

GIVE AWAY THIRD 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 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
Reviewing	Permalink

Tip: Click a field to edit

Close

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
Reviewing	Permalink

Tip: Click a field to edit

Close

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

Close

Story #15

Summary

Creation of User Registrtrion and Login Screens and Corresponding Validations

Detail

As part of Mobile Client Creating screens for User Registrtrion 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

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. 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:**

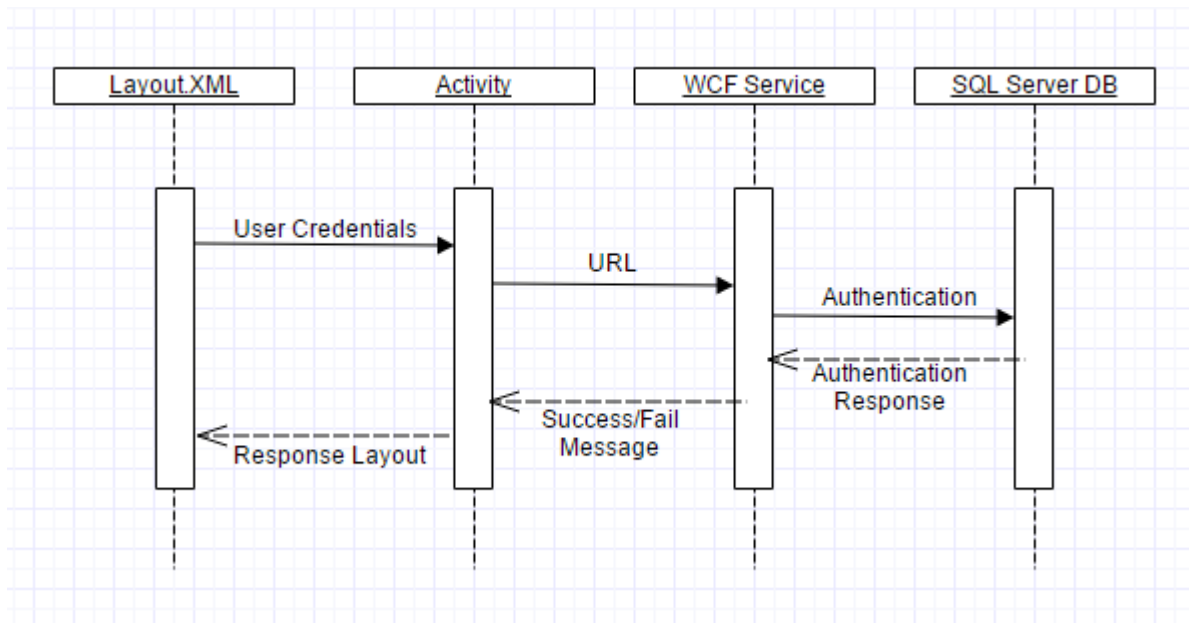
- This service is used to update the details of an item in the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, Method Name and the required input parameters.

- V. Item Retrieval Service:**

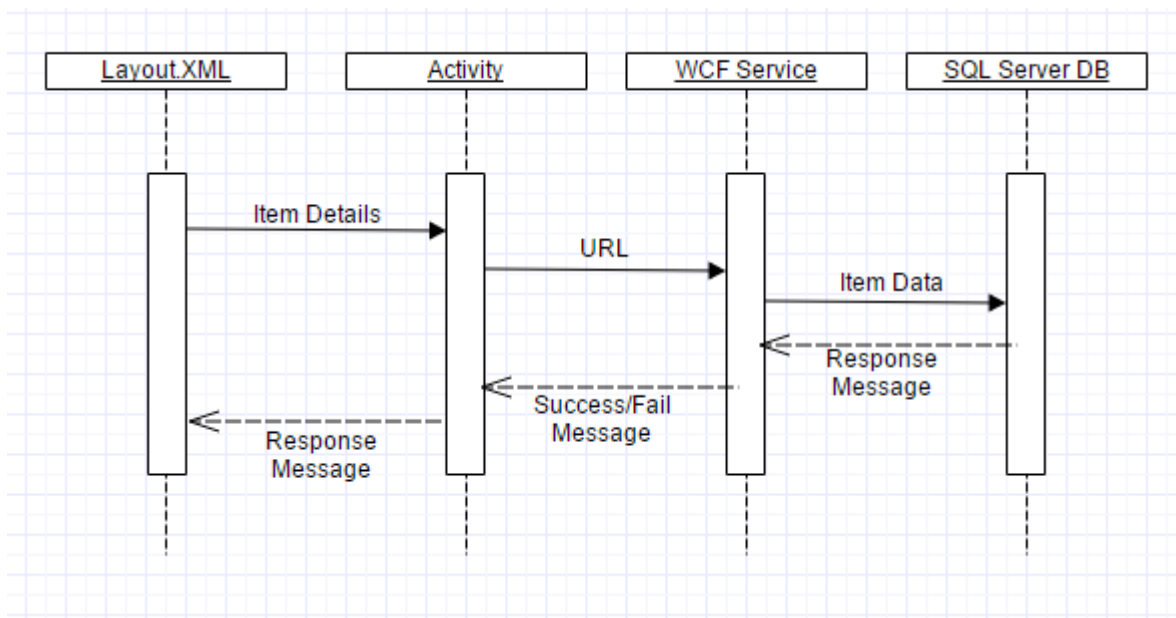
- This service is used to retrieve the list of items from the Database, in which we hit the SQL Server Database using the corresponding IP Address, Port Number, and Method Name and get the required details.

