GIVE AWAY

FOURTH INCREMENT REPORT

Submitted by:

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

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.

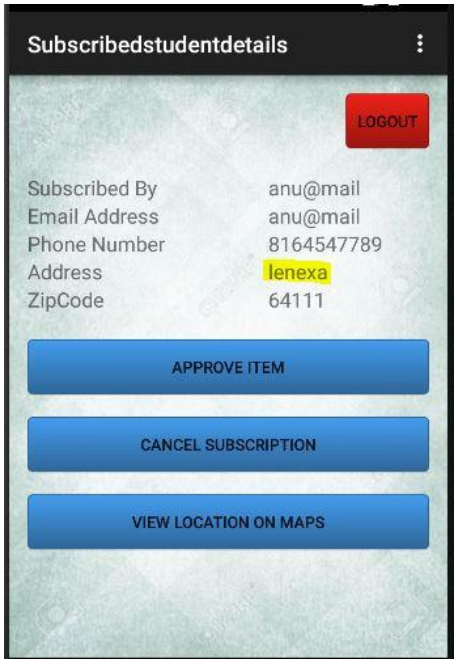
Import Existing API or Services:

Google Maps API:

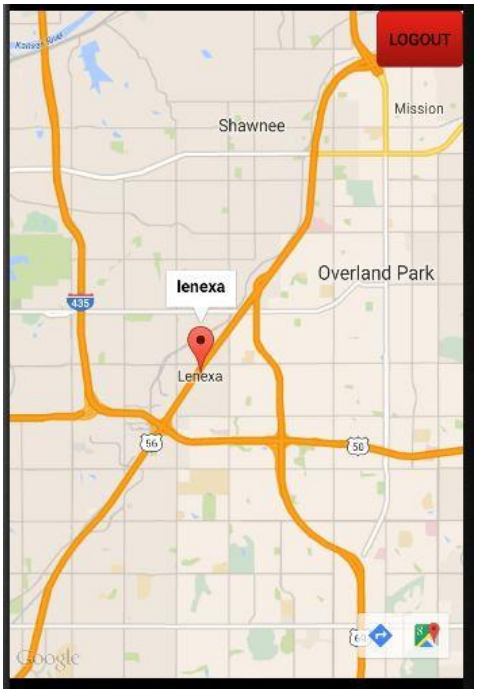
We have implemented Google maps API for locating the location of user who wish to grab items from donor. This service can be used on android mobile to locate the user and to get the driving directions to the student's destination which will help donor in delivering the products.

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

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

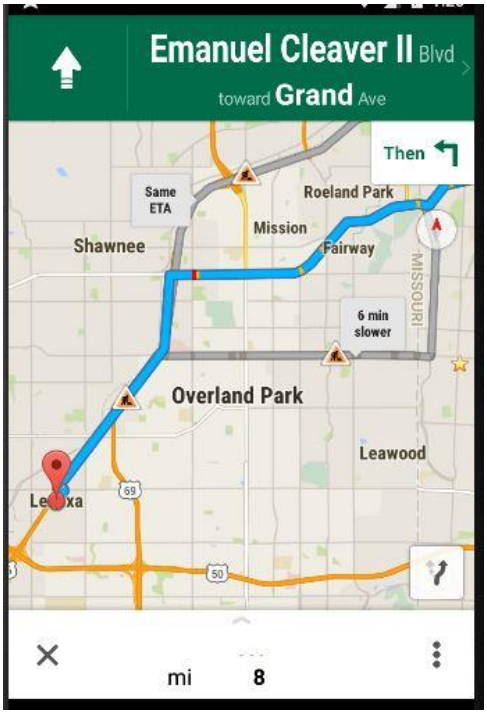


Google Maps Screen: Here student location is identified on the Google maps and is pointed with a marker.



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

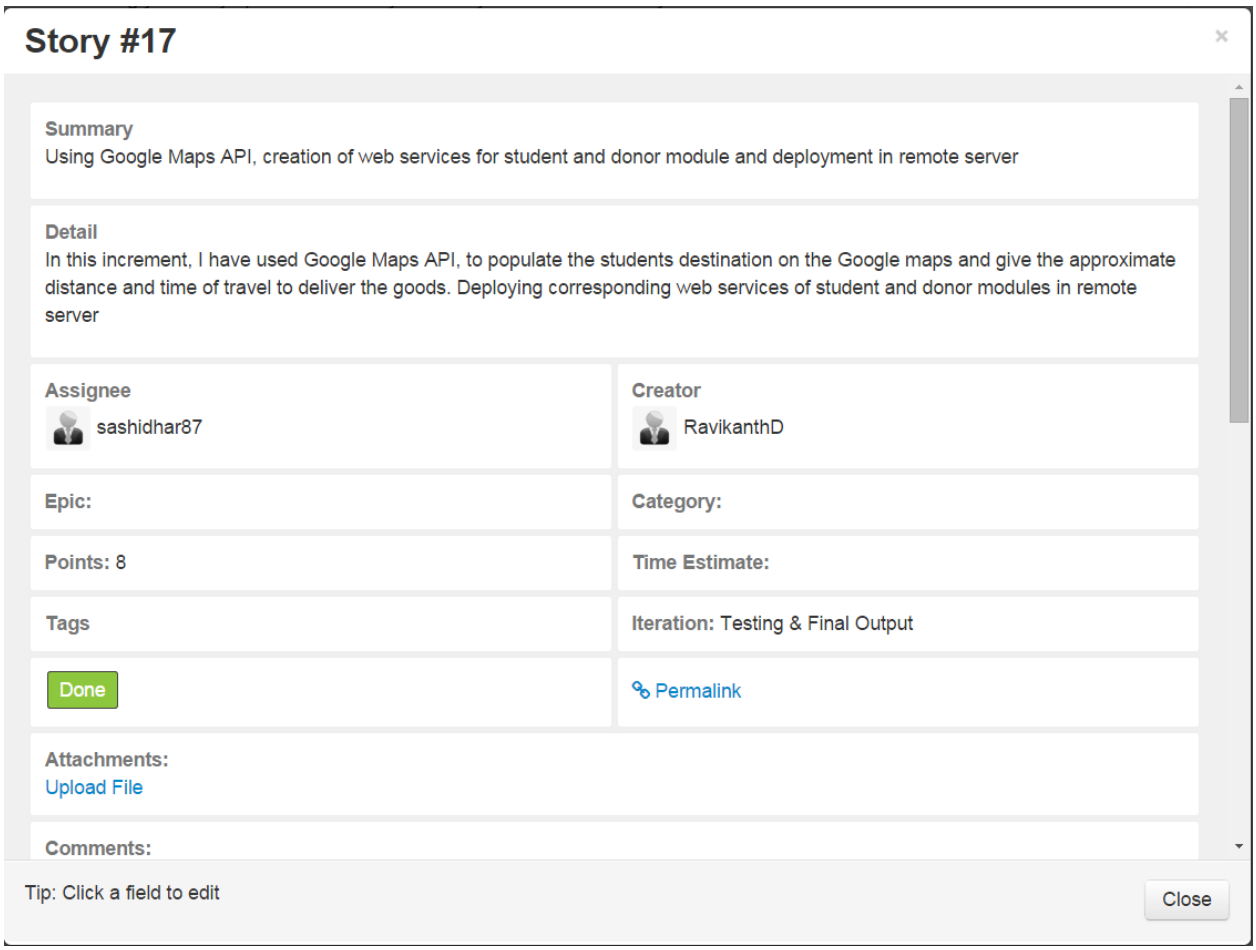
Driving directions to Students location:

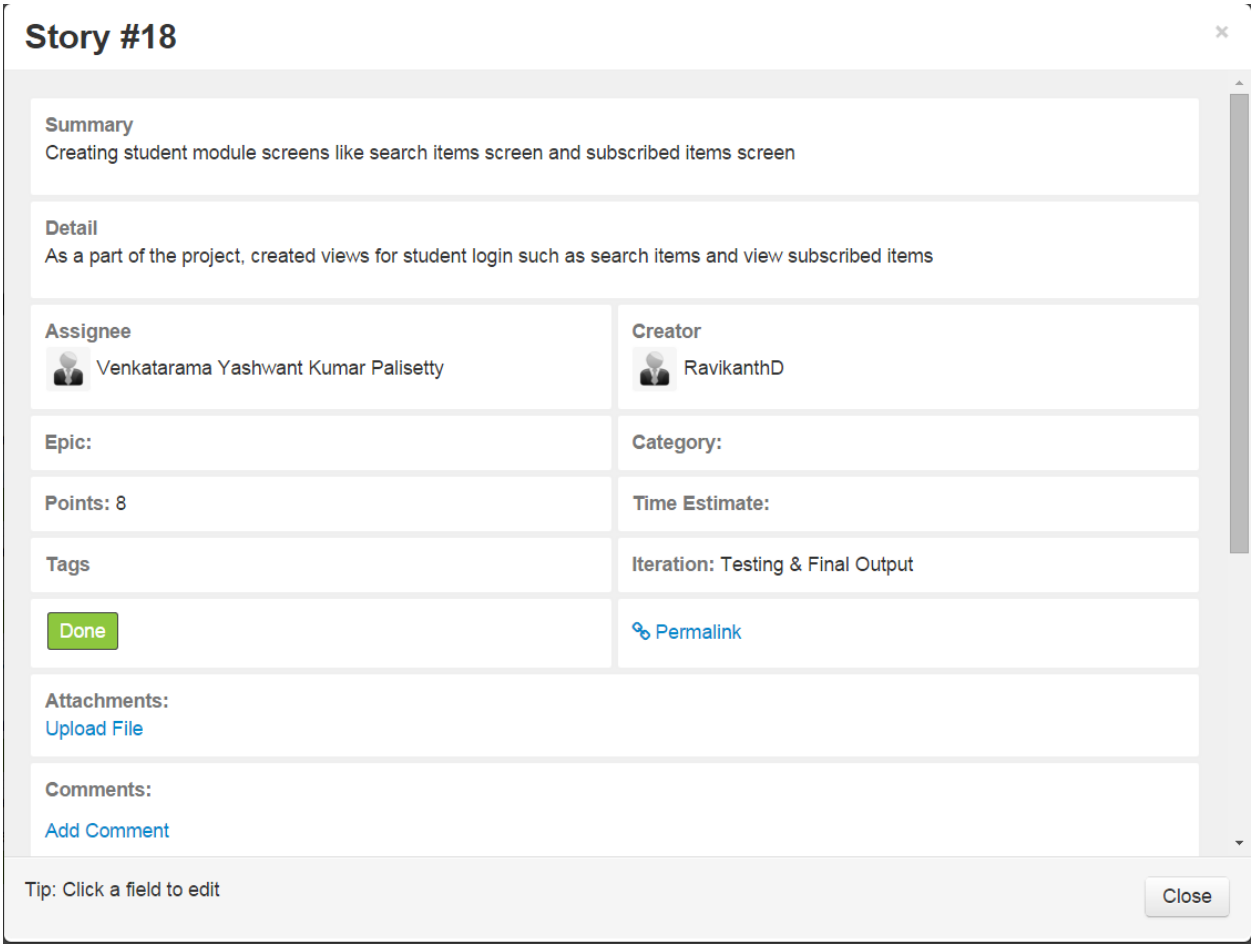


Detail Design of Services:

- **User stories using ScrumDo:**

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




Story #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
Done
Permalink
Attachments:
Upload File
Comments:
Add Comment
Tip: Click a field to edit
Close

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

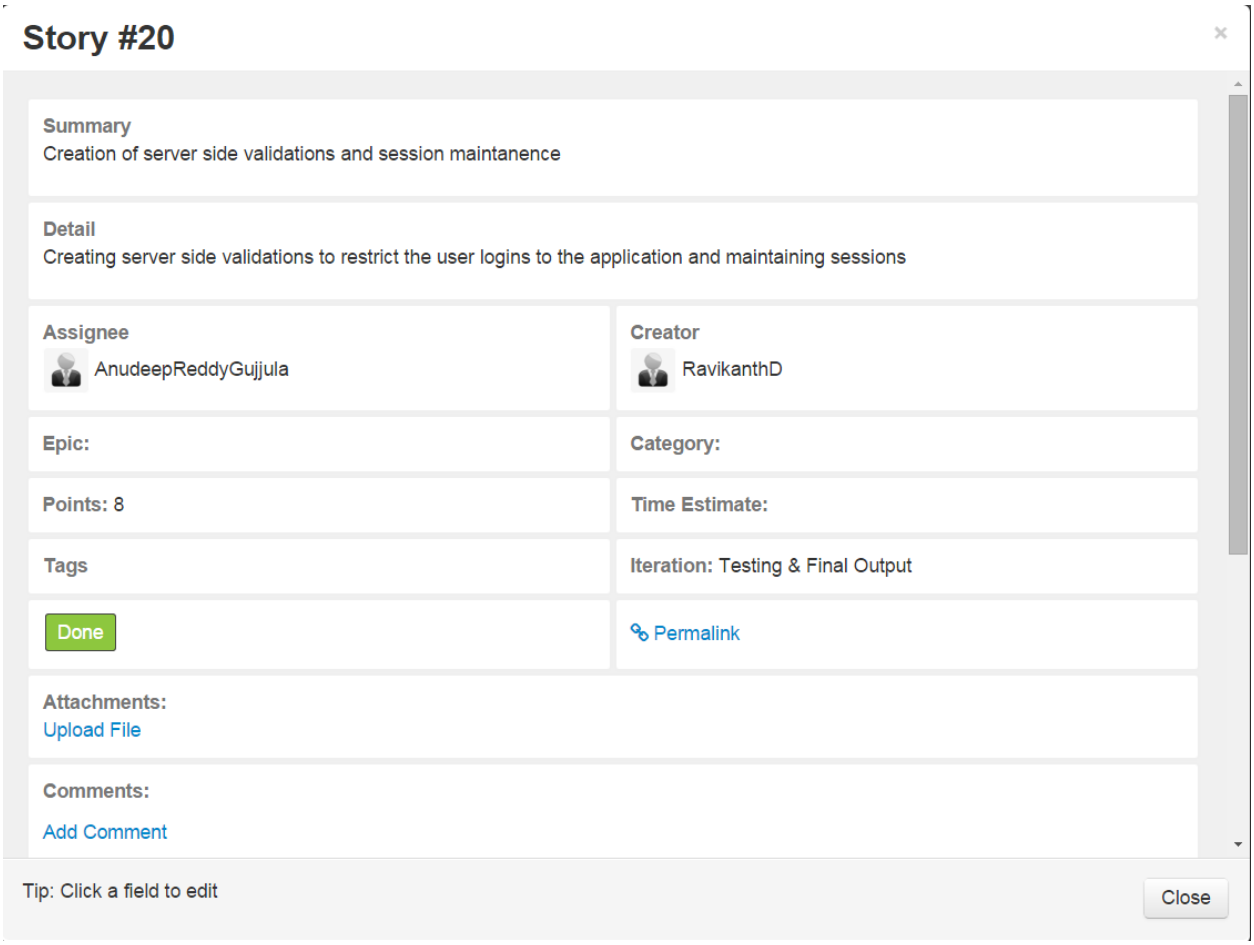
Assignee RavikanthD	**Creator** RavikanthD
Epic:	**Category:**
Points: 8	**Time Estimate:**
Tags	**Iteration:** Testing & Final Output
Done	Permalink

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.

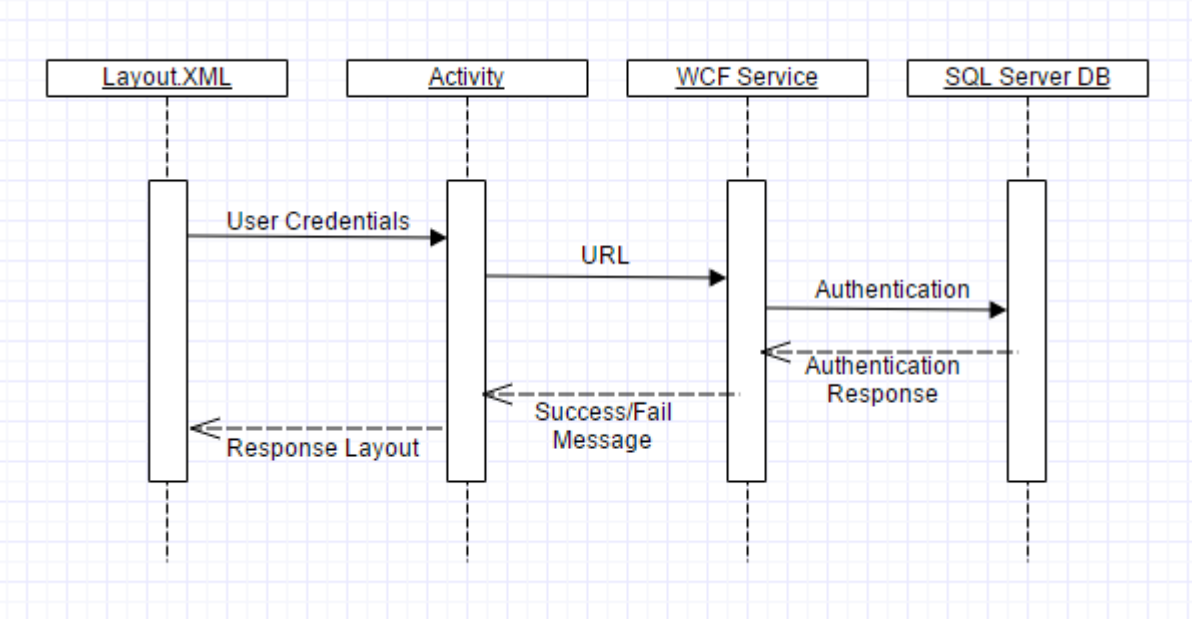
VI. **Search for item Service:** This service is used to search for an item and populating them on the corresponding screen.

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.

