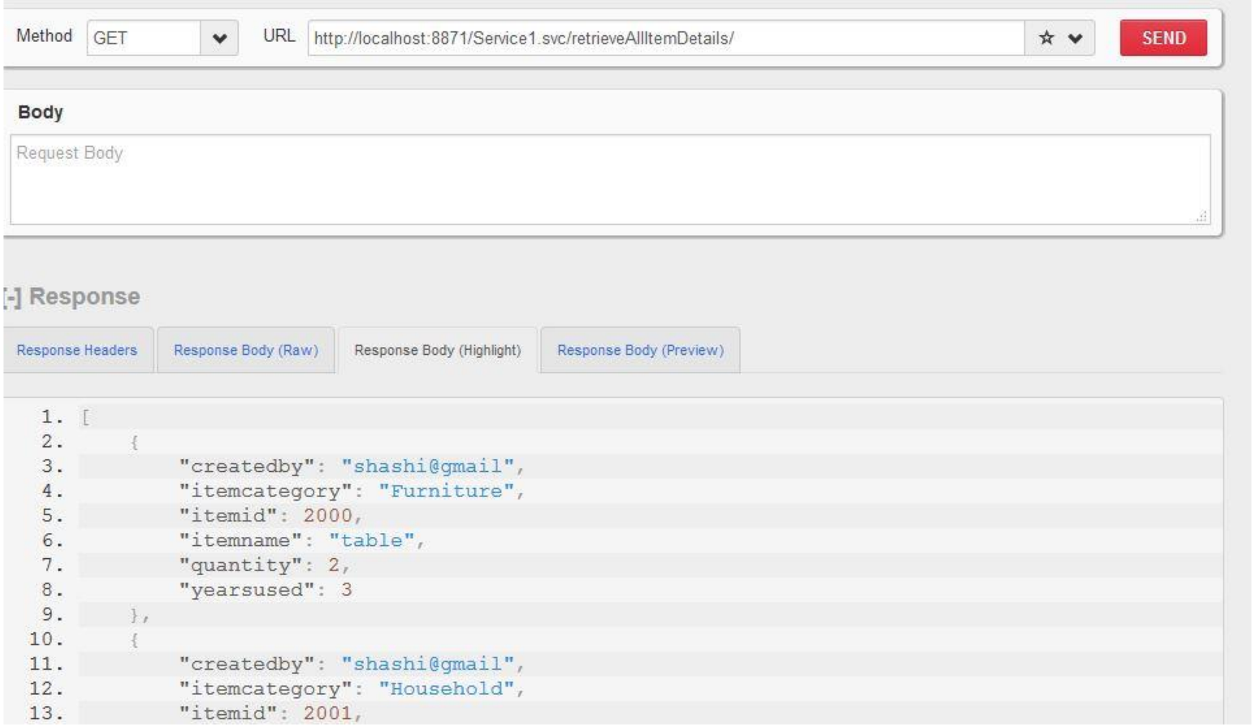
8871/Service1.svc/updateItemDetails/'2001','chair','Household','2','04-10-2015','shashi@gmail','2'

```
"Updated data_statementUpdate ITEMS set ITEMNAME='chair',ITEMCATEGORY='Household',QUANTITY='2',YEARSOFUSAGE='2' where ITEMID = '200
1'Updated data"
```

V. Item Retrieval Service:

This service is used to retrieve the list of items from the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and get the required details.

URL: http://10.0.2.2:8871/Service1.svc/retrieveAllItemDetails/



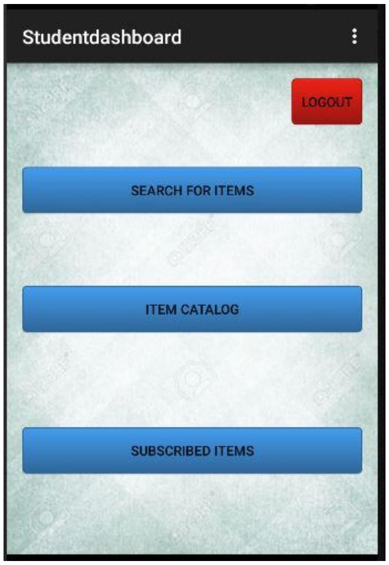
http://localhost:8871/Service1.svc/retrieveAllItemDetails/

```
[
    {
        "createdby": "shashi@gmail",
        "itemcategory": "Furniture",
        "itemid": 2000,
        "itemname": "table",
        "quantity": 2,
        "yearsused": 3
    },
    {
        "createdby": "shashi@gmail",
        "itemcategory": "Household",
        "itemid": 2001,
```

