**CS551 Advanced Software Engineering**

**Second 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 and have used two external services. The web services we implemented are as follows -Student registration a modified version including assignment of volunteers of previous increment, SendNotifications to send notification emails, IntelligentSysytem that assigns Volunteers to students based up on the time availability. The external APIs used are Weather API for Climate details, Google Map API for Navigation. The main purpose of these web services is to assign the students with volunteers based on their matching schedules and sending notifications to Volunteers when a student has been assigned.

-Student registration (Modified) service for students to register and then it uses IntelligentSysytem to assigns volunteers and gives the assigned Volunteers name after successful registration.

- IntelligentSysytem service is for assigning Volunteers to students based on their matching times.

- SendNotifications service is for sending notifications to Volunteers when a student has been assigned.

- Weather API used for showing the climate details of the arrival location for students.

- Google Map API used by Volunteers for navigating to the student specified destination locations.

**Detail Design of Services**

**User Stories:**

We have five stories in iteration2

1. As a student when there is a volunteer available to my request I get notified so that I get to know who is coming to pick me up
2. As a volunteer when I update my available timings, the system assigns a student to me so that I am contributing my free time
3. As a student when I update my flight arrival timings the system assigns a volunteer to me so that I have a volunteer who is coming to pick me up
4. As a student I want to cancel the pickup so that someone else can use the service
5. As a student I want re-request a cancelled request so that I can use the cancelled service

**Service description:**

*Intelligent system* is a web service that is the core component of this project. The main task of this system is allot volunteer to students upon their arrival timings.

Assigning a volunteer is a part of the web service that is invoked when a volunteer is registered or change his available timings. This service is invoked after the volunteer has successfully registered there by checking his availability timings against student arrival timing. The algorithm that we developed here follow certain rules in assigning a volunteer to the student.

Assigning a student is also a part the of web service that is invoked when the student gets registration. His arrival timing is checked against the volunteer availability. If any volunteer is available he is assigned to that volunteer.

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.

**sendNotification:**

sendNotification service will send email notifications to volunteer and users regarding their pickup. When a student is assigned to a volunteer. Both receive an email notification of the service. Email notification can be set to different services such as registration, change in timings and update to a new volunteer and so on.

**IntelligentSysytem:**

IntelligentSysytem service will assign volunteers to students based on the available times of Volunteers and students arrival times. If there are is no volunteer available for the pickup of a student, it will send email notification to Admin to have volunteer for picking that student. The algorithm we followed to assign Volunteers is as follows.

**Algorithm**

We have come up with our own algorithm to assign volunteers for picking students. We have taken input for the available times of Volunteers for a week, assuming he will have a recursive schedule that will be same for all weeks. For each day in the week, we have taken 8 bit input, each bit refers to his availability of every 3 hours. So, the volunteers 24 hours availability is taken input for each 3 hours i.e. {0-3, 3-6, 6-9, 9-12, 12-15, 15-18, 18-21, 21-24}.

**Requirement1:** The Volunteers availability is to be stored in DB as follows, if a Volunteer is available on Monday, Tuesday, Saturday and if he is available from 12-18h on Monday, from 21-24h on Tuesday, fully available on Saturday, he will input his availability as (48-00110000, 128-10000000, 0, 0, 0, 255-11111111, 0). This should be stored in DB as availability of Volunteers.

**Requirement2:** The arriving time of the students is to be stored in the DB as DateTime format.

**Input:** String: StudentID

**Step1:** Extract the arrival time of the student from DB based on input StudentID.

**Step2:** Get the dayOfWeek of arrival time of student, let say Sunday.

**Step3:** Look for the time slot student is coming. I.e is he is coming at 11:15 AM the his daySlot will be 8 - (00001000) bit 1 at (9-12 time period).

**Step4:** From the details of the dayOfWeek and time slot, query the Volunteer database for the week schedule of dayOfWeek

such that Get TOP of VolunteerAvaialability &(bitwise AND) daySlot !=0 AND sort by noassignedstudents ASC.