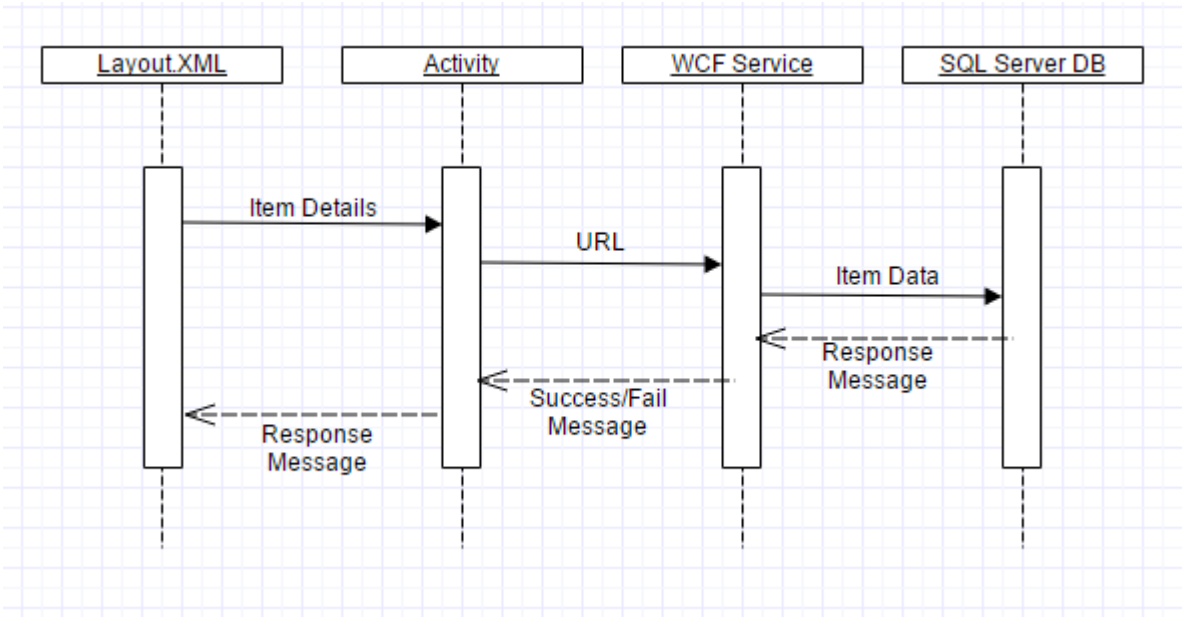
- **Architecture Diagram:**

- **Sequence Diagram:**

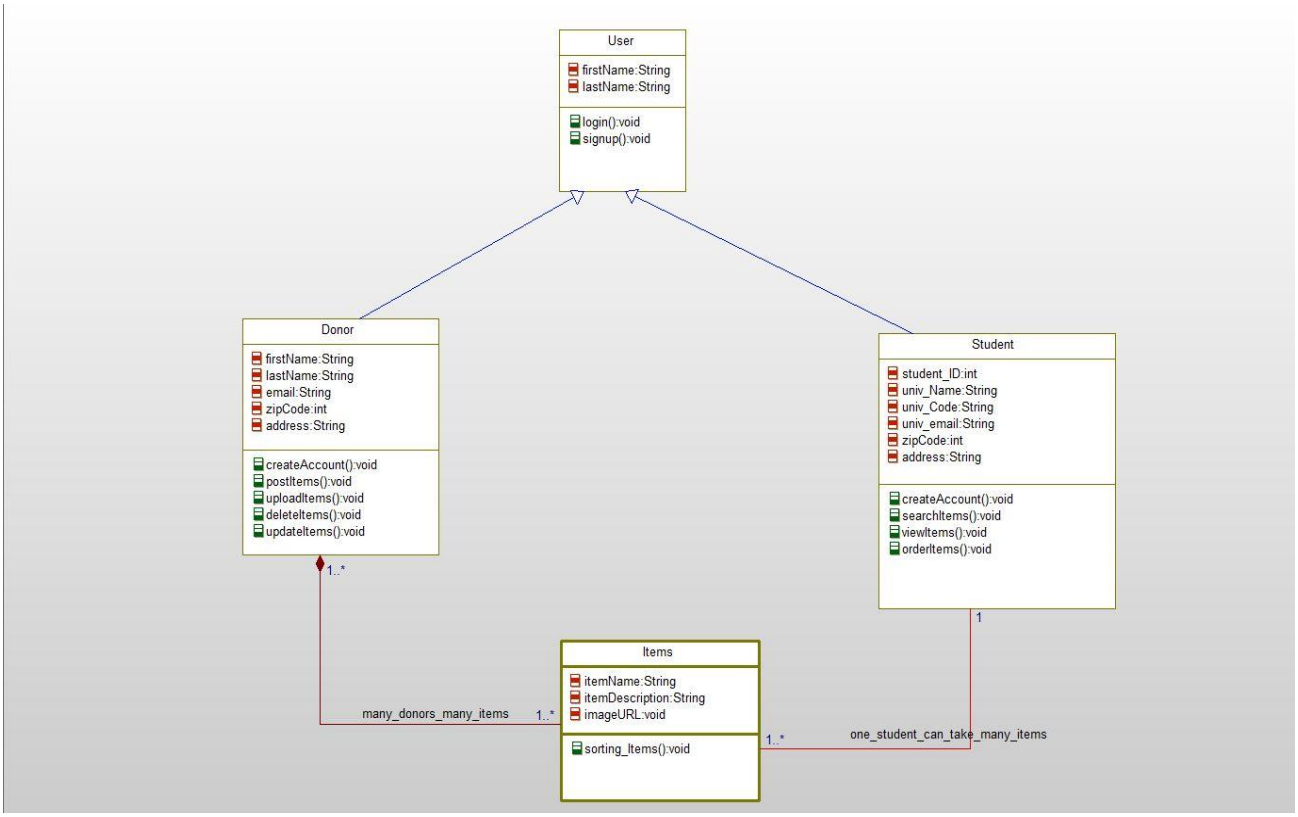
I. **Sequence Diagram for User Login/Registration:**


Layout.XML
Activity
WCF Service
SQL Server DB
User Credentials
URL
Authentication
Authentication Response
Success/Fail Message
Response Layout

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	**Pass**

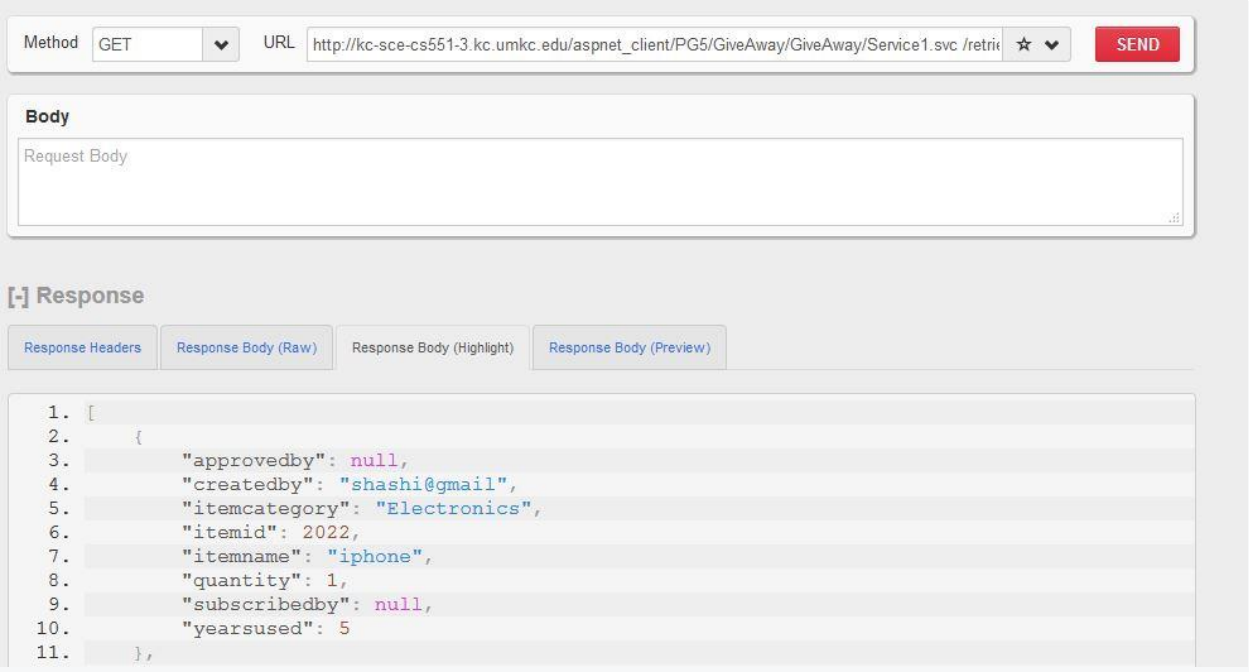
			characters are not accepted.	
10	Adding Items	Enter valid details.	System should allow the user to add the item.	**Pass**

Implementation:

- **Implementation of Rest Services:**
  - **I. Item Search 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.

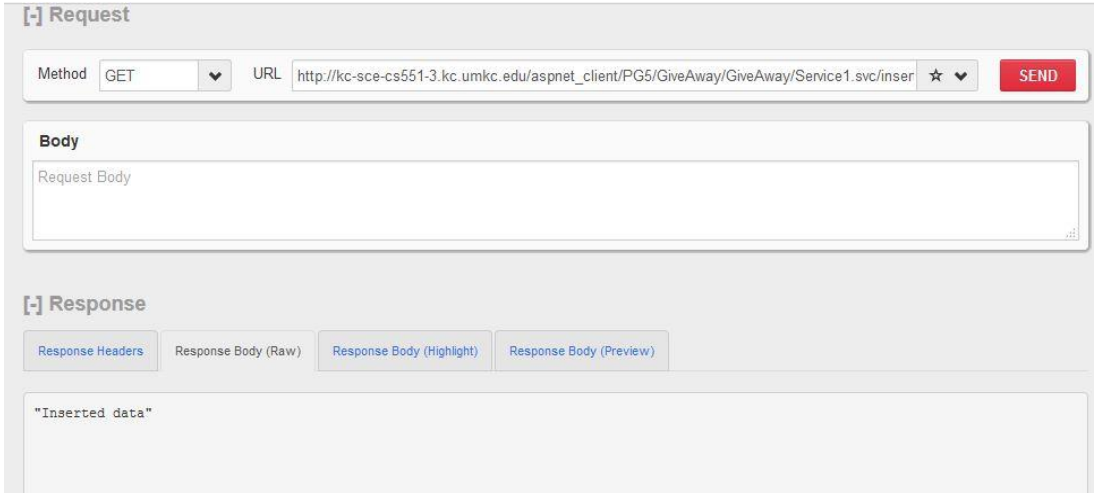
URL: http://kc-sce-cs551-3.kc.umkc.edu/aspnet_client/PG5/GiveAway/GiveAway/Service1.svc/retrieveAllItemDetails



  - **II. 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.

