**CS551 Advanced Software Engineering**

**First Increment Report**

**Project Title: PickMeUp**

**Submitted by**

PG6 (SG14 and SG15)

Ponnada Rahul (Class ID-39)

Ghanta Surya Prabha (Class ID-19)

Tummala Anvesh (Class ID-48)

Anumolu Satish Chowdary (Class ID-2)

**Import Existing Services/API**

In this Increment, we implemented three web services each one for a different purpose. The web services are as follows -Student and volunteer login, student registration and volunteer registration. There are no external API used as of now. The main purpose of these web services is to validate the student and volunteers then register their details to a centralized storage.

-Login web service for students and volunteers to login

-Registration web service for students and volunteers to register with the app

**Detail Design of Services**

**User Stories:**

We have four stories in iteration1

1. As a student, I want to create an account so that I can log-in to access the services provided.
2. As a student I want to log-in to my account so that I can access the services provided.
3. As a volunteer, I want to create an account so that I can log-in to access the services provided.
4. As a volunteer I want to login to my account so that I can access the services provided.

**Service description:**

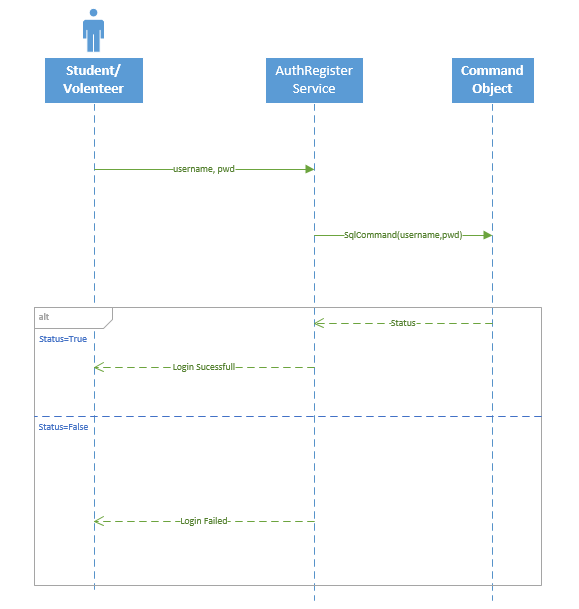
Login is a web service that takes *student id* and *password* as parameters. Then the service authenticates whether the specified student exists or not. If a student exists, then his password is authenticated against the password from his record from a centralized database and the return status as true, otherwise false. Hence, this service connects to the underlying database to fetch passwords of the respective users for validation.

Student and volunteer registration web services are pretty much similar except in the parameters they provide and underlying database table they access. It access the data provided by the respective user (Student or Volunteer) and records those data onto a respective database table and returns 1 for successful registration.

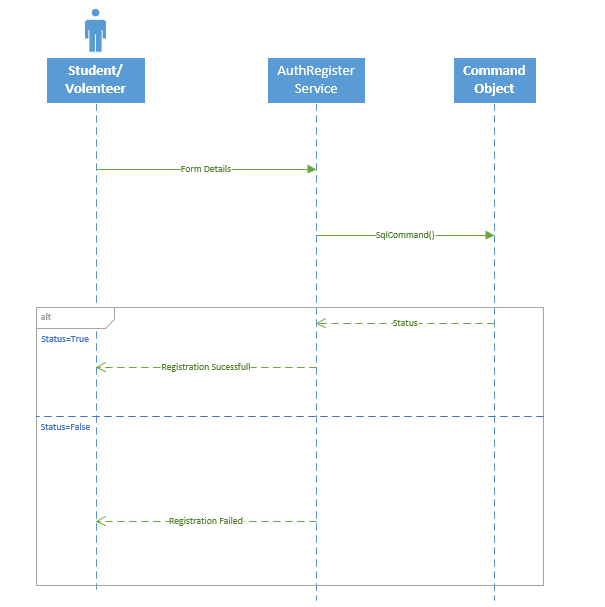
These web services return JSON data back to the client to verify the operation has successfully completed on the server side. REST uses JSON as it data exchange format so as in here for all the web services. Microsoft SQL server is the persistence storage that these services store data on to table and retrieve them for later verification, validation and population purposes.