- **Sequence Diagram:**

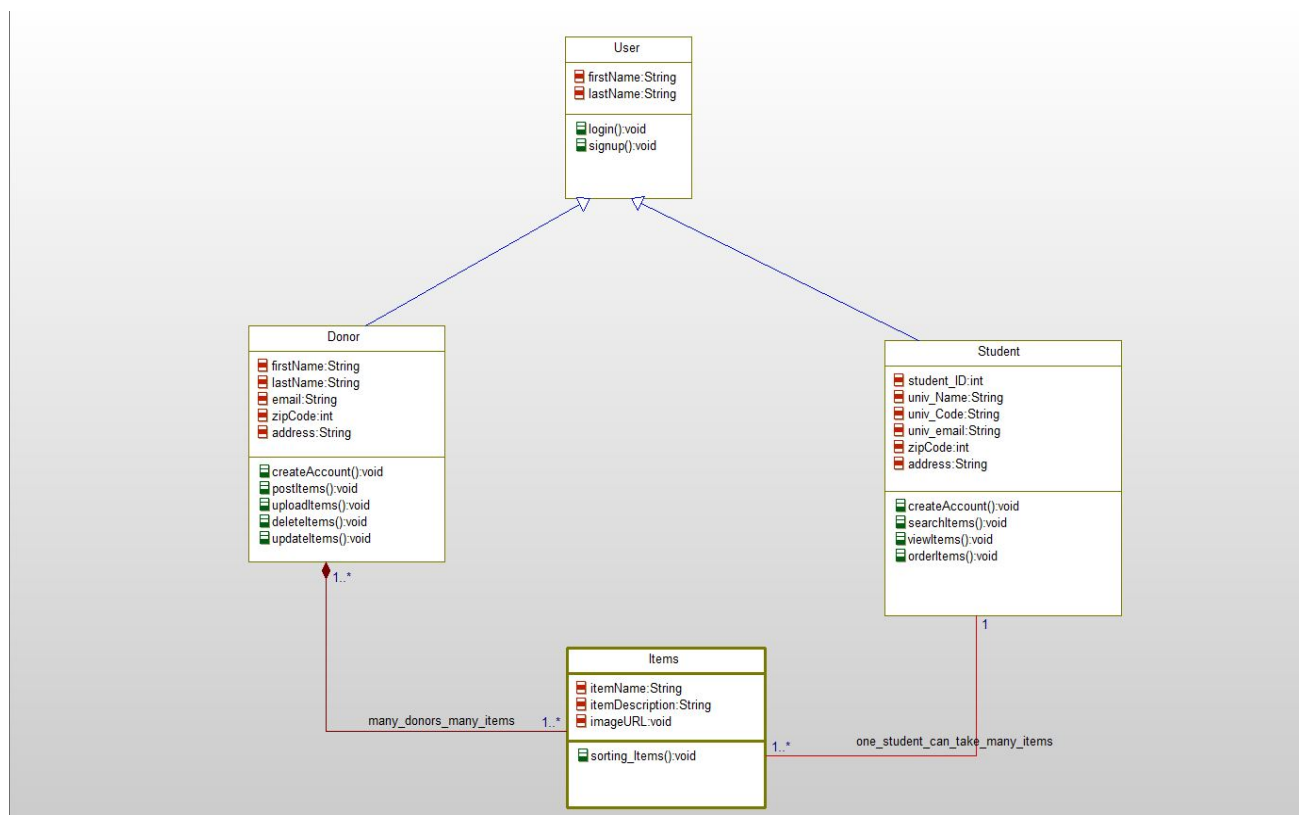
- I. Sequence Diagram for User Login/Registration:**



