**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 in their home page.

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

**Detail Design of Services**

**User Stories:**

We have five stories in iteration2

1. As a volunteer, I want to receive notifications when a new student is assigned to me for the pick up.
2. As a volunteer, I want to see all the students who are assigned to me for the pickup with navigation functionality to their respective address.
3. As a student, I want an volunteer to be assigned to me when I am registered for the pickup service.
4. As a student, I want to see my personal details and current weather report of the destination location in my home page.

**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.

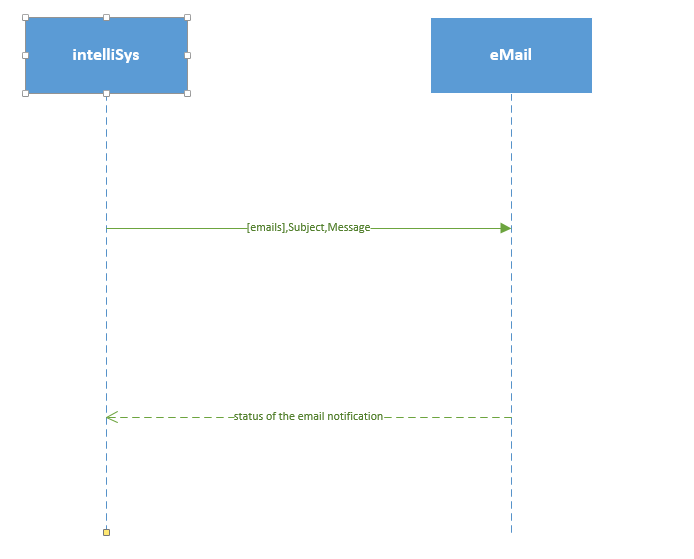
Then we will get the Volunteer who is available for the pickup of student having less no of students being assigned.

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

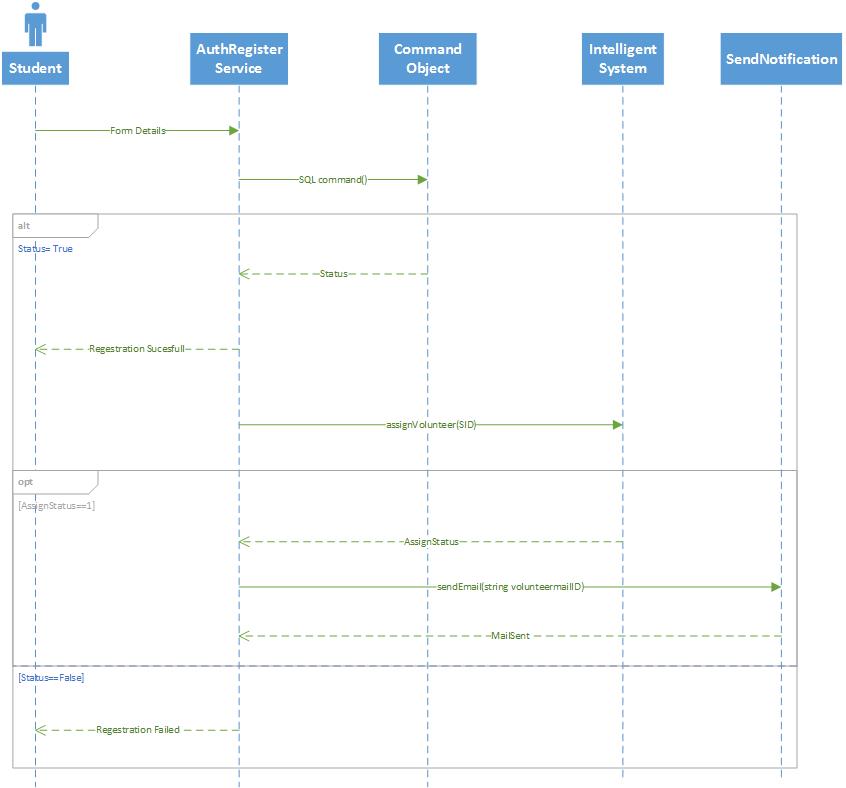
**Step6:** Return status.