- **Implementation of User Interface:**

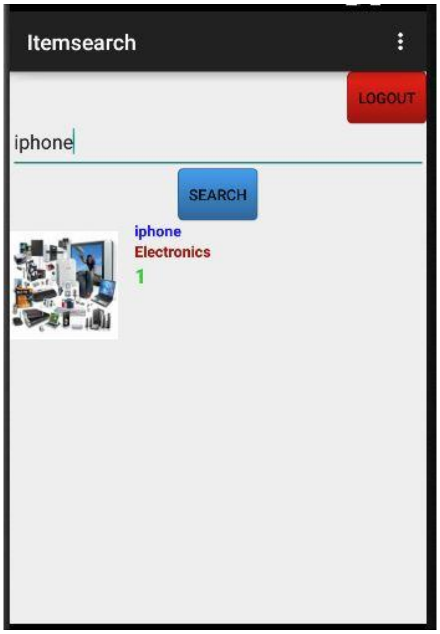
I. Student Dashboard Screen:

This is the dashboard screen to user which allows the user to search for items, lookup the item catalogue and view the items subscribed by him.



II. Search Items Screen:

This screen is used to search for an item by the student. He will be displayed with all the items that the donor is willing to donate.

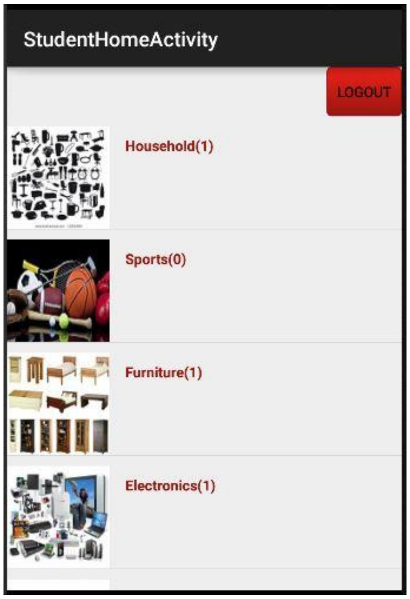


iphone

iphone
Electronics
1

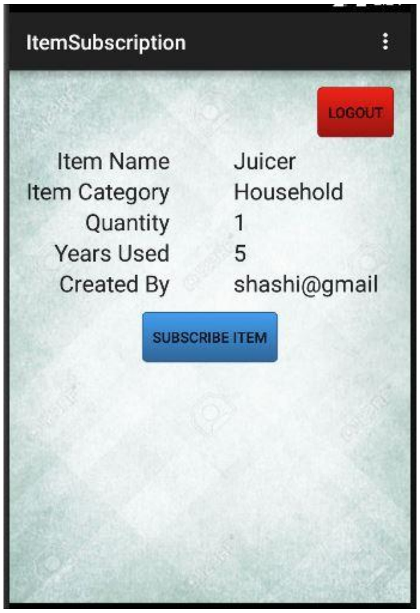
III. View Items Catalogue by Categories Screen:

This screen is used to select a list of items depending on the category which the user can select from a drop down list.

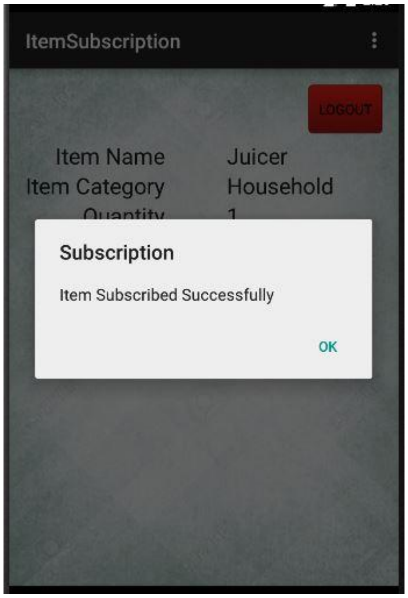


IV. Item Subscription Screen:

This screen is used to display the item details along with a button for subscription.