**Sequence diagrams:**

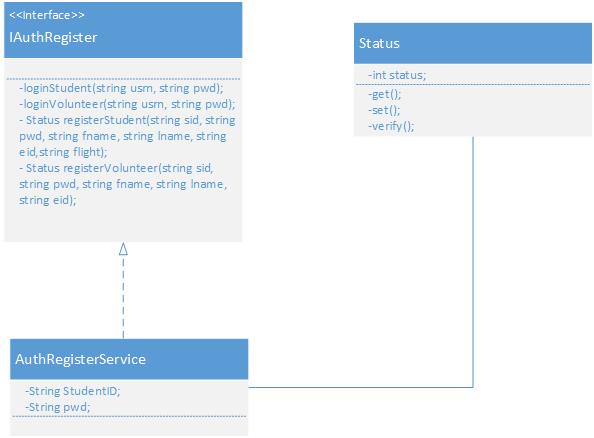
**Student/Volunteer Login**



**Student/Volunteer registration:**

****

**Class Diagram:**

****

**Design of Mobile Client Interface:**

Mobile client interface is an Android application which is a rich client. An application is considered rich client if it has all the UI required on client’s side. Overall UI design is developed using Android studio. We are using the Base.V21.Theme.AppCompat theme for our application and Nexus 5 API 21 as emulator for testing our application. As of now, for the first iteration, app consists of six different screens.

1. Start Screen which prompts user to select student or volunteer.
2. Login Screen for student and volunteer.
3. Student Registration screen.
4. Volunteer Registration screen.
5. Student home screen after successful login
6. Volunteer home screen after successful login.

User Interacts by a touch based smartphone there by navigating to other screens and perform operations on the server. Typical mobile client flow of operations is as follows. When the student or a volunteer install the app, they are asked to identify themselves (to distinguish between student and volunteer). Then they are redirected to a login page where already existing users can login and new users can register. After successful registration, users are navigated back to the Start screen form where they can login to their respective home pages (Student Home Screen & Volunteer Home Screen).

**Design of Unit test cases (using NUnit tool):**

Test cases are designed to test the Login and Registration services. This is implemented using visual studio and executed by NUnit Client. Our test case consists of four methods to test the Student login, Volunteer Login, Student Registration and Volunteer Registration functionalities.

**Implementation**

**Implementation of REST services:**

WCF (Windows Communication Framework) is used to implement REST web services on Visual Studio 2010. Web service project has an endpoint IAuthRegister.cs, which is also called as contract and Implementation of these resources is in AuthRegister.svc.cs. The implementation has several resources implemented and are ready to be consumed from a client. Resources communicate directly with the underlying database.

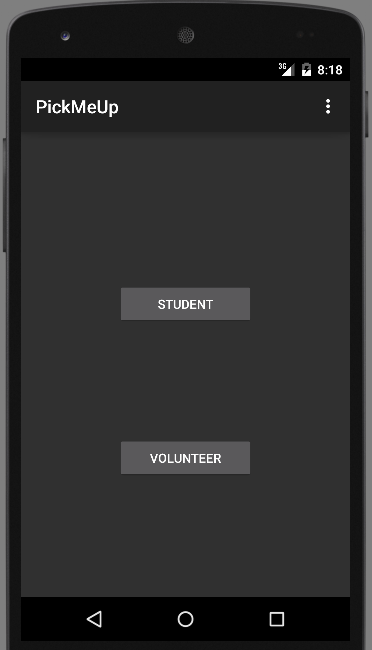
Login service validates existing student or volunteer by the taking the username and password from the mobile client side and runs a select query on the respective table. Response will be returned by the service as true if the login is successful or false otherwise.

Registration services saves student and volunteer account information in the system. Both the services will take the respective account information from the Mobile client side and Updates the respective student or volunteer tables respectively.