- 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>

The screenshot shows a REST client interface. At the top, the 'Method' is set to 'GET' and the 'URL' is 'http://localhost:8871/Service1.svc/userLoginDetails/shashi@gmail'. A 'SEND' button is on the right. Below the URL bar is a 'Body' section with a 'Request Body' text area. Underneath is a 'Response' section with tabs for 'Response Headers', 'Response Body (Raw)', 'Response Body (Highlight)', and 'Response Body (Preview)'. The 'Response Body (Raw)' tab is selected, displaying a JSON object:

```
1. {  
2.   "password": "reddy",  
3.   "username": "shashi@gmail"  
4. }
```

- 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'>

The screenshot shows a REST client interface. At the top, the 'Method' is set to 'GET' and the 'URL' is 'http://localhost:8871/Service1.svc/insertRegistrationDetails/'chair','Household','2','04-10-2015','s'. A 'SEND' button is on the right. Below the URL bar is a 'Body' section with a 'Request Body' text area. Underneath is a 'Response' section with tabs for 'Response Headers', 'Response Body (Raw)', 'Response Body (Highlight)', and 'Response Body (Preview)'. The 'Response Body (Raw)' tab is selected, displaying the text:

```
"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'>

The screenshot shows a REST client interface with the following details:

- Request:**
 - Method: GET
 - URL: `http://localhost:8871/Service1.svc/insertItemDetails/'chair','Household','2','04-10-2015','shashi@`
 - Body: Request Body
- Response:**
 - Response Headers: 1. Status Code : 200 OK, 2. Cache-Control : private, 3. Connection : Close, 4. Content-Length : 15, 5. Content-Type : application/json; charset=utf-8, 6. Date : Mon, 13 Apr 2015 22:34:32 GMT, 7. Server : ASP.NET Development Server/10.0.0.0, 8. X-AspNet-Version : 4.0.30319
 - Response Body (Raw): (Empty)
 - Response Body (Highlight): (Empty)
 - Response Body (Preview): (Empty)

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>

The screenshot shows a REST client interface with the following details:

- Request:**
 - Method: GET
 - URL: `8871/Service1.svc/insertItemDetails/'2001','chair','Household','2','04-10-2015','shashi@gmail','2'`
 - Body: Request Body
- Response:**
 - Response Headers: (Empty)
 - Response Body (Raw): (Empty)
 - Response Body (Highlight): (Empty)
 - Response Body (Preview): `"Updated data_statementUpdate ITEMS set ITEMNAME='chair',ITEMCATEGORY='Household',QUANTITY='2',YEARSOFUSAGE='2' where ITEMID = '2001'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/>

The screenshot shows a REST client interface with the following components:

- Method:** GET
- URL:** <http://localhost:8871/Service1.svc/retrieveAllItemDetails/>
- Body:** Request Body (empty)
- Response:** Expanded view showing the response body in raw JSON format.

The response body is a JSON array containing two objects:

```
1. [  
2.   {  
3.     "createdby": "shashi@gmail",  
4.     "itemcategory": "Furniture",  
5.     "itemid": 2000,  
6.     "itemname": "table",  
7.     "quantity": 2,  
8.     "yearsused": 3  
9.   },  
10.  {  
11.    "createdby": "shashi@gmail",  
12.    "itemcategory": "Household",  
13.    "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.

The image displays two versions of a mobile application's registration screen, titled "Register Here". Both screens have a dark header bar with the text "Register_Activity" and a menu icon. The form contains the following fields: First Name, LastName, UserName, Password, Email Address, Mobile Number, Date Of Birth, Address, and Zipcode. A blue "REGISTER" button is located at the bottom of the form.

Left Screenshot: The form is empty, with all input fields ready for user input.

Right Screenshot: The form shows validation errors. A red exclamation mark icon is next to the "LastName" field, with a tooltip that says "This field is required". Red exclamation mark icons are also present next to the "UserName", "Password", "Email Address", "Mobile Number", "Date Of Birth", "Address", and "Zipcode" fields, indicating they are required but empty.

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.

The image displays two versions of a login screen. Both screens have a white background with a faint pink floral pattern. The left screen shows the initial login form with the email field filled with 'shashi@gmail', the password field masked with dots, and the 'Student' radio button selected. The right screen shows the same form after an incorrect password attempt, with a red exclamation mark icon and a black error message box stating 'This password is incorrect'.

shashi@gmail

.....

☐ Donor

☒ Student

SIGN IN

REGISTER

shashi@gmail

...