Item Name	Juicer
Item Category	Household
Quantity	1
Years Used	5
Created By	shashi@gmail



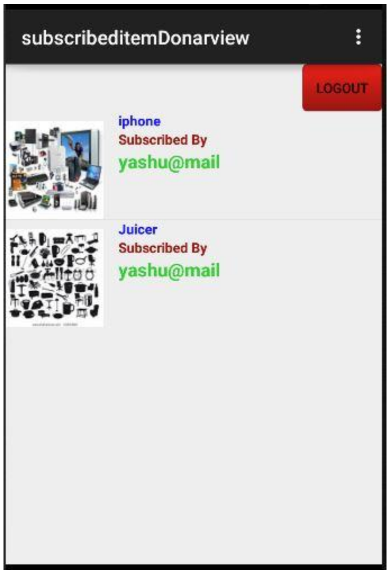
Item Name	Juicer
Item Category	Household
Quantity	1

Subscription

Item Subscribed Successfully

V. View Subscribed Items by Donor Screen:

This screen is used to display all the list of items that the donor has subscribed to.



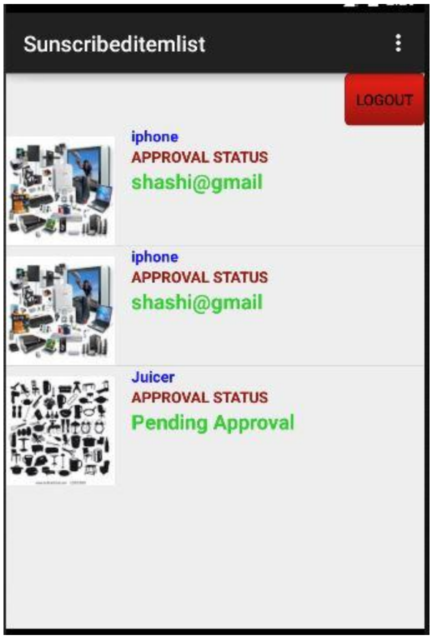
subscribeditemDonarview

iphone
Subscribed By
yashu@mail

Juicer
Subscribed By
yashu@mail

VI. Approved Items Screen:

This screen is used to view the list of items that are approved by the donor.



Sunscribeditemlist

iphone
APPROVAL STATUS
shashi@gmail

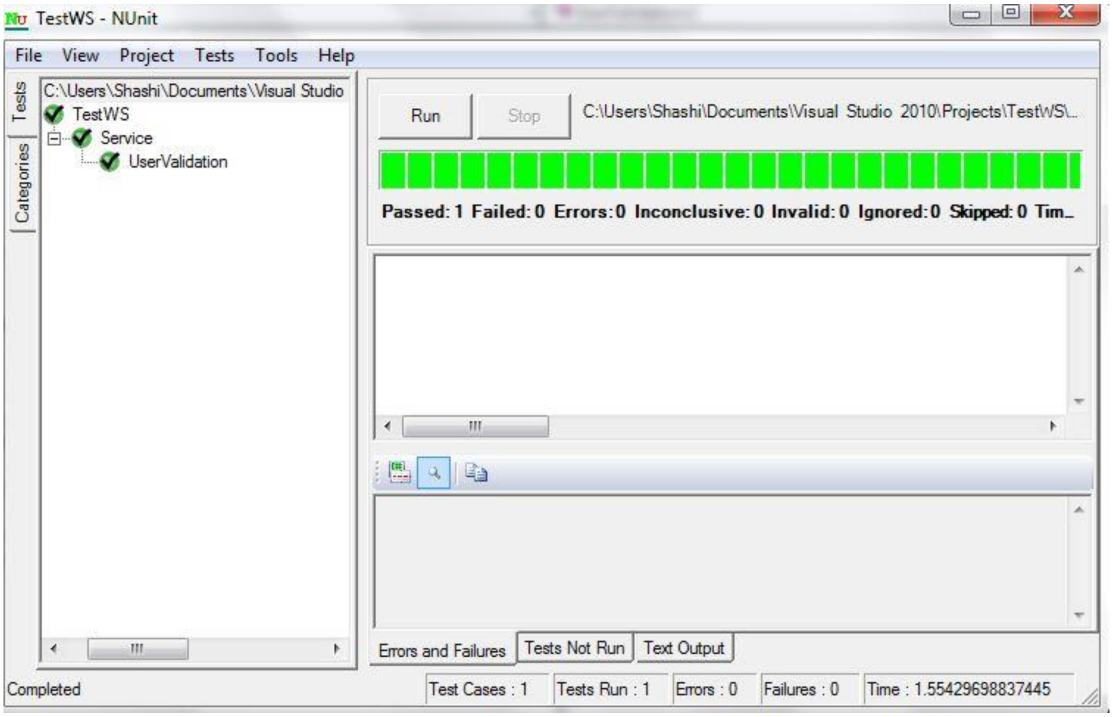
iphone
APPROVAL STATUS
shashi@gmail

Juicer
APPROVAL STATUS
Pending Approval

Testing:

NUnit Test Cases:

a. User Login:



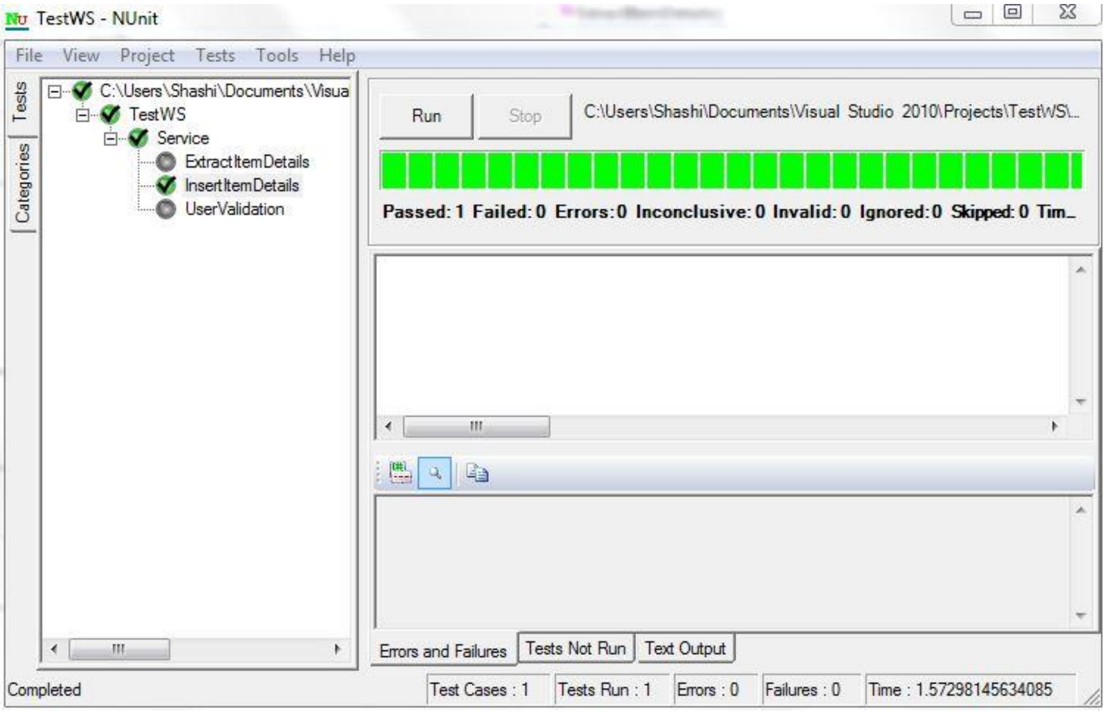
C:\Users\Shashi\Documents\Visual Studio
TestWS
Service
UserValidation

C:\Users\Shashi\Documents\Visual Studio 2010\Projects\TestWS\...

Passed: 1 Failed: 0 Errors: 0 Inconclusive: 0 Invalid: 0 Ignored: 0 Skipped: 0 Tim...

Completed | Test Cases : 1 | Tests Run : 1 | Errors : 0 | Failures : 0 | Time : 1.55429698837445

b. Item Addition:



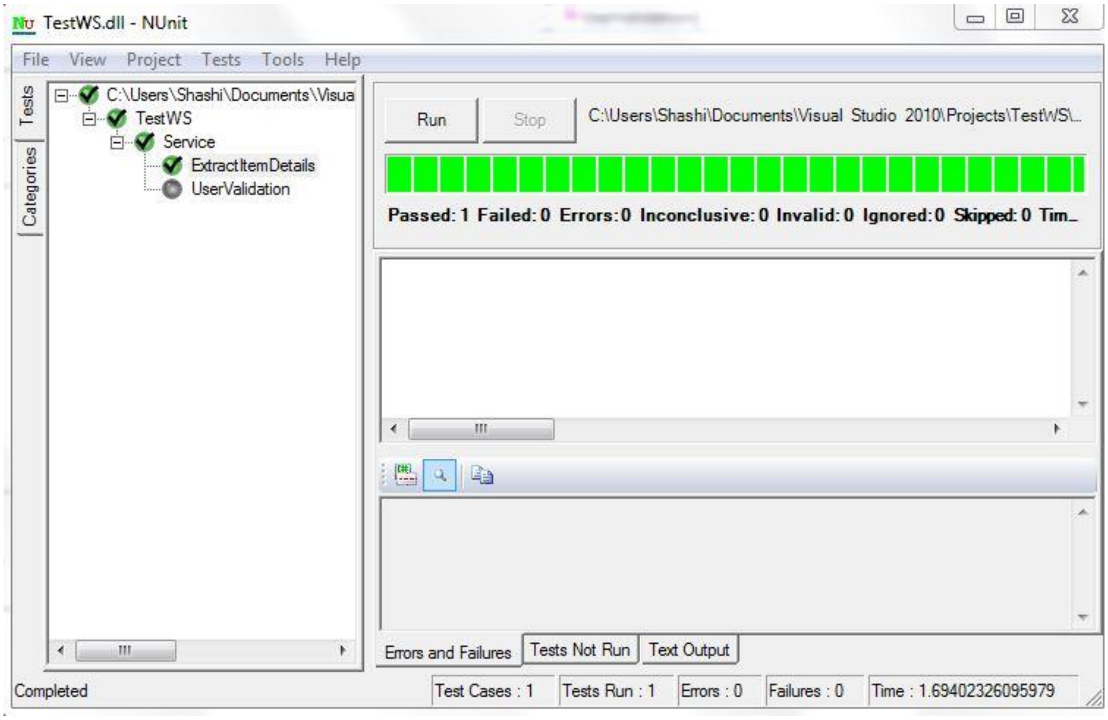
C:\Users\Shashi\Documents\Visua
TestWS
Service
ExtractItemDetails
InsertItemDetails
UserValidation

C:\Users\Shashi\Documents\Visual Studio 2010\Projects\TestWS\...

Passed: 1 Failed: 0 Errors: 0 Inconclusive: 0 Invalid: 0 Ignored: 0 Skipped: 0 Tim...

Completed | Test Cases : 1 | Tests Run : 1 | Errors : 0 | Failures : 0 | Time : 1.57298145634085

c. **Item List Retrieval:**



TestWS.dll - NUnit
File View Project Tests Tools Help
Tests
Categories
C:\Users\Shashi\Documents\Visua
TestWS
Service
ExtractItemDetails
UserValidation
Run
Stop
C:\Users\Shashi\Documents\Visual Studio 2010\Projects\TestWS\...
Errors and Failures
Tests Not Run
Text Output

Passed: 1 Failed: 0 Errors: 0 Inconclusive: 0 Invalid: 0 Ignored: 0 Skipped: 0 Tim...

Completed | Test Cases : 1 | Tests Run : 1 | Errors : 0 | Failures : 0 | Time : 1.69402326095979

Deployment:

- **SrumDo:**

https://www.scrumdo.com/projects/project/giveaway/summary

- **GitHub:**

https://github.com/sashi987/ASE/tree/master/Increment2

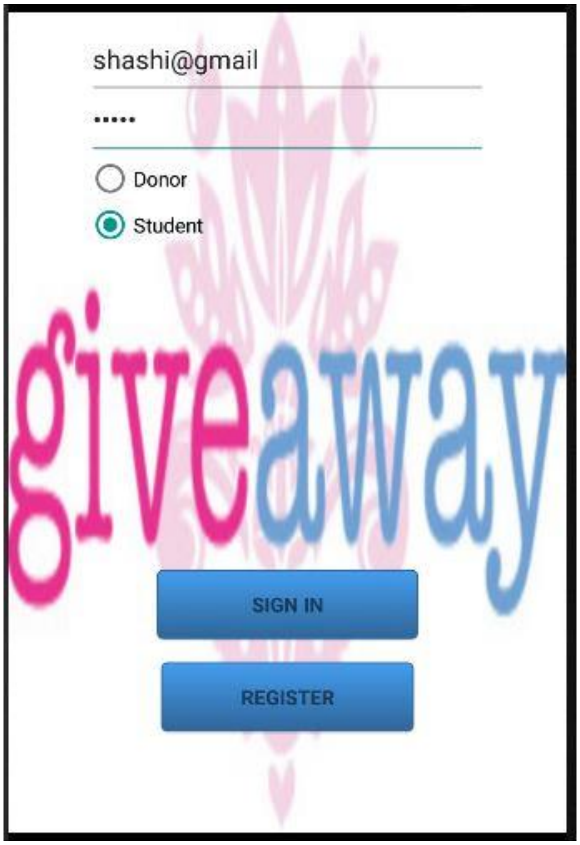
<u>Report:</u>

The Give Away application is being developed as an Android application using Android SDK framework and SQL Server Database for data.