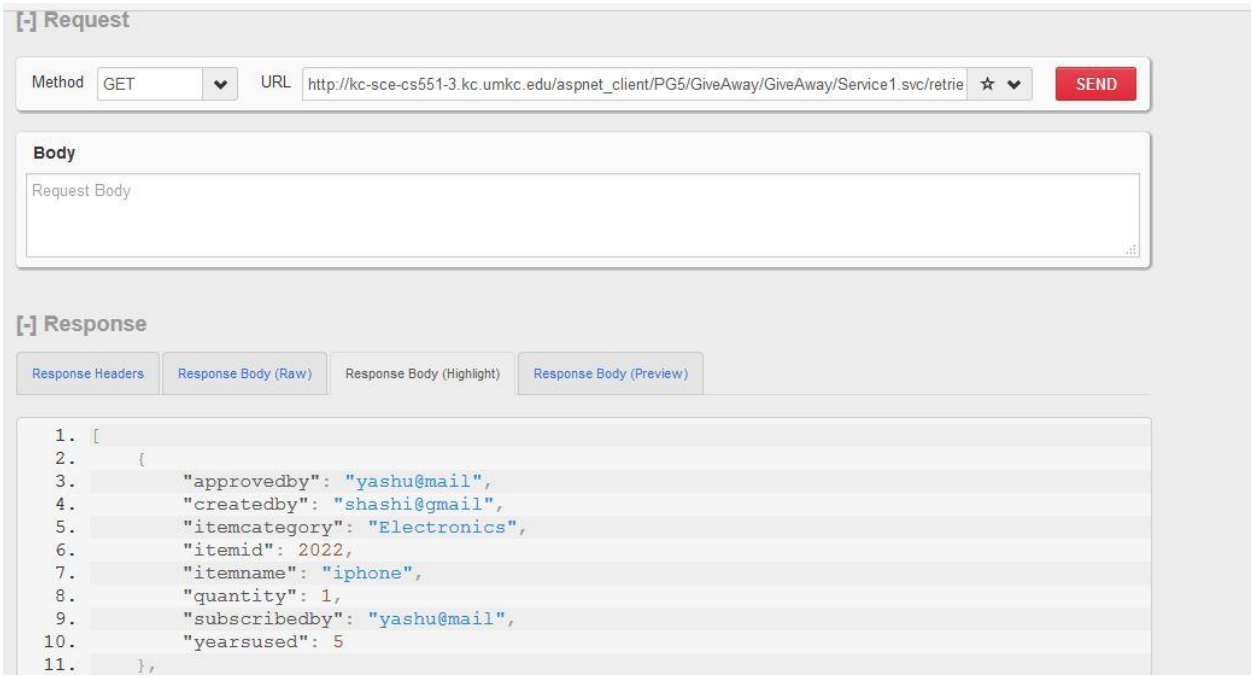
    URL: http://kc-sce-cs551-3.kc.umkc.edu/aspnet_client/PG5/GiveAway/GiveAwayService1.svc/insertItemDetails/'chair','Household','2','04-10-2015','shashi@gmail','4'



III. Subscribed ItemList Service:

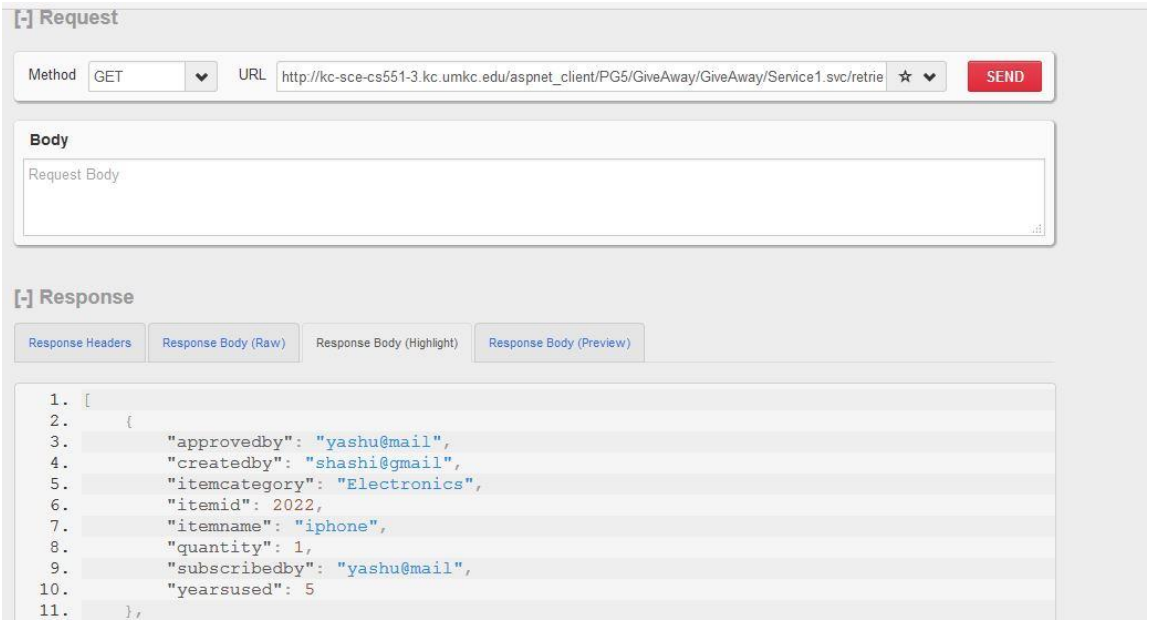
This service is used to retrieve the list of items from the Database which are subscribed by student, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and get the required details.

URL: http://kc-sce-cs551-3.kc.umkc.edu/aspnet_client/PG5/GiveAway/GiveAway/Service1.svc/ retrieveItemsbySubsciption/yashu@mail.umkc.edu



IV.**Subscribed Items in Donor view**: This service is used to retrieve the list of items from the Database which are subscribed by student, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and get the required details.

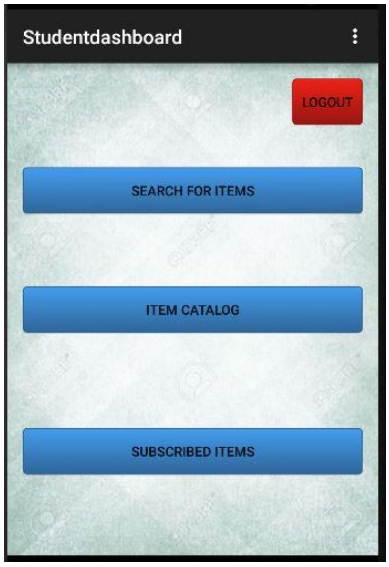
URL: http://kc-sce-cs551-3.kc.umkc.edu/aspnet_client/PG5/GiveAway/GiveAway/Service1.svc/ extractItemDetails/shashi@gmail.com



- **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:

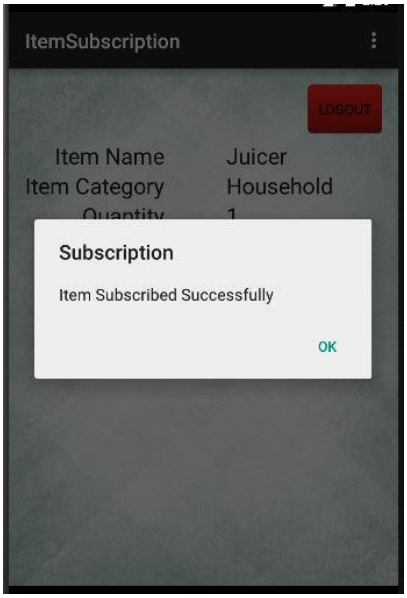
This screen is used to search for an item by the student. He will be displayed with all the items that the donor is willing to donate.

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.



V. View Subscribed Items by Donor Screen:

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

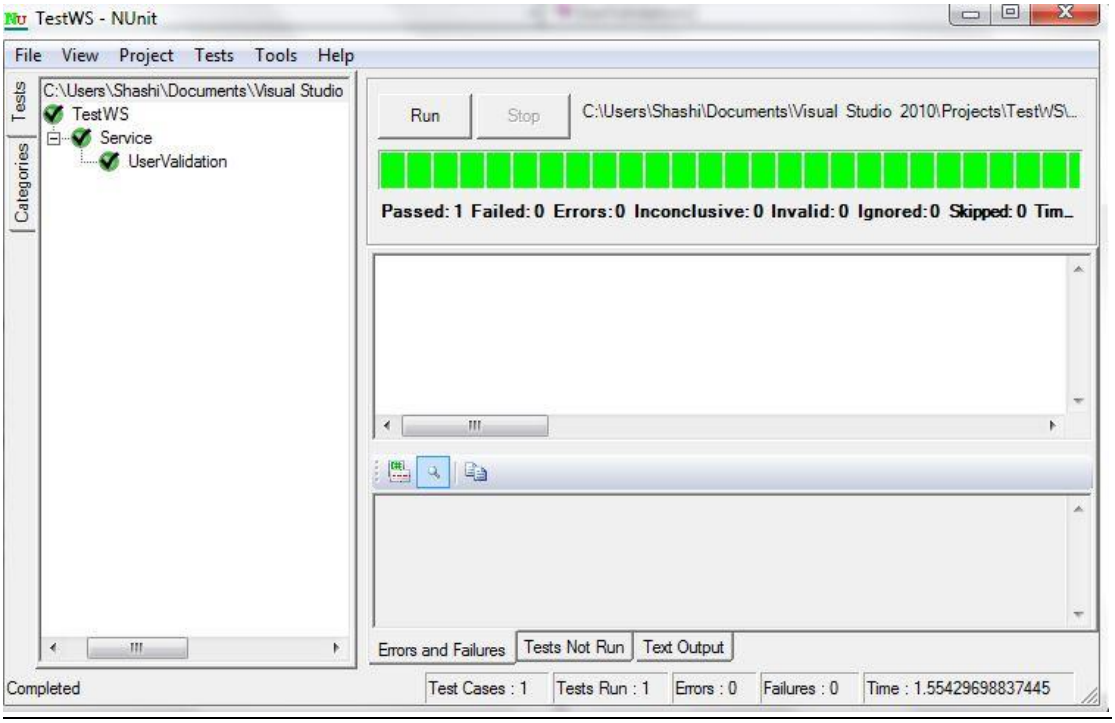
VI. Approved Items Screen:

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

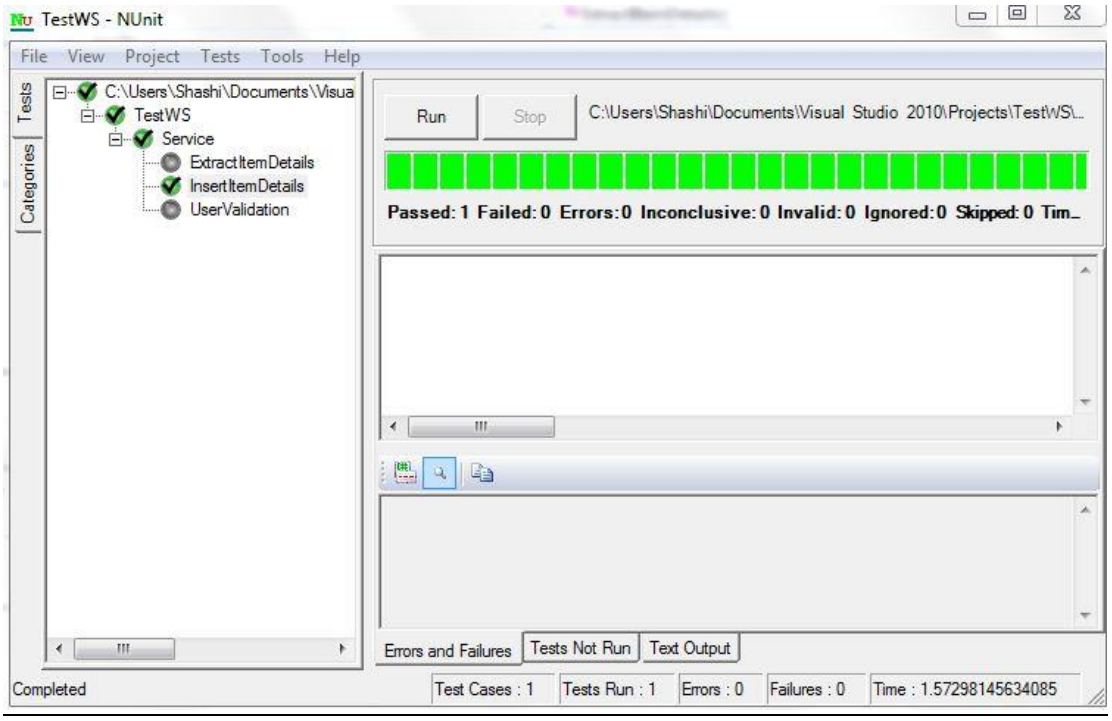
Testing:

NUnit Test Cases:

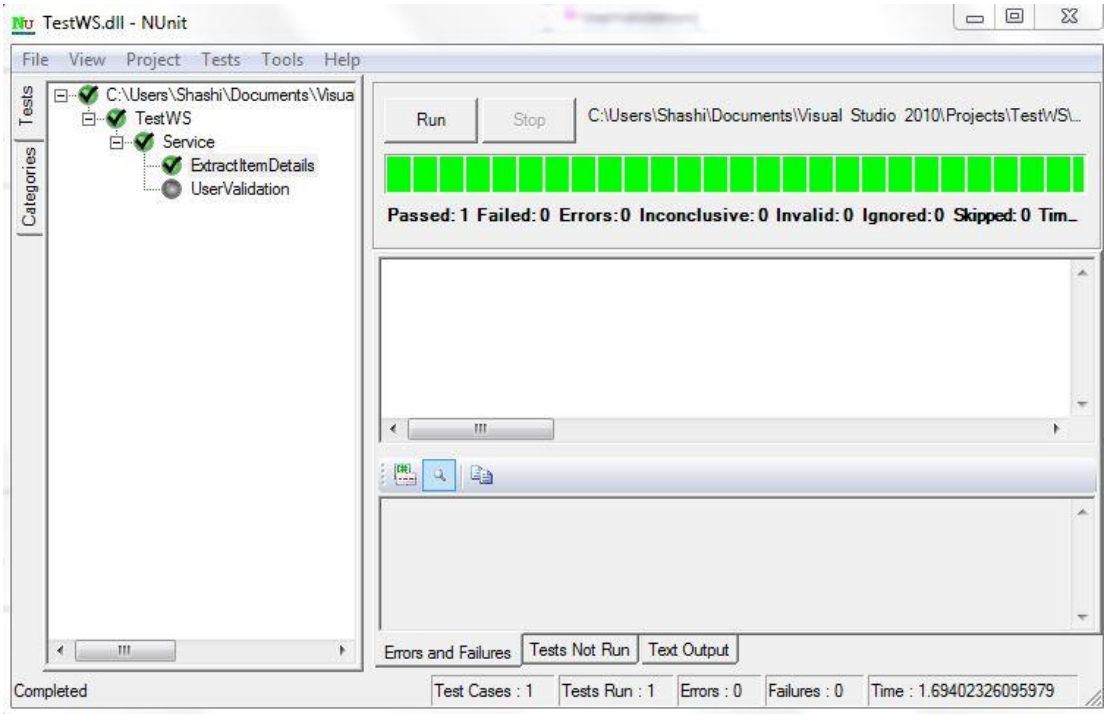
a. User Login:



b. Item Addition:



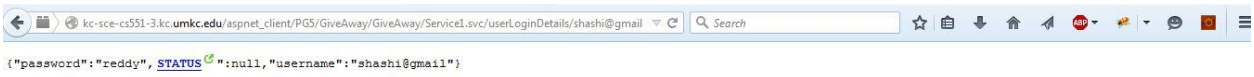
c. Item List Retrieval:

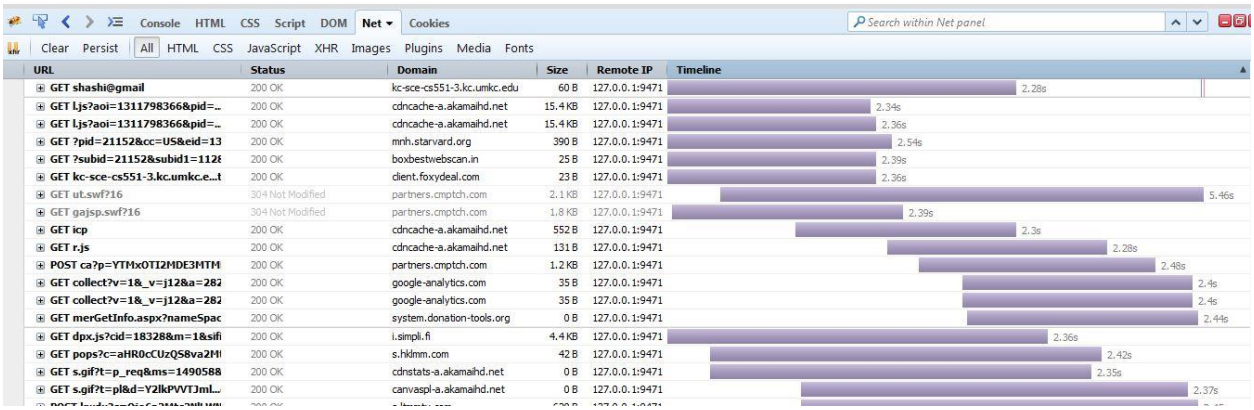


Deployment Testing:

FireBug:

Firebug is a free and open-source web browser extension for Mozilla Firefox that facilitates the live debugging, editing, and monitoring of any website's CSS, HTML, DOM, XHR, and JavaScript.Firebug is licensed under the BSD license and was initially written in January 2006 by Joe Hewitt, one of the original Firefox creators. The Firebug Working Group oversees the open source development and extension of Firebug. It has two major implementations: an extension for Mozilla Firefox and a bookmarklet implementation called Firebug Lite which can be used with Google Chrome.



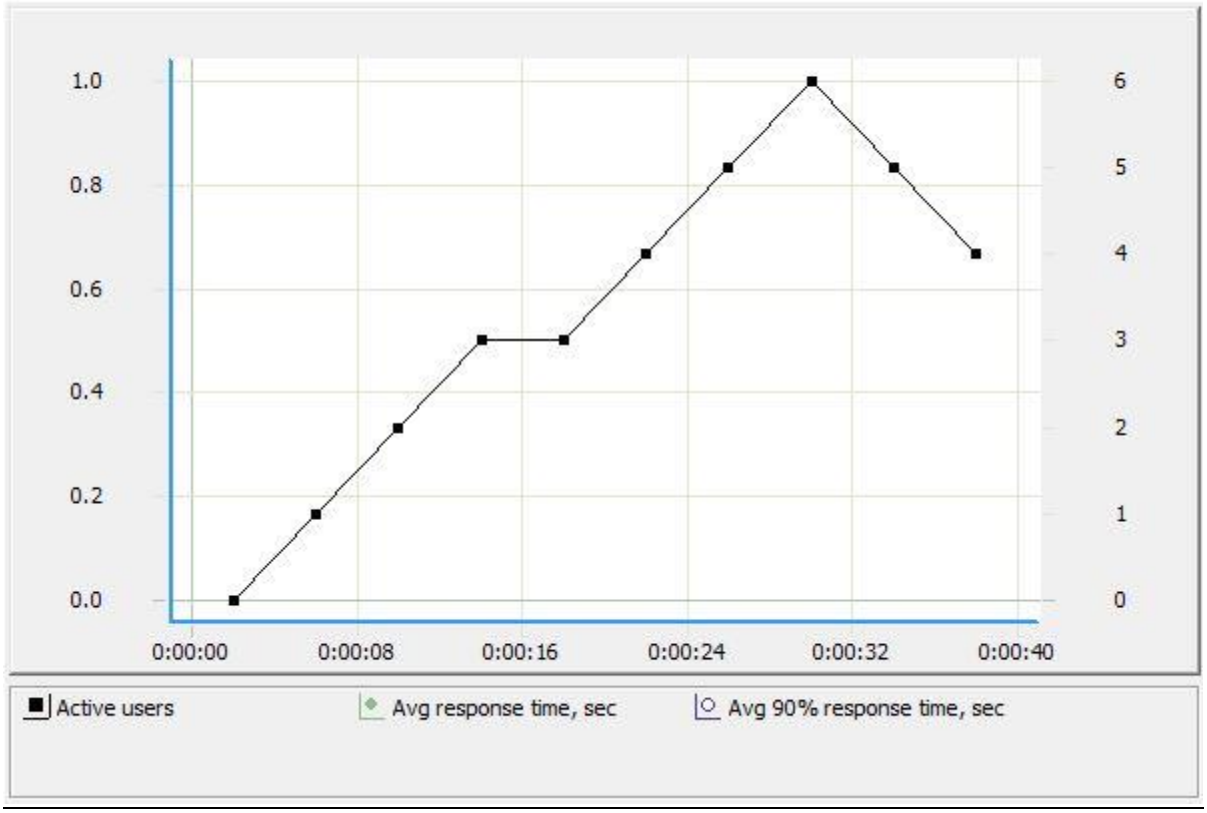


Wapt Tool:

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/summary

- **GitHub:**

https://github.com/sashi987/ASE/tree/master/Increment4

Report:

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

The following is the flow of our project,

Login as Donor:

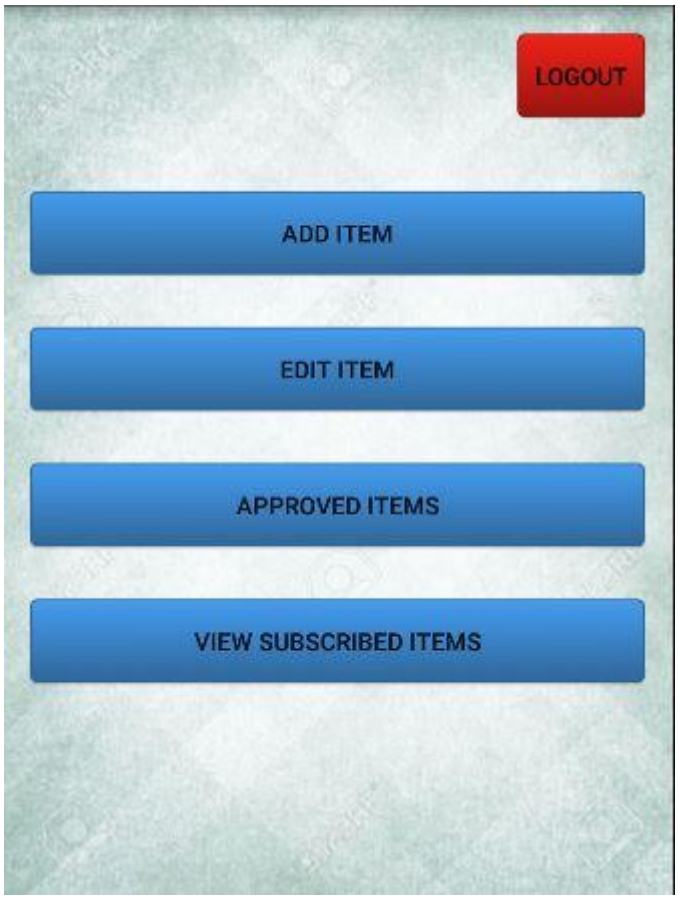
I. **Register Screen:**
In our project, there are two users namely a donor and a student. Initially, the donor who is willing to donate a product will register into the system using the register screen as follows,

II. Login Screen as Donor:

When the user registers as Donor, he can login to the system as donor.

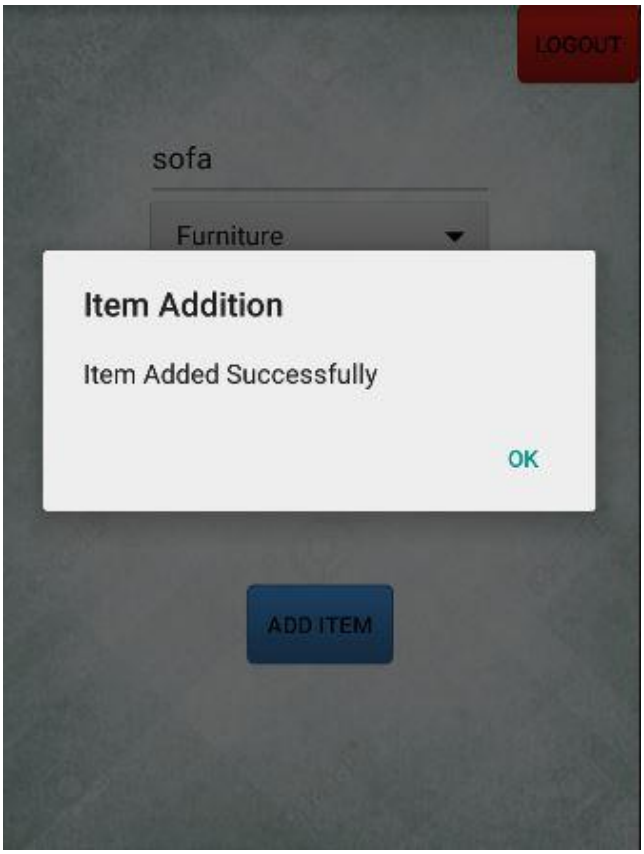
III. Dashboard of the Donor:

After the donor gets logged in, he will be redirected to the dashboard screen, from where he will be having various options to add an item, edit an item, approve an item and viewing the list of items that are subscribed by the Student.



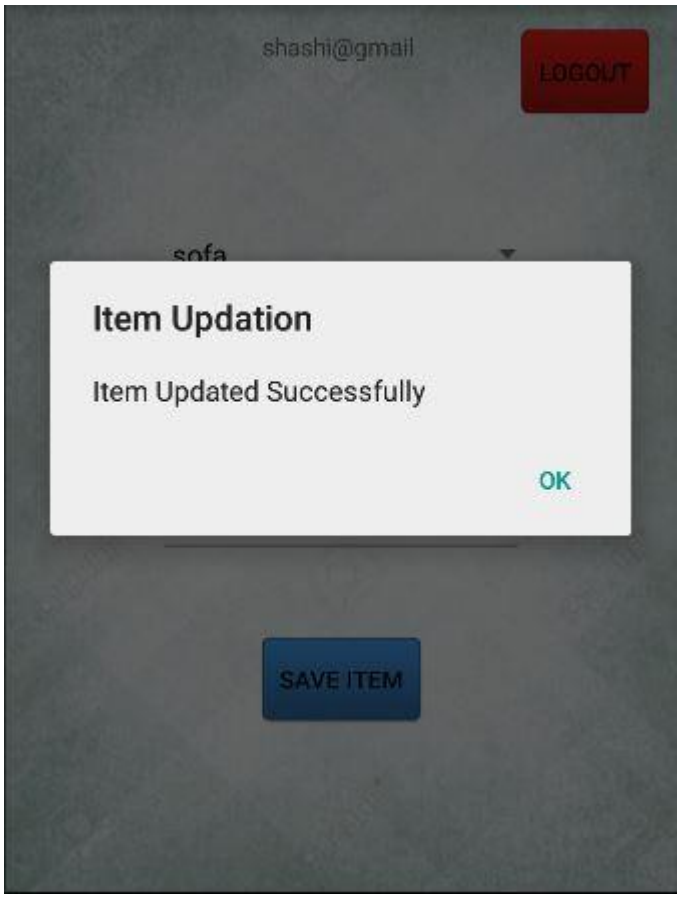
IV. Add Items Screen:

On click of Add Item button in Dashboard, donor will be directed here where he can add the item.



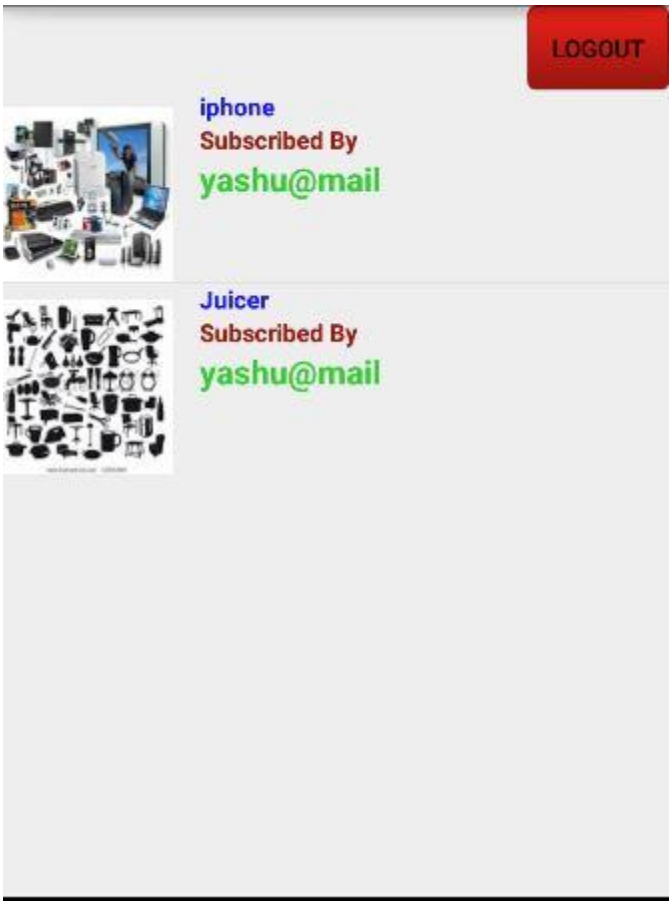
V. Edit Item Screen:

On click of Edit Item button in Dashboard, donor will be directed to this screen where he can edit the details of the item he has added.