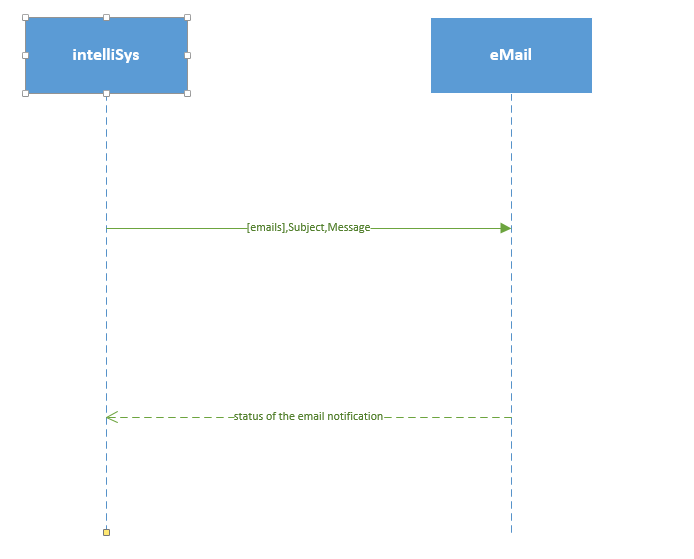
Then we will get the Volunteer who is available for the pickup of student having less no of students being assigned.

**Step5:** Notify the Volunteer about the assigned student details.

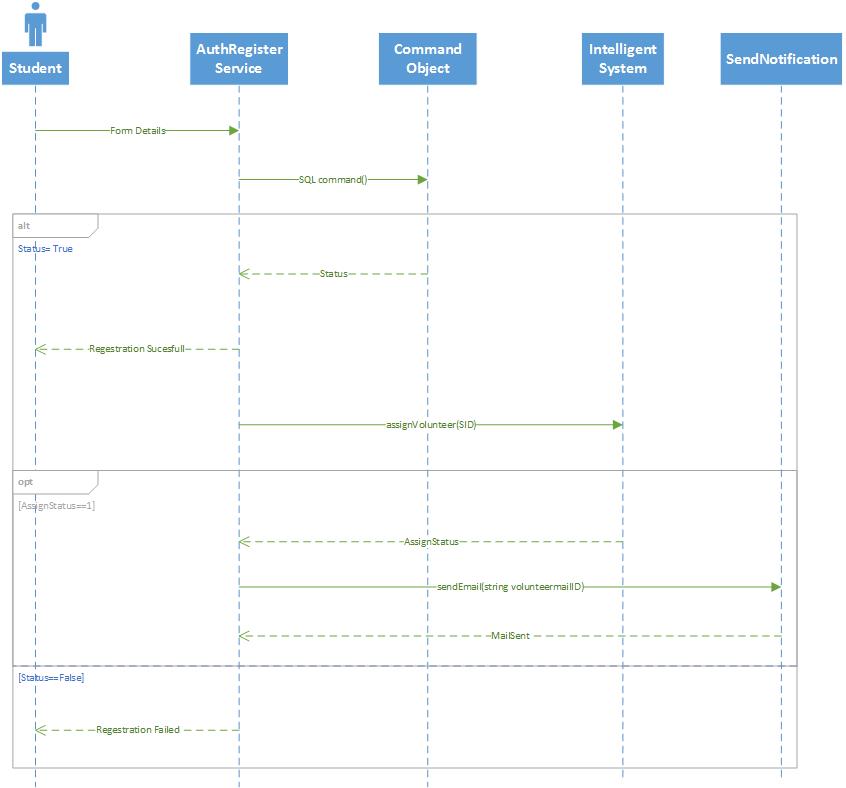
**Step6:** Return status.

**Sequence diagrams:**

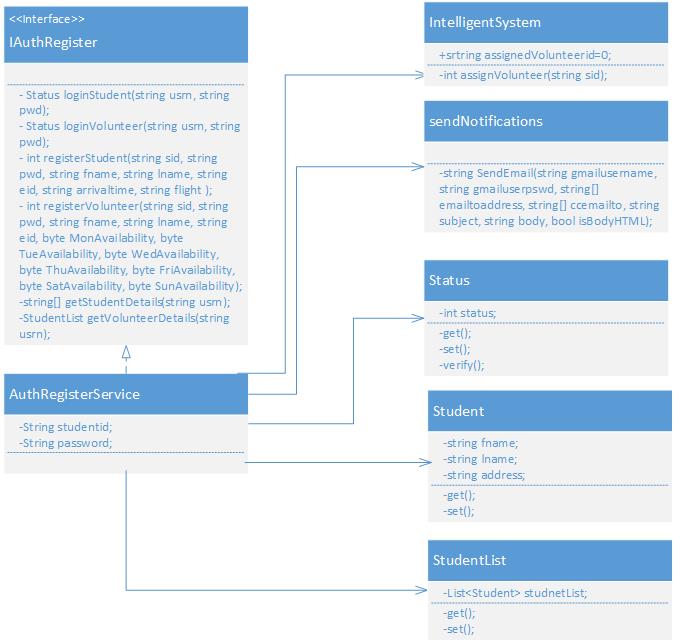
**EmailNotification:**



**Student 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, Volunteer Registration, get student Details and Get Volunteer details functionalities. Intelligent System and notification service are tested by testing Student Registration as that service invokes the call to the intelligent system and intern invokes call to notification.