The following are the screens that we are using as part of our project,

I. Login Screen:

The user after the successful registration has to login to the application by using his/her credentials. He has to select one of the options between the donor and the student.

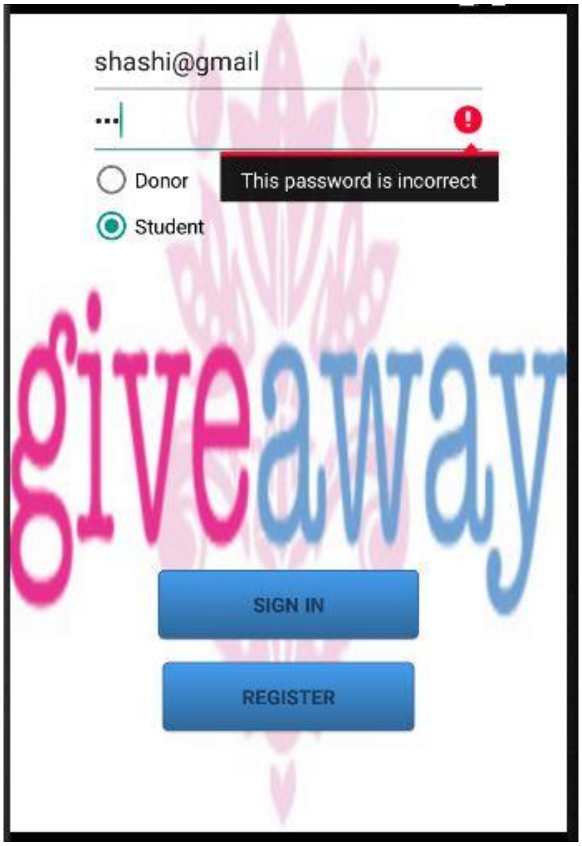


shashi@gmail

•••••

○ Donor

● Student



shashi@gmail

•••

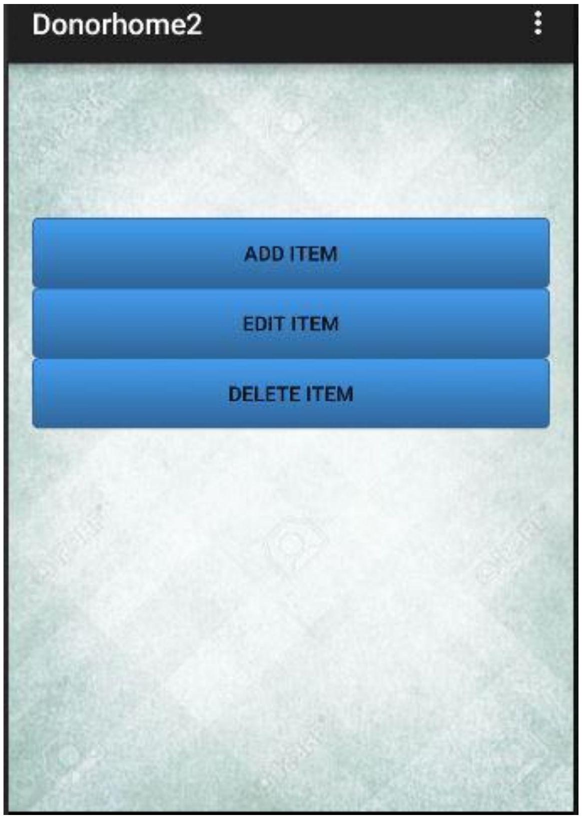
This password is incorrect

○ Donor

● Student