**Implementation of user interface (Mobile Apps):**

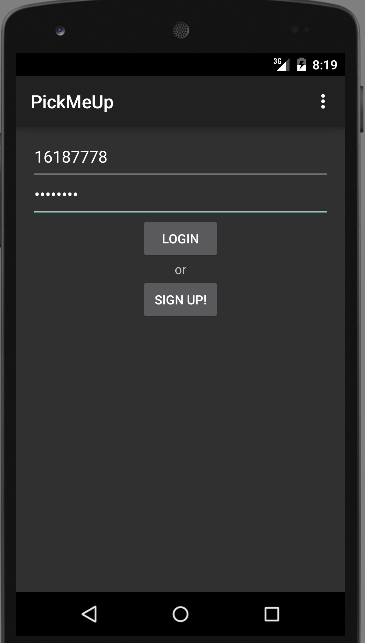
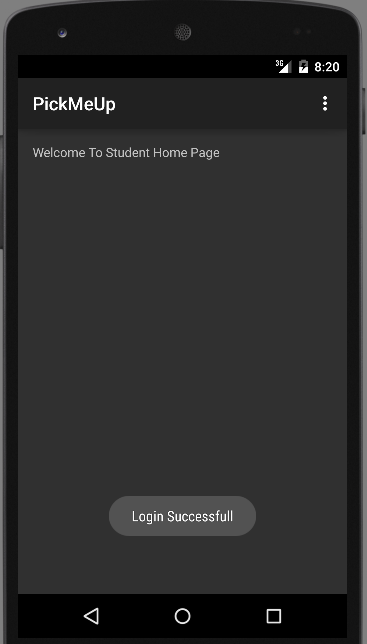
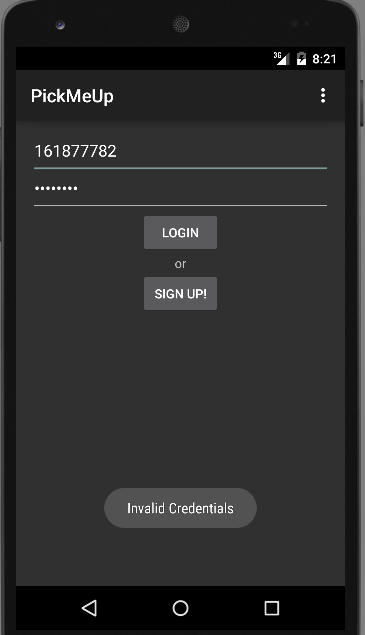
Android studio is being used to implement the Mobile App. User Interface of the app is XML based and is relatively changeable to the screen size. A total of six activities are created for the six screens designed for this increment.

1. **Start Screen:** It is the MainActvity consists of two buttons student and volunteer. Upon click of any button, it will navigate to Login screen and will also pass the respective button name to distinguish between student and volunteer for the next levels of navigation from login screen.

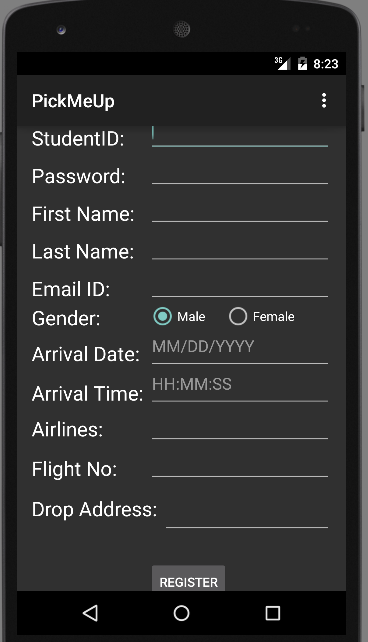
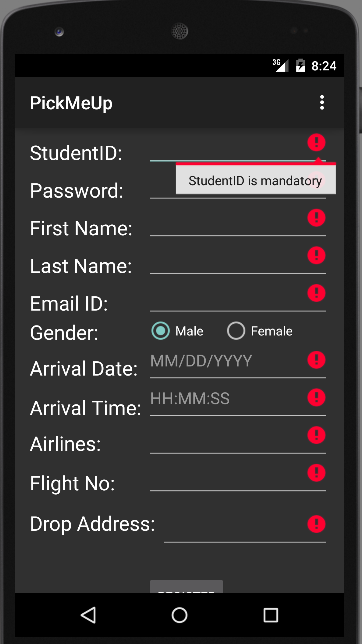
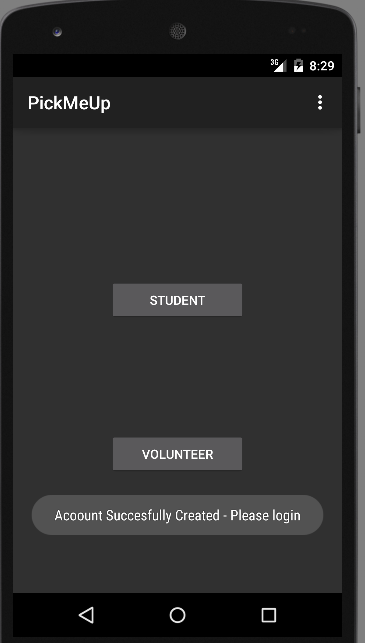