**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.

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.

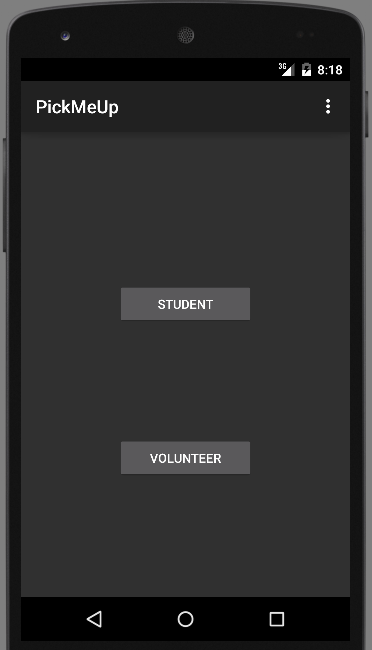
IntelligentSystem service takes the input as the studentID and get the details of student from DB. It will query the Volunteer DB for the Volunteers who are available for the student pickup and assigns him for pickup. From Mobile client side when ever the student updates his flight arrival time this service gets invoked and assigns him volunteer.

Notification service is used by intelligent system to send mail to Volunteer when a student is being assigned. It used the Gmail server to send mail. When Intelligent system assigns student to Volunteer then this notification service is invoked to send mail to volunteer.

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

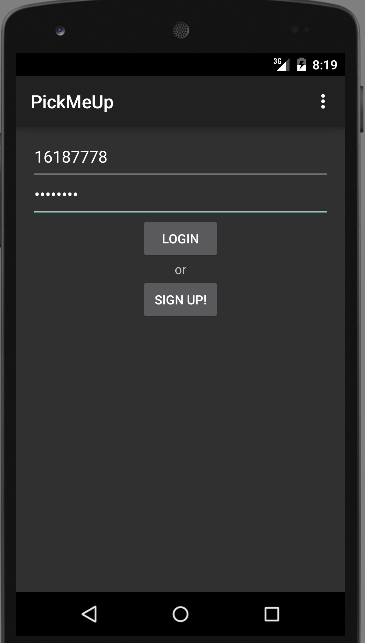
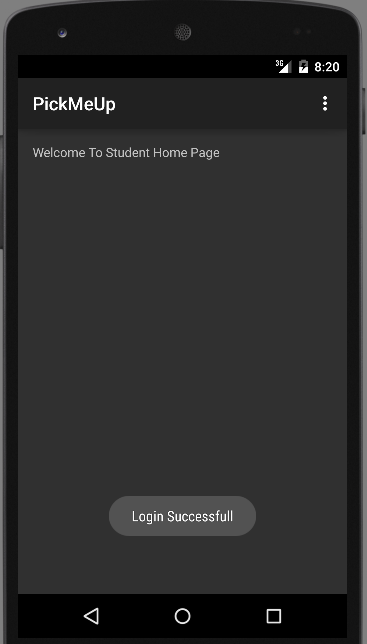
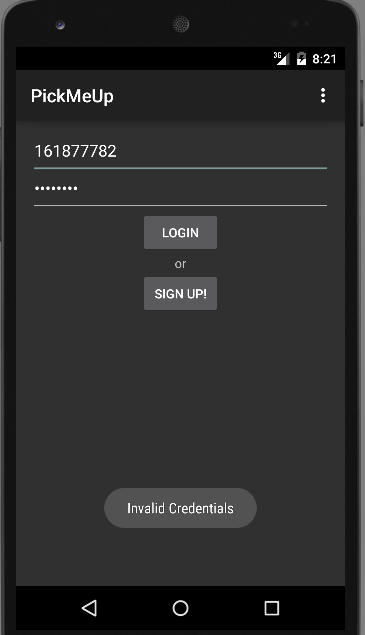
Android studio is being used to implement the Mobile App. User Interface of the app is XML based and is relatively changeable to the screen size. A total of six activities are created for the six screens designed for this increment.

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.