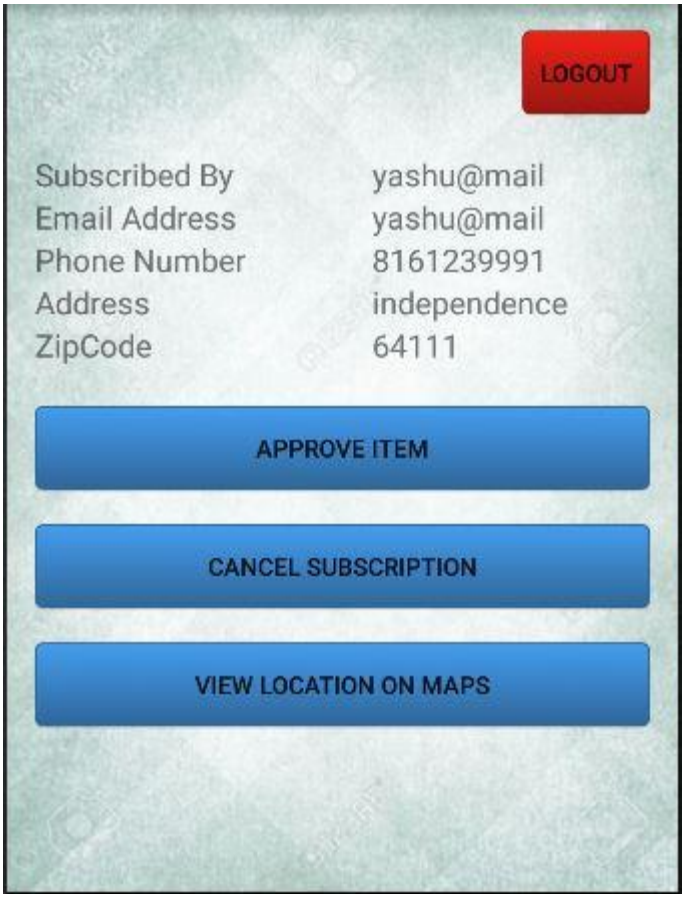
VI. View Subscribed Items Screen:

On click of View Subscribed Items button in Dashboard, donor will be directed to this screen where the donor can have a view of all the items that he has donated that are subscribed by the students.



VII. Subscribed Student Details Screen:

On click of an item in the View Subscribed Items Screen, donor will be directed to this screen where the donor can approve the item/cancel subscription for the student/ view the location on maps for the user.



VIII. Approve Item Screen (Pop Up):

When the item has been approved, the item will be approved for the student.

IX. Cancel Subscription Screen (Pop Up):

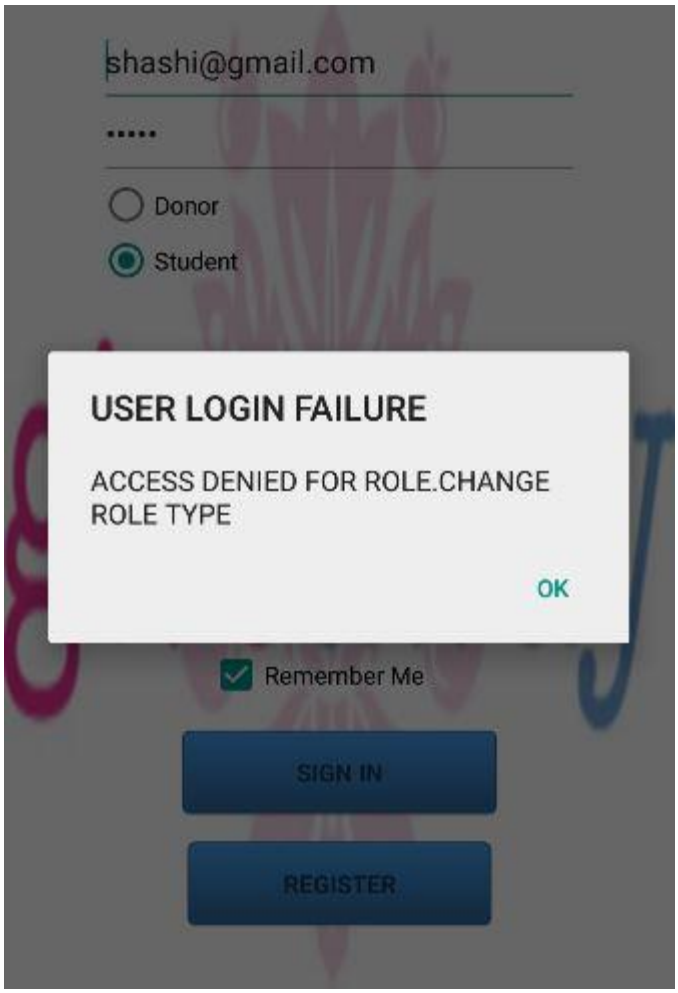
When the items subscription has been cancelled for a student, this pops up.

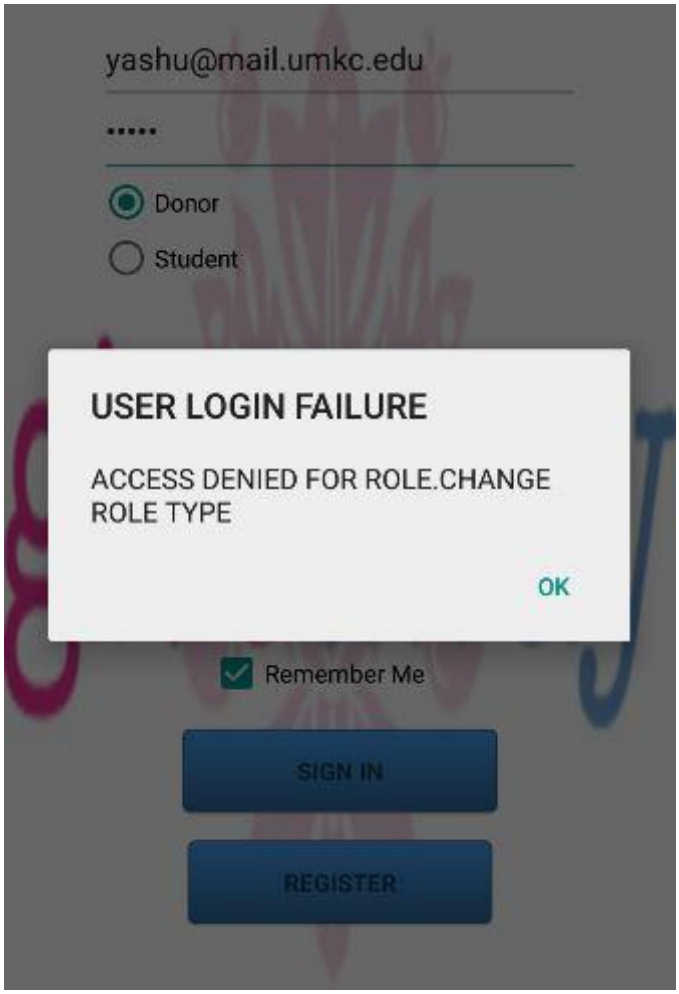
X. View Location on Maps Screen (Google Maps):

On click of View Location on Maps button in the Subscribed student Details screen, the user's location will be displayed in the Google maps.

XI. Login Failures Screen (Pop Up):

When a donor (usually won't have login with .edu domain) tries to login to the system as Student/the student (usually will have mail ending with .edu domain) tries to login to the system as donor the following error pops up.





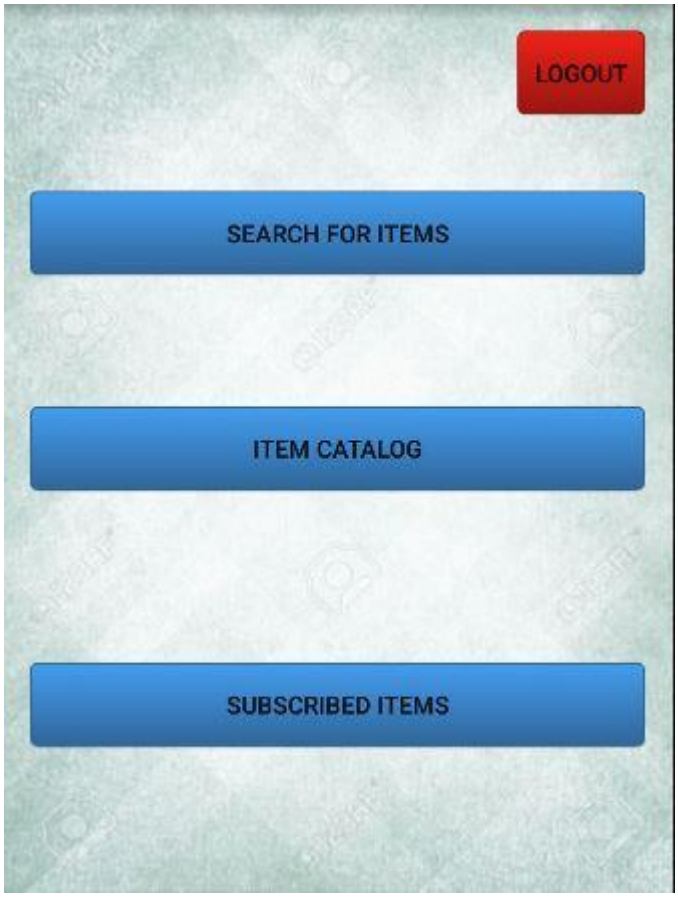
Login as Student:

I. Login Screen:

When the user registers as Student, he can login to the system as Student.