1. **Login Screen:** It is the LoginActivity which consists of StudentID, Password fields and Login, Sign Up buttons. Existing users will navigate to their respective Home screen by the providing the login details. A new private class AuthenticationService is written inside the LoginActivity which uses HTTP request to call the Login Service from the Mobile client side. Upon click of the login button, respective login service is called to validate the login credentials and response of true or false is returned. If true is returned, a Toast message “Login Successful” is displayed and the user will be navigated to the respective Home screens. If false is returned, a Toast message “Invalid Credentials” will be displayed.

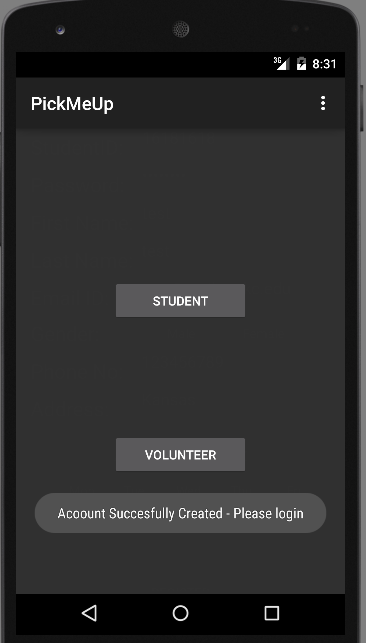
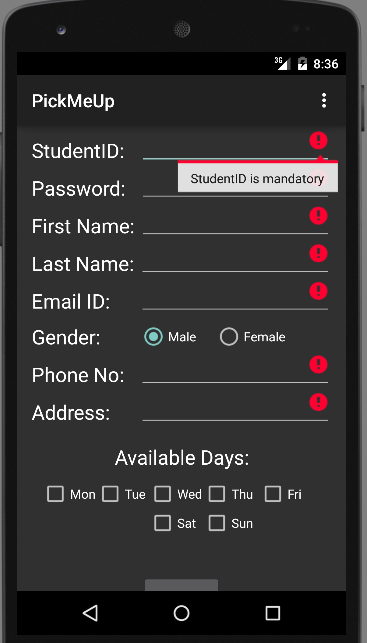
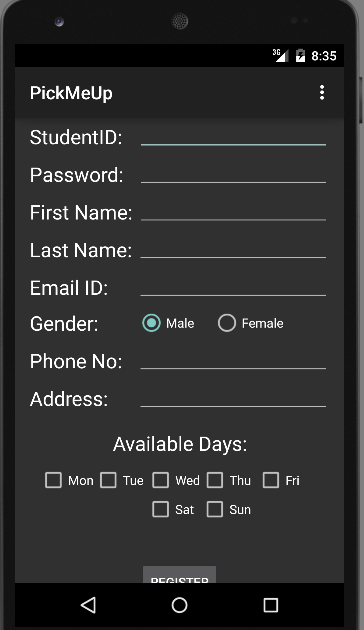
Upon click of the signup button, users will be navigated to Student Registration screen or Volunteer registration screen respectively.