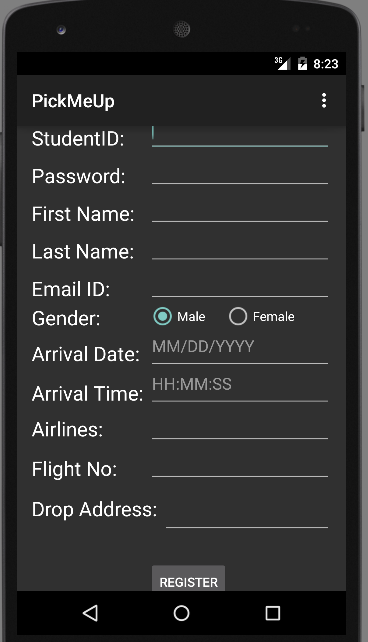
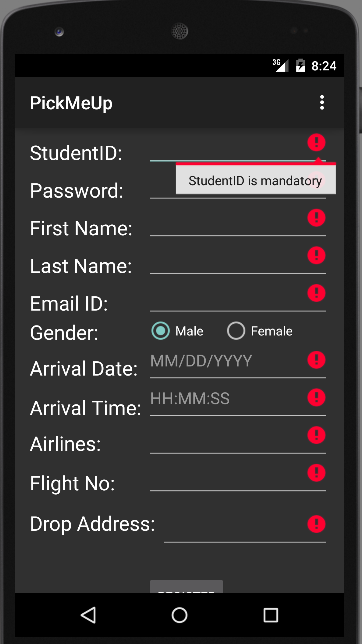
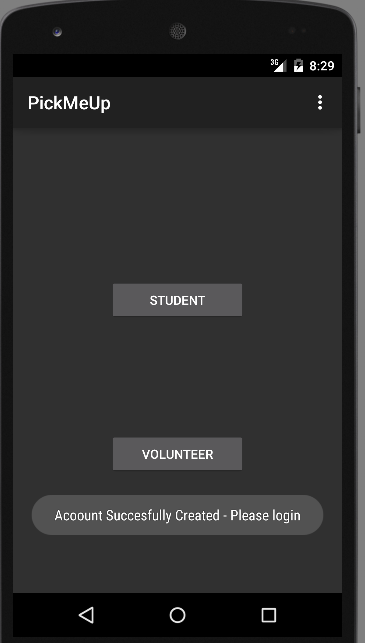


1. **Login Screen:** It is the LoginActivity which consists of StudentID, Password fields and Login, Sign Up buttons. Existing users will navigate to their respective Home screen by the providing the login details. A new private class AuthenticationService is written inside the LoginActivity which uses 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.

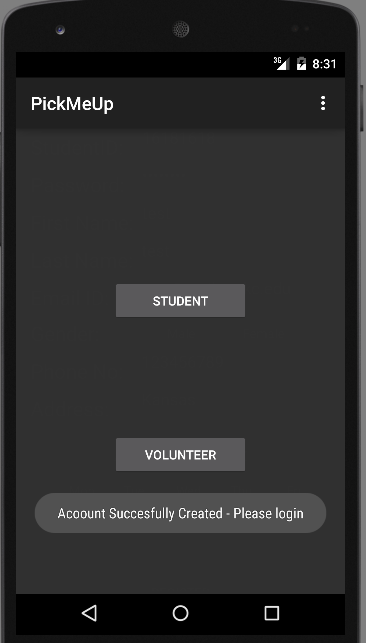
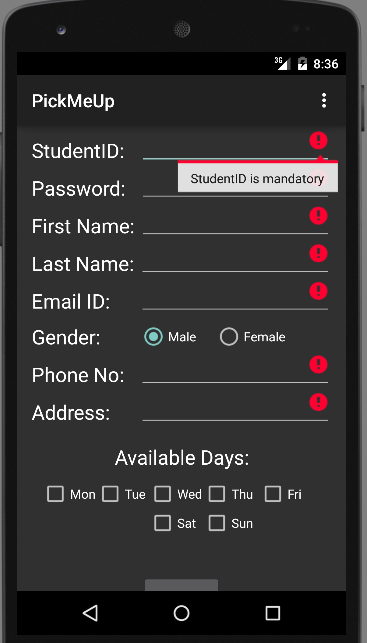
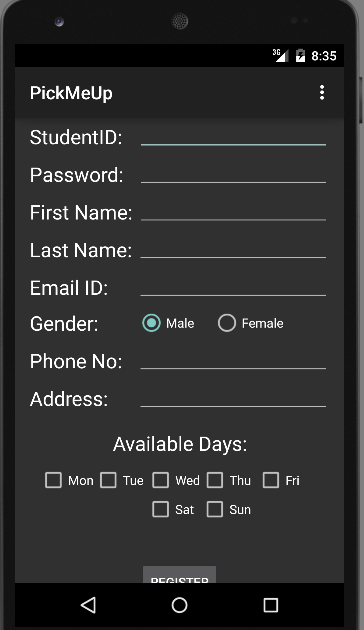
Upon click of the signup button, users will be navigated to Student Registration screen or Volunteer registration screen respectively.