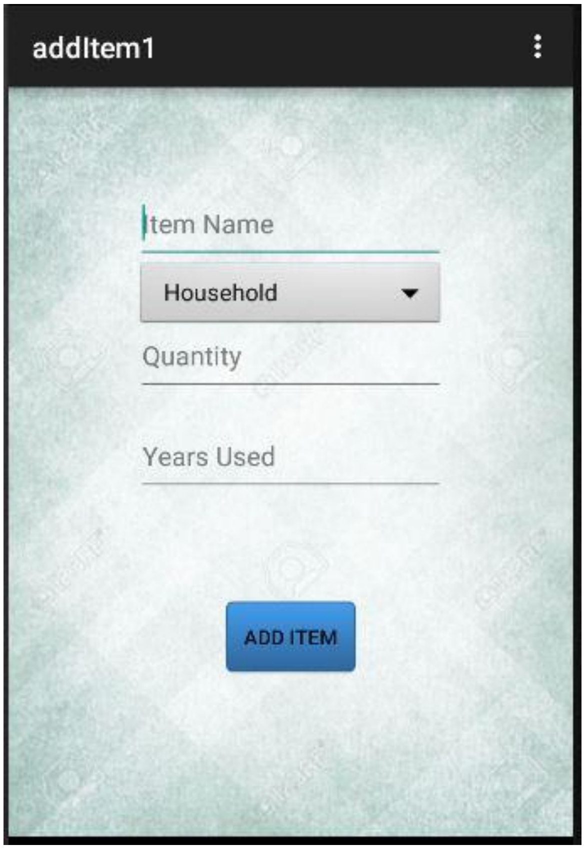
II. Donor Home Screen:

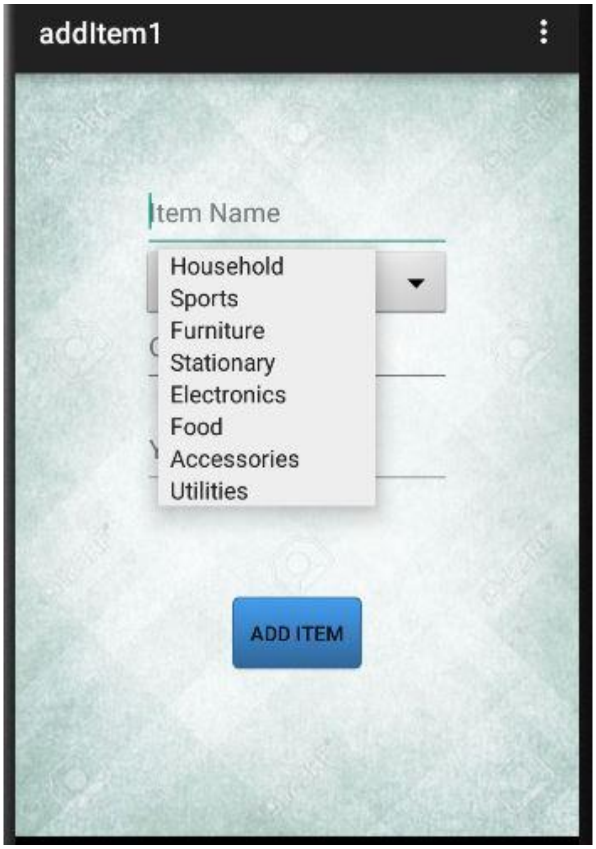
This screen contains the tabs for selecting the next operation. It has Add Item, Edit Item and Delete Item Tabs.



III. Item Addition Screen:

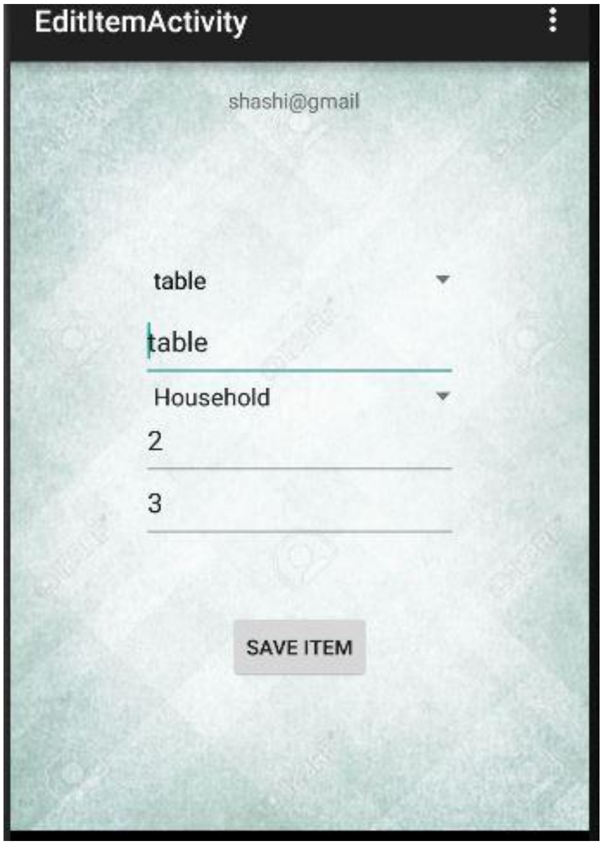
The user uses this screen for adding an item for Giveaway. He has to give the details like Item Name, Item Category, Quantity and the number of years used.