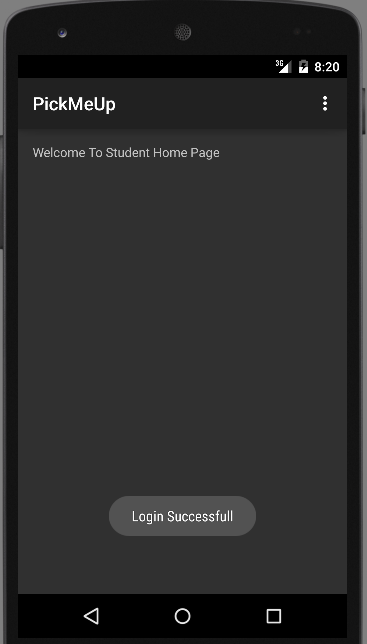
1. **Student Registration Screen:** It is the StudentRegActivity which consists of fields like StudentID, Password, FirstName, LastName, Email ID, Gender, Arrival Date, Arrival Time, Airlines, Flight No and Drop Address to create a student account. Here the studentID field is numeric type, password is Password type, Gender is Radio Button type, Arrival Date and Time are of Date type, Airlines is autocomplete Text view type fields and remaining fields are normal text fields. Many of these are mandatory fields and Validated after the click of Register button. A new service StudentRegistration is written inside the StudentRegActivty which uses HTTP request to call the Student Registration service from the mobile client side. Upon validation of validation of required fields, this service will be called to create an account for the student. A new Toast message “Account Created Successfully – Please Login” will be displayed to the user and is navigated to the Start Screen.

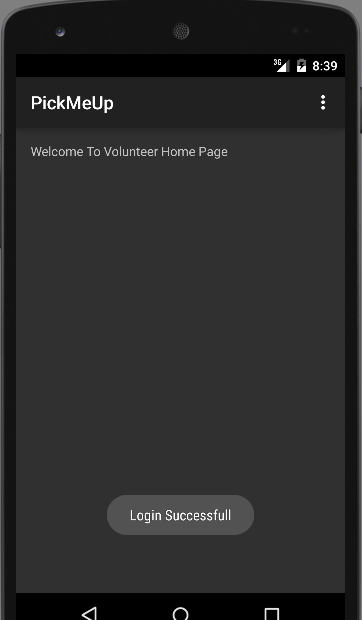
1. **Volunteer Registration Screen:** It is the VolunteerRegActivity which consists of fields like StudentID, Password, FirstName, LastName, Email ID, Gender, Phone No, Address and Available Days fields to create a volunteer account. Here the studentID field is numeric type, password is Password type, Gender is Radio Button type, Available Days consists of Check Boxes for all the days of the week and remaining fields are normal text fields. Many of these are mandatory fields and Validated after the click of Register button. A new service VolunteerRegistration is written inside the VolunteerRegActivty which uses HTTP request to call the Student Registration service from the mobile client side. Upon validation of validation of required fields, this service will be called to create an account for the student. A new Toast message “Account Created Successfully – Please Login” will be displayed to the user and is navigated to the Start Screen.



1. **Student Home Screen:** It is the StudentHomeActivity which consists of simple text message “Welcome to Student Home Page”. This screen will be designed in the next increments.



1. **Volunteer Home Screen:** It is the VolunteerHomeActivity which consists of simple text message “Welcome to Volunteer Home Page”. This screen will be designed in the next increments.

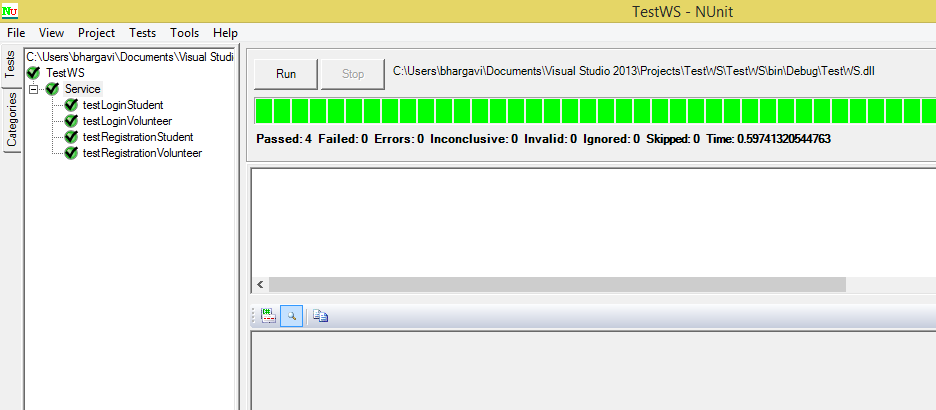


**Implementation of test cases:**

Test cases are implemented by using visual studio and executed by NUnit Client. We have a NUnit test class Service. It consists of four test methods, testLoginStudent- to test the student Authentication process, testLoginVolunteer - to test the Volunteer Authentication process, testRegistrationStudent – to test the student Registration process and testRegistrationVolunteer - to test the volunteer Registration Process.

**Testing: Perform Unit testing (using NUnit tool)**

All the implemented tests are executed using NUnit Client and all the four test cases are passed.

****

**Deployment:**

**WebService URLs**

**Student Login:**

[**http://kc-sce-cs551.kc.umkc.edu/aspnet\_client/Group6/PickMeUpService/AuthService.svc/student/login?username=12345&passwd=erty**](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group6/PickMeUpService/AuthService.svc/student/login?username=12345&passwd=erty)

**Volunteer Login:**

[**http://kc-sce-cs551.kc.umkc.edu/aspnet\_client/Group6/PickMeUpService/AuthService.svc/volunteer/login?username=12345&passwd=erty**](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group6/PickMeUpService/AuthService.svc/volunteer/login?username=12345&passwd=erty)

**Student Registration**

[**http://kc-sce-cs551.kc.umkc.edu/aspnet\_client/Group6/PickMeUpService/AuthService.svc/register/volunteer?studentid=123453&passwd=qwerty&firstname=george&lastname=bush&email=bush@whitehouse.com&gender=M&phone=8164821266&address=WashingtonDC&available=TuThFrSaSuMoWe**](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group6/PickMeUpService/AuthService.svc/register/volunteer?studentid=123453&passwd=qwerty&firstname=george&lastname=bush&email=bush@whitehouse.com&gender=M&phone=8164821266&address=WashingtonDC&available=TuThFrSaSuMoWe)

**Volunteer Registration:**

[**http://kc-sce-cs551.kc.umkc.edu/aspnet\_client/Group6/PickMeUpService/AuthService.svc/register/volunteer?studentid=jtykkjdkj&passwd=cdkkjs&firstname=dkjsj&lastname=dskjksj&email=dsdsjhjskj&gender=Mnmdc&phone=dhshjk&address=dsdsas&available=jksak**](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group6/PickMeUpService/AuthService.svc/register/volunteer?studentid=jtykkjdkj&passwd=cdkkjs&firstname=dkjsj&lastname=dskjksj&email=dsdsjhjskj&gender=Mnmdc&phone=dhshjk&address=dsdsas&available=jksak)

**Project Management:**

**ScrumDo Link:** <http://www.scrumdo.com/projects/project/umkc_pg6/iteration/119777>

**Implementation status report:**

**Work Completed:**

**Description:**

1. As a student, I want to create an account so that I can log-in to access the services provided.

Responsibility: Rahul

Time Taken: 20 hrs

Contribution: 100%

1. As a student I want to log-in to my account so that I can access the services provided.

Responsibility: Anvesh

Time Taken: 20 hrs

Contribution: 100%

1. As a Volunteer, I want to create an account so that I can log-in to access the services provided.

Responsibility: Satish

Time Taken: 20 hrs

Contribution: 100%

1. As a Volunteer, I want to login to my account so that I can access the services provided.

Responsibility: Prabha

Time Taken: 20 hrs

Contribution: 100%

**Work To be completed:** None

**Issues/Concerns:**

* There are few compatibility issues using Visual Studio 2010 & 2013
* Data type conversion issues when inserting data into the database
* Build issues were there