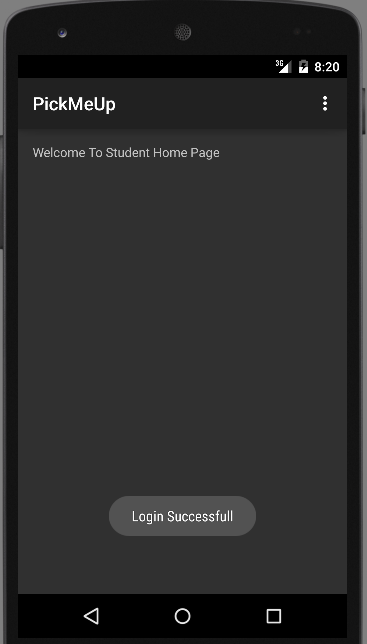
1. **Student Registration Screen:** It is the StudentRegActivity which consists of fields like StudentID, Password, FirstName, LastName, Email ID, Gender, Arrival Date, Arrival Time, Airlines, Flight No and Drop Address to create a student account. Here the studentID field is numeric type, password is 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.

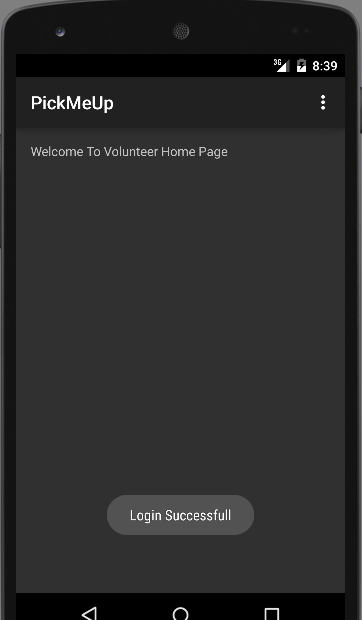
1. **Volunteer Registration Screen:** It is the VolunteerRegActivity which consists of fields like StudentID, Password, FirstName, LastName, Email ID, Gender, Phone No, Address and Available Days fields to create a volunteer account. Here the studentID field is numeric type, password is Password type, 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.

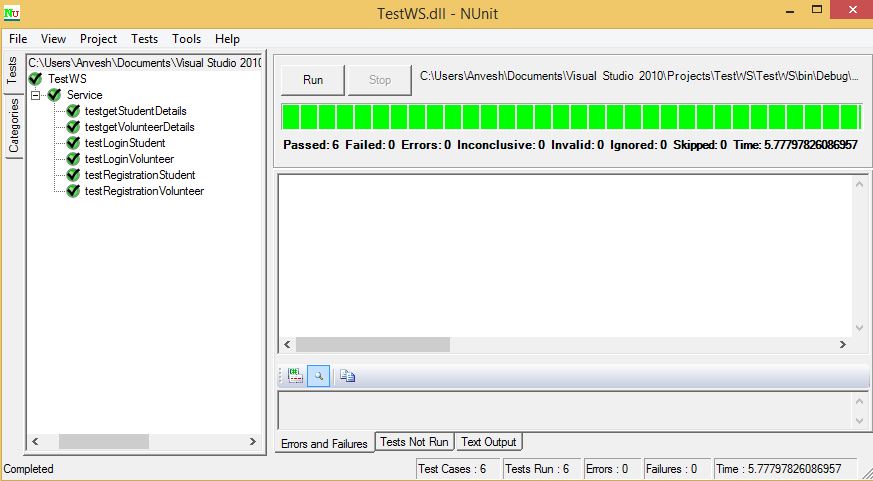


**Implementation of test cases:**

Test cases are implemented by using visual studio and executed by NUnit Client. Our Nuint class Service consists of six test methods, testLoginStudent- to test the student Authentication process, testLoginVolunteer - to test the Volunteer Authentication process, testRegistrationStudent – to test the student Registration process that successful assignment of the Volunteer for the student will return 1, testRegistrationVolunteer - to test the volunteer Registration Process, testgetStudentDetails to test the details of the students returned and testgetVolunteerDetails to test the details of the students assigned for Volunteers.

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

All the six test cases are successful when our service test in TestWS class is executed with NUnit Client.



**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