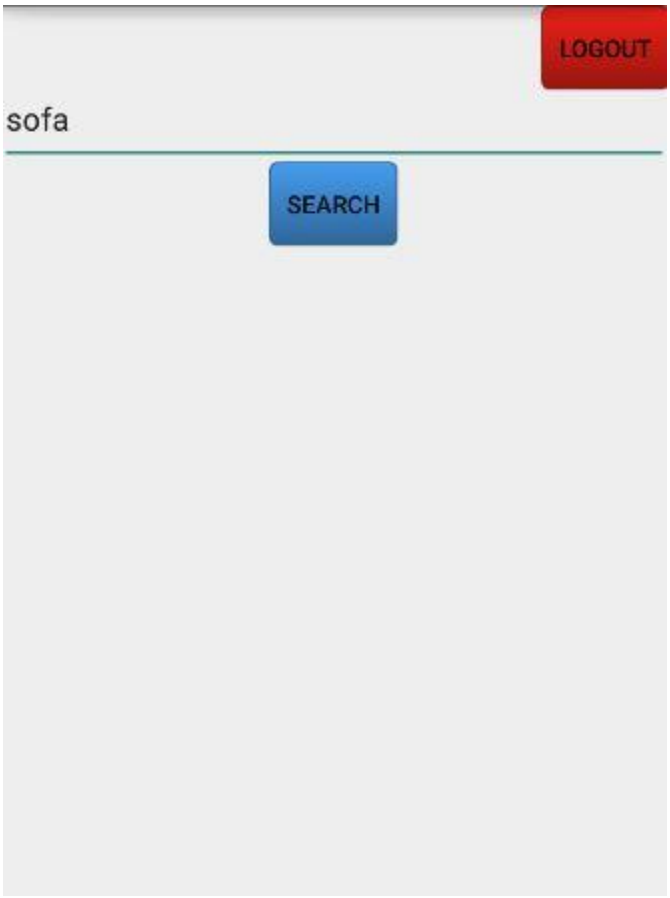
II. Student Dashboard Screen:

When the student gets logged in, he will be redirected to the Dashboard Screen, where the student will be able to search for the items and displayed with the list of items that were posted by the donors and the items that the logged in student has subscribed to.



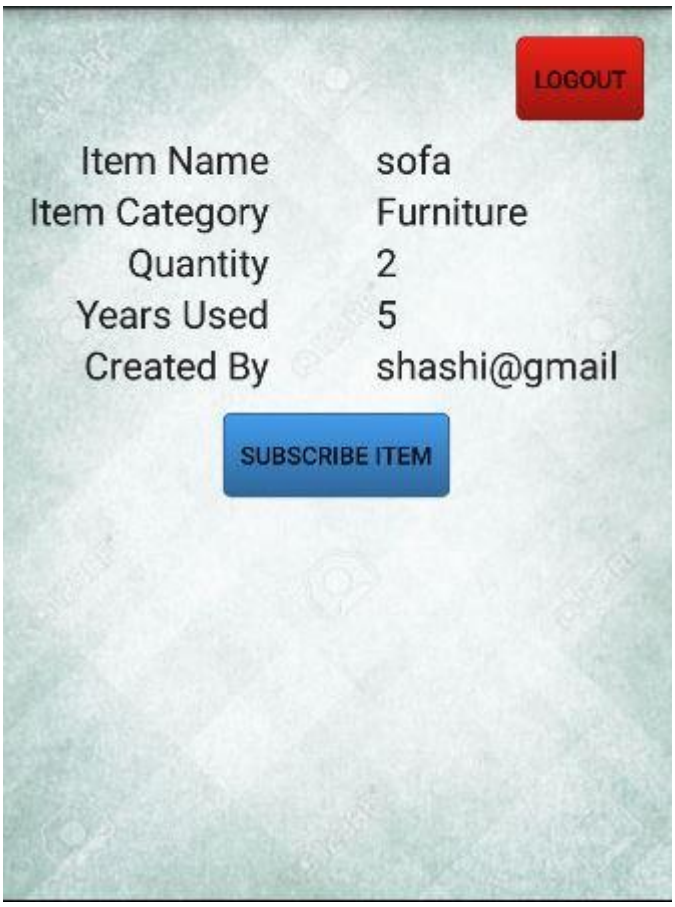
III. Search for Items Screen:

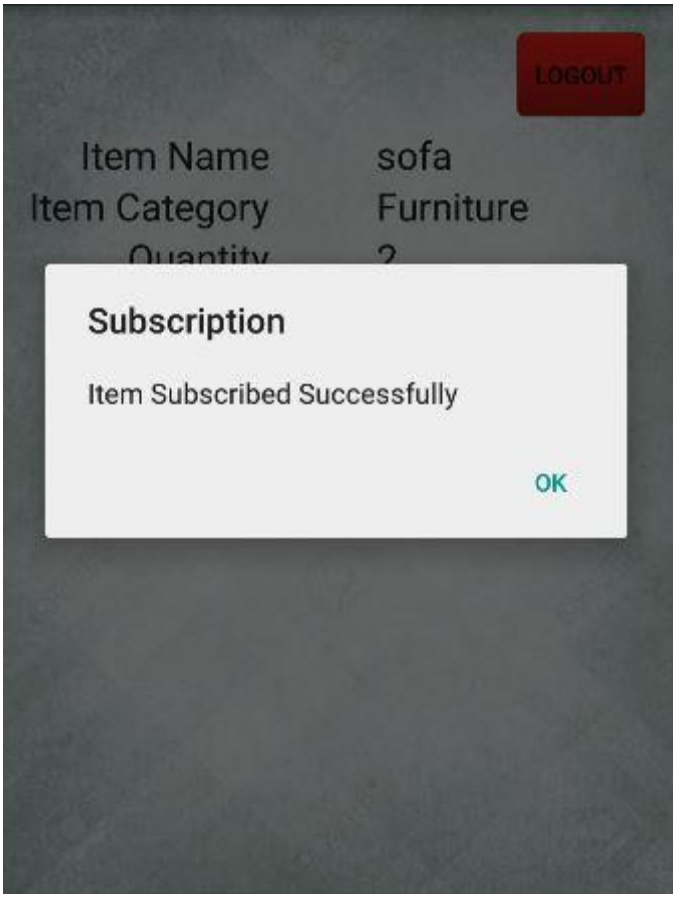
When the student click on the Search for Items button in the Dashboard, he will be directed here where he can search for any item of his wish.



IV. Item Subscription Screen:

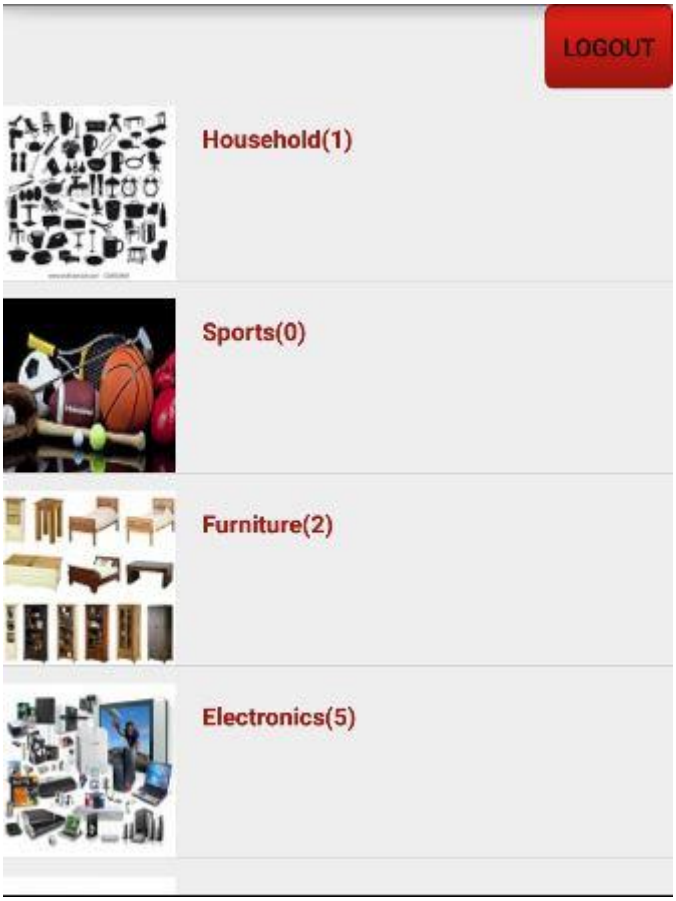
When the student clicks on a particular item in the displayed search results, he will be directed to this screen, where he will be provided with an option for subscribing for the item.





V. Item Catalog Screen:

When the student clicks on Item Catalogue button in the Dashboard, he will be directed to this screen where he will be presented with a catalogue of all the items that were being donated by all the donors.



VI. Item Listing Screen:

When the student clicks on a particular catalogue, then he will be directed to this page where all the items that were listed under that particular catalogue will be displayed.

VII. Subscribed Items List Screen:

When the student clicks on Subscribed Items button in the Dashboard, he will be directed to this screen where he will be listed with all the items that he has subscribed to.

Project Management:

Work completed:

- Description:
    a) Usage of Google maps API for location mapping and distance and time calculation.
    b) Creation of various UI screens and populating the data into them.
    c) Unit testing using JUNIT.
- Responsibility (Task, Person):

    a. **Sashidhar Reddy Gowra**: Usage of Google maps API, to populate the student's destination on the Google maps and give the approximate distance and time of travel for delivery of goods and Deploying corresponding web services of student and donor modules in remote server.

    b. **Venkataramana Yashwant Kumar Palisetty**: Creating Student module screens like search items and view subscribed items screen and populating data into them.

    c. **Ravikanth Devanaboyina**: Creating Donor module screens like Approve Items, Subscribe Items and the data population into the corresponding screens.

    d. **Anudeep Reddy Gujjula**: Creating server side validations to restrict the user logins to the application and maintaining sessions.
- Time taken (#hours):
    UI Design, Service Creation and Server Validations, deploying into server:100 hrs.

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

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

Future Enhancements: Displaying images for the items that are being donated.