☐ Donor

☒ Student

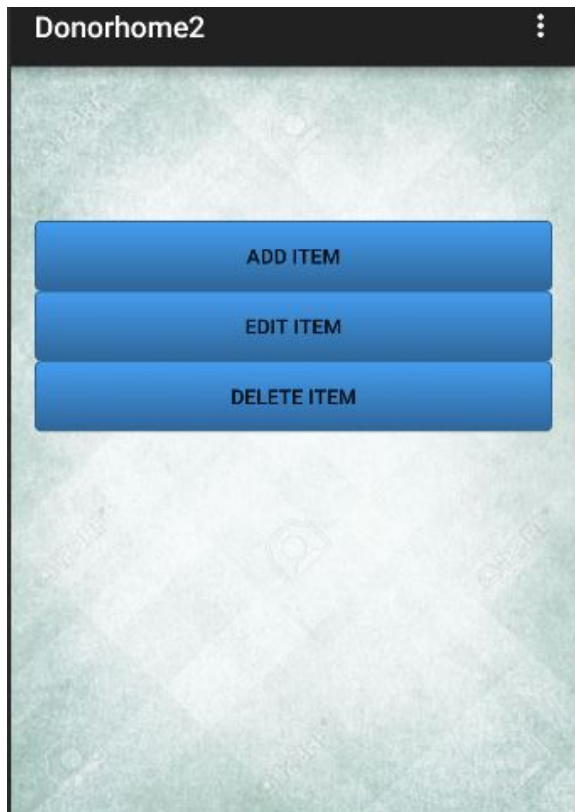
This password is incorrect

SIGN IN

REGISTER

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.

The image displays two side-by-side screenshots of a mobile application screen titled 'addItem1'. Both screens feature a light green background with a faint, repeating pattern of icons. The form on the screen includes the following elements:

- Item Name:** A text input field at the top.
- Item Category:** A dropdown menu below the name field. In the left screenshot, it shows 'Household' with a downward arrow. In the right screenshot, the dropdown is open, revealing a list of categories: Household, Sports, Furniture, Stationary, Electronics, Food, Accessories, and Utilities.
- Quantity:** A text input field below the category dropdown.
- Years Used:** A text input field at the bottom of the form.
- ADD ITEM:** A blue rectangular button located at the bottom center of the screen.

The left screenshot shows the form in its default state with the category dropdown closed. The right screenshot shows the same form with the category dropdown menu open, displaying the available options.

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.



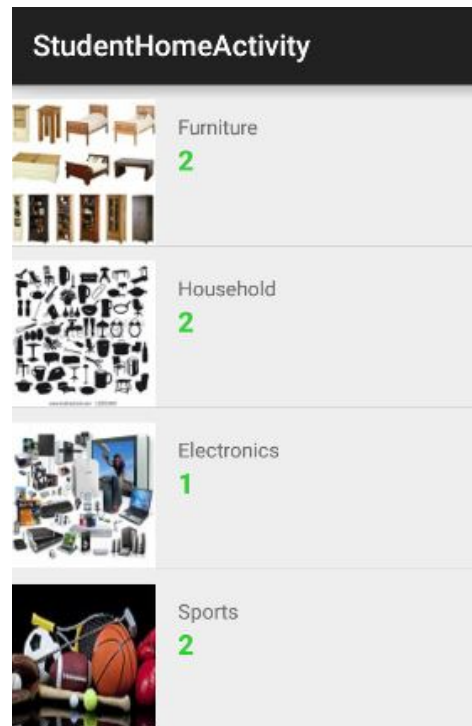
The screenshot displays the 'EditItemActivity' screen. At the top, the title bar shows 'EditItemActivity' and a three-dot menu icon. Below the title bar, the email address 'shashi@gmail' is displayed. The main content area contains a form with the following elements:

- A dropdown menu currently showing 'table'.
- A text input field containing 'table'.
- A dropdown menu currently showing 'Household'.
- A text input field containing '2'.
- A text input field containing '3'.

At the bottom of the form, there is a button labeled 'SAVE ITEM'.

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:

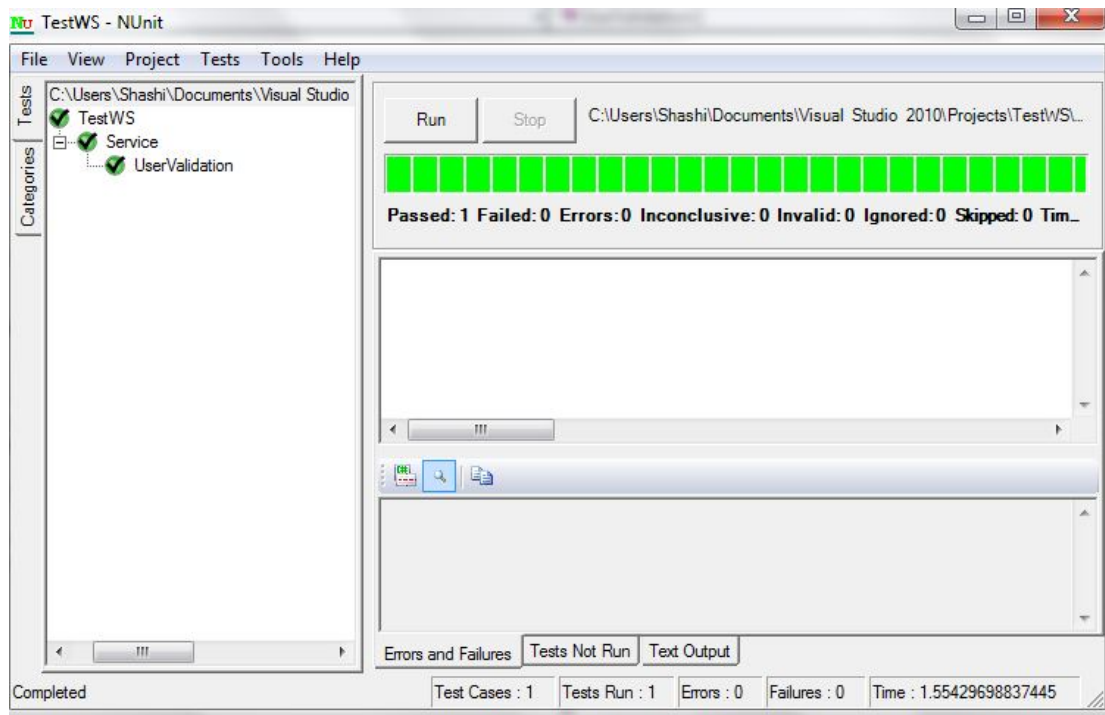
The Student can subscribe to any item which he likes using this screen.

Testing:

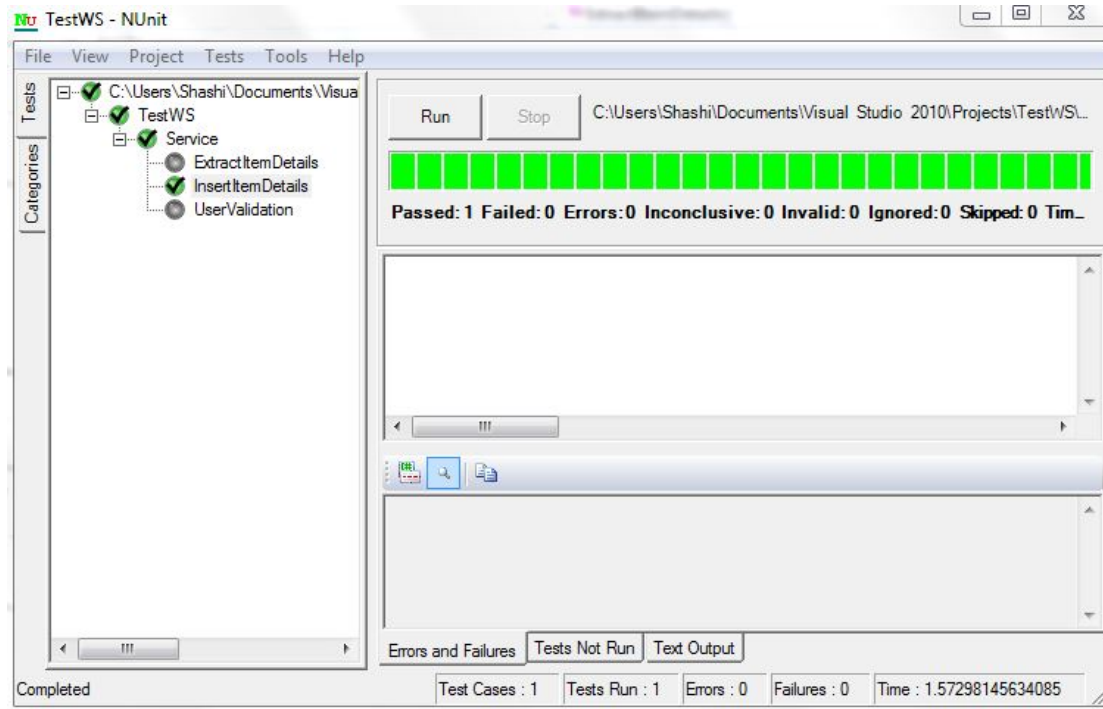
Functional Testing

NUnit Test Cases:

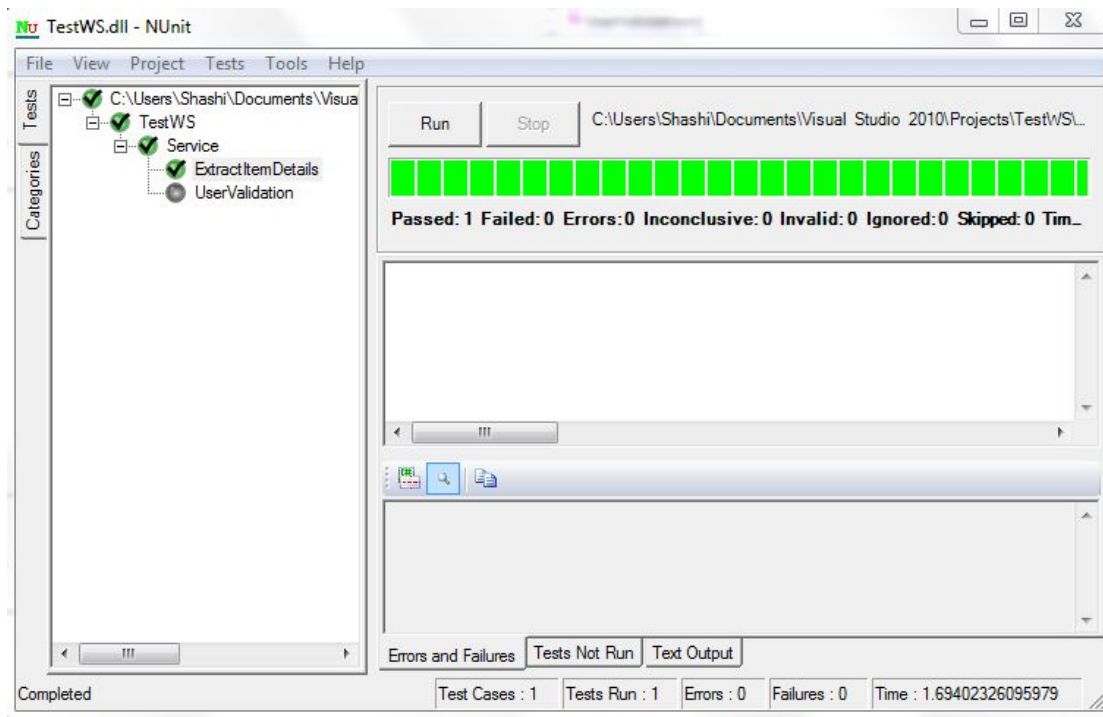
a. User Login:



b. Item Addition:



c. Item List Retrieval:



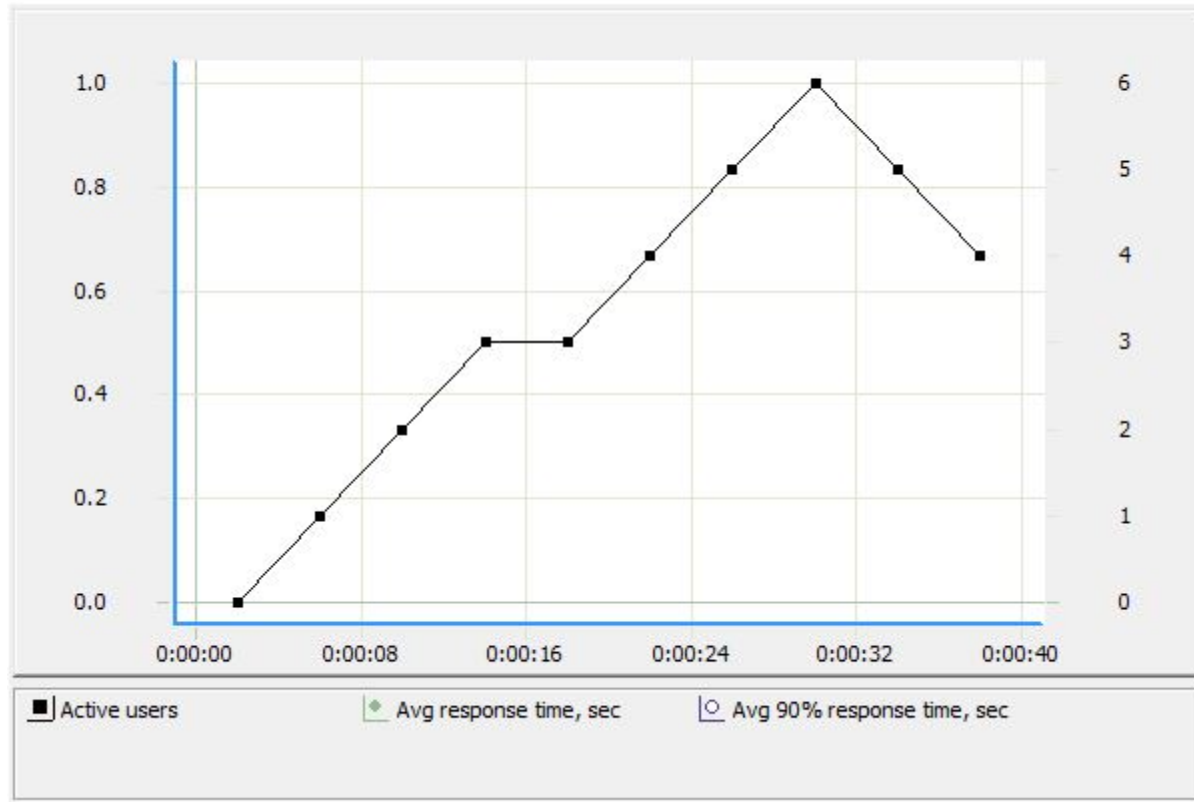
Deployment Testing:

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:

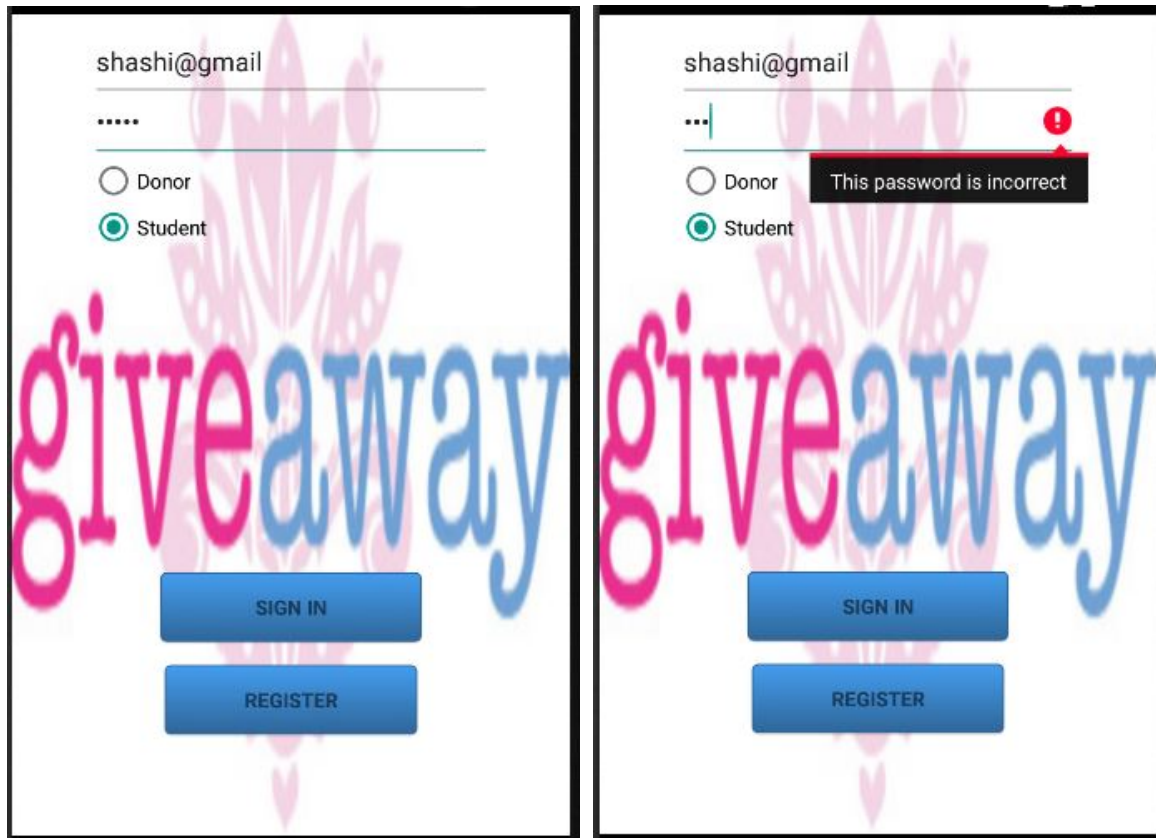
The user has to first register to the application by giving the required data like First Name, Last Name, User Name, Password, Email ID, Mobile Number, Date of Birth, Address and Zip Code.

The image displays two screenshots of the 'Register Here' screen in an Android application. The screen has a title bar 'Register_Activity' and a subtitle 'Register Here'. Below the subtitle, there are nine input fields: 'First Name', 'LastName', 'UserName', 'Password', 'Email Address', 'Mobile Number', 'Date Of Birth', 'Address', and 'Zipcode'. At the bottom, there is a blue 'REGISTER' button.

The left screenshot shows the form with all fields empty. The right screenshot shows the form with validation errors. A red exclamation mark icon is next to the 'First Name' field. A black tooltip with the text 'This field is required' is shown next to the 'LastName' field. Red exclamation mark icons are also shown next to the 'UserName', 'Password', 'Email Address', 'Mobile Number', 'Date Of Birth', 'Address', and 'Zipcode' fields.