**Sequence diagrams:**

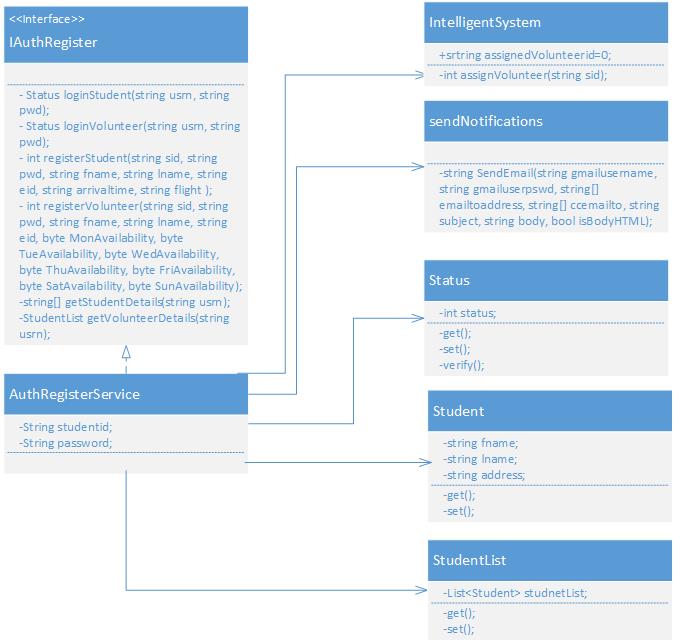
**EmailNotification:**



**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 second, we have designed 2 screens.

1. Student home screen after successful login
2. 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). Student Home Page will consist of the necessary student details and the current weather report of the destination location. Volunteer home page consists of primary details of the assigned students and the google navigation functionality to navigate to their respective dropping addresses.

**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 whenever 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.

Student Home page service which is used to retrieve student details like first name, last name, email id, airlines, flight No based on the username of the student and pass the respective details back to display in the student home page. By taking the username of the student as the input, it will query the Student DB and gets the respective details.

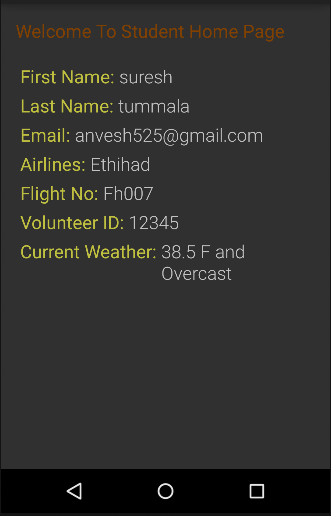
Volunteer Home page service which is used to retrieve primary details like first name, last name and dropping address of all the students who are assigned to the respective volunteer based on the username of the volunteer and display them in the home page of the volunteer. By taking the username of the volunteer as the input, it will query the Student DB and gets the necessary details of all the students assigned to that 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.

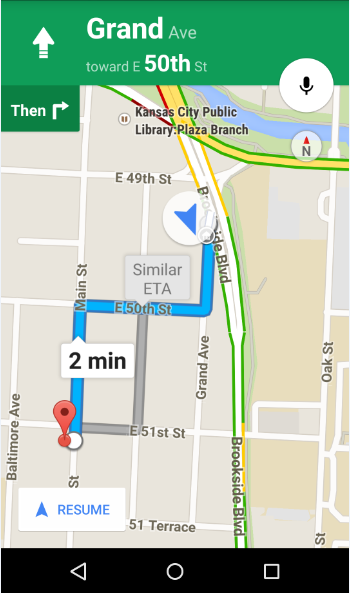
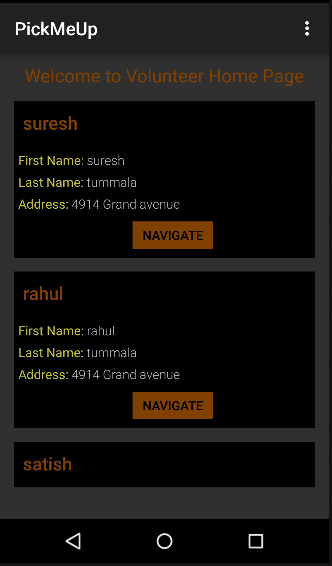
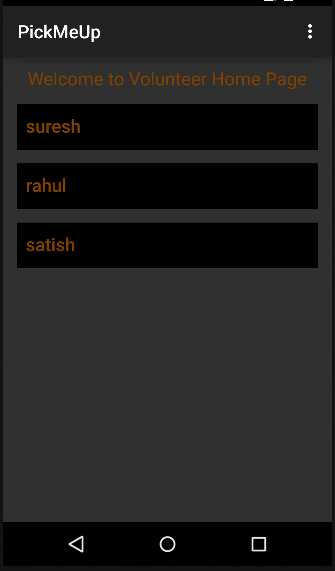
1. **Student Home Screen:** It is the StudentHomeActivity which consists of simple text message “Welcome to Student Home Page” followed by the personal details of the current student and the ID of the volunteer assigned to him and the current weather report of respective location. Upon successful login of the student, student home rest service will be called and of the response will be captured and displayed in the student home screen.

