**CS5551