IV. Item Updating Screen:

If the user wants to update the information of the items which he/she has posted, he can use this screen.



EditItemActivity

shashi@gmail

table

table

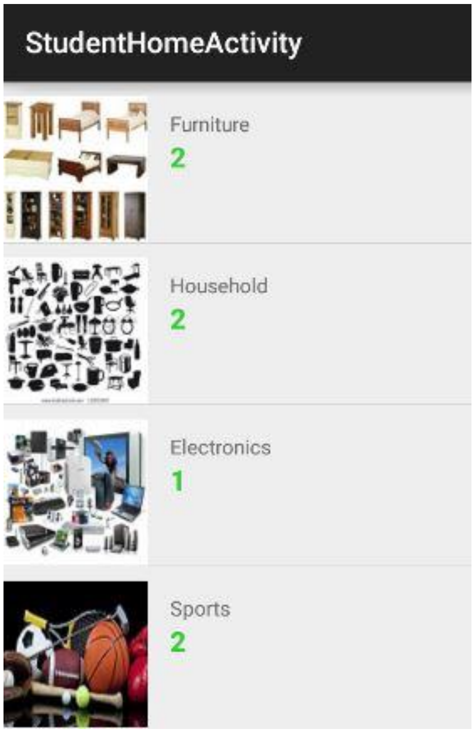
Household

2

3

V. Student Home Screen:

After the successful login, the student will be redirected to this screen, where the list of all items will be populated.



StudentHomeActivity

Furniture
2

Household
2

Electronics
1

Sports
2

Project Management:

Work completed:

- Description:
  a. Registration and Login Services for Users
  b. Adding/Updating/Deleting an Item
  c. Retrieval of the list of Items

- Responsibility (Task, Person):

  a. **Sashidhar Reddy Gowra**: Creation of Services for User Login, User Registration, Item Addition, Item Updating and Item retrieval.

  b. **Venkataramana Yashwant Kumar Palisetty**: Creation of UI Screens for User Login and Registration and the corresponding validations.

  c. **Ravikanth Devanaboyina**: Creation of UI Screen for Displaying the Item list.

  d. **Anudeep Reddy Gujjula**: Creation of UI Screens for Item Addition, Item Updating and Item Deletion.

- Time taken (#hours):
  UI Design, Service Creation and Client Validations: 80 hrs.

- Contributions (members/percentage):
  a. Sashidhar Reddy Gowra - 25%
  b. Venkataramana Yashwant Kumar Palisetty - 25%
  c. Ravikanth Devanaboyina - 25%
  d. Anudeep Reddy Gujjula - 25%.

Work to be completed:

- Description: Using the already existing Google maps API for sharing the address location of a user, Unit Testing, System Testing, Testing the application on Android Devices
- Responsibility (Task, Person):
  Creating Web services, Hash Functions: Sashidhar Reddy Gowra, Ravikanth Devanaboyina.
  Populating screens, UI alignments: Venkataramana Yashwant Kumar Palisetty, Anudeep Reddy Gujjula.
- Time to be taken (estimated #hours) 100 hrs.

Issues/Concerns: Alignments and data insertion/while inserting the email id of the user.

Final Project Evaluation

As part of the completion of this project we would like to evaluate our system. The main aim of this project is to provide a platform for the students who are migrating to new places in arranging the household items from the ones who are willing to donate the items.

We have started with the UML diagrams of the project that laid a stepping stone for the success of this project. It made us come to a conclusion on the architectural and system requirements. During the entire phase of the project we have followed agile methodology and used the agile tool Scrum do for managing the work contributed by the team members. It helped us a lot in creation of the stories for individuals and time requirements for the individual story etc.

Meetings held at regular intervals helped us in keeping a track of the work that every individual is doing as per the iterations specified in the scrum do. Design of the user interface has been very challenging in these type of applications as the application needs to handle a huge amount of data load which yet times results in the crashing of the application. Using google api's, creation of web services and integrating them has been a challenging task throughout the implementation of this application. Credits to the entire team for making the deliverables ready on time.

Video File:

https://youtu.be/EBJBI0gHe6A