Along with student home service, wunderground external webservice is also called to get the weather report of the current location.



1. **Volunteer Home Screen:** It is the VolunteerHomeActivity which consists of simple text message “Welcome to Volunteer Home Page” followed by the collapsible layout with the names of the students assigned to that volunteer. After successful login of volunteer, Volunteer Home service is called and response of the student details will be captured and displayed. Once any of the student screen is the expanded it will show the first name, last name and dropping address of the student followed by navigate button.

After clicking on the navigate button, google navigation service will be called in a driving mode from the current location to the dropping address of the respective student.

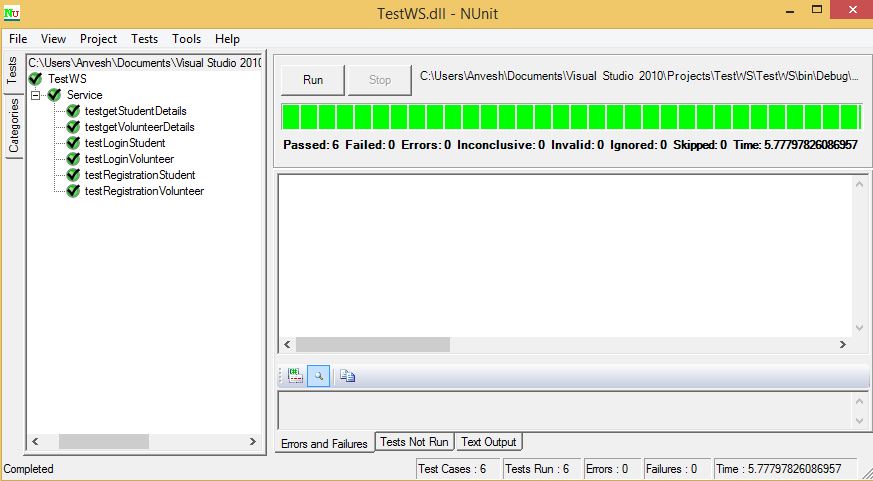


**Implementation of test cases:**

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>

**Webservice URLS:**

Get Student details:

<http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group6/PickMeUpService/AuthService.svc/get/student/{usrn}>

Get Volunteer details:

[http://kc-sce-cs551.kc.umkc.edu/aspnet\_client/Group6/PickMeUpService/AuthService.svc/get/volunteer/{usrn}](http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group6/PickMeUpService/AuthService.svc/get/volunteer/%7busrn%7d)

**Implementation status report:**

**Work Completed:**

**Description:**

1. As a volunteer, I want to receive notifications when a new student is assigned to me for the pick up.

Responsibility: Rahul

Time Taken: 20 hrs

Contribution: 100%

1. As a volunteer, I want to see all the students who are assigned to me for the pickup with navigation functionality to their respective address.

Responsibility: Satish

Time Taken: 20 hrs

Contribution: 100%

1. As a student, I want an volunteer to be assigned to me when I am registered for the pickup service.

Responsibility: Anvesh

Time Taken: 20 hrs

Contribution: 100%

1. As a student, I want to see my personal details and current weather report of the destination location in my home page.

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