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.



The image displays two screenshots of a login screen for an application. Both screens feature a light blue background with a faint pink floral pattern and a large, semi-transparent 'giveaway' watermark in the center.

Left Screenshot (Initial Login Screen):

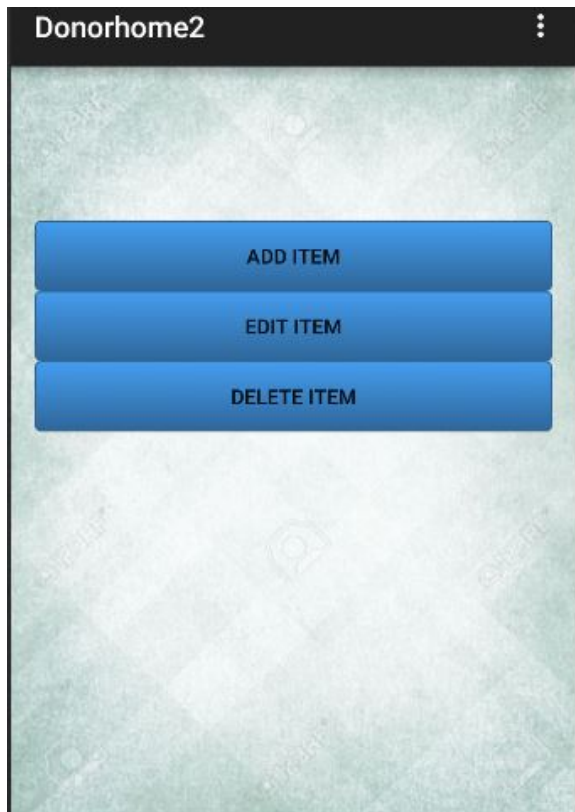
- Email field: shashi@gmail
- Password field: masked with dots (.....)
- Role selection: ☐ Donor, ☒ Student
- Buttons: SIGN IN, REGISTER

Right Screenshot (Login Failure Screen):

- Email field: shashi@gmail
- Password field: masked with dots (....), with a red exclamation mark icon indicating an error.
- Error message: A black box with white text stating 'This password is incorrect'.
- Role selection: ☐ Donor, ☒ Student
- Buttons: SIGN IN, REGISTER

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.

The image displays two side-by-side screenshots of a mobile application screen titled 'addItem1'. The screen has a light green background with a faint pattern of icons. The form consists of four input fields: 'Item Name', 'Item Category', 'Quantity', and 'Years Used'. The 'Item Category' field is a dropdown menu with 'Household' selected. A blue 'ADD ITEM' button is located at the bottom center of the screen. The right screenshot shows the dropdown menu open, displaying a list of categories: Household, Sports, Furniture, Stationary, Electronics, Food, Accessories, and Utilities.

addItem1

Item Name

Household

Quantity

Years Used

ADD ITEM

addItem1

Item Name

Household
Sports
Furniture
Stationary
Electronics
Food
Accessories
Utilities

ADD ITEM

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.



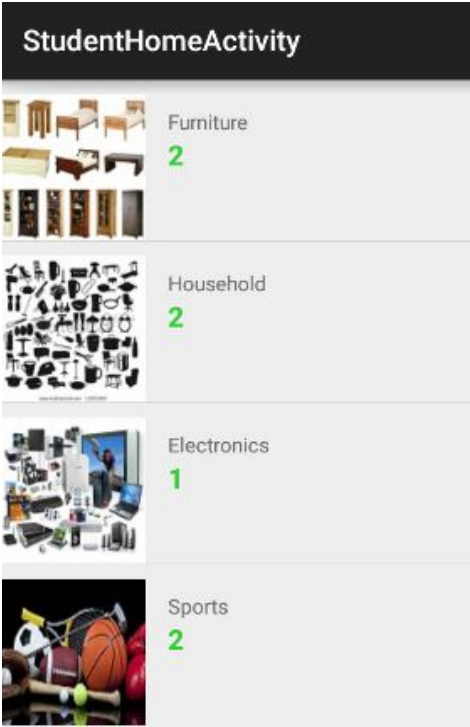
The screenshot displays the 'EditItemActivity' screen. At the top, the title bar shows 'EditItemActivity' and a three-dot menu icon. Below the title bar, the email address 'shashi@gmail' is displayed. The main content area contains a form with the following elements:

- A dropdown menu currently showing 'table'.
- A text input field containing 'table'.
- A dropdown menu currently showing 'Household'.
- A text input field containing '2'.
- A text input field containing '3'.

At the bottom of the form, there is a button labeled 'SAVE ITEM'.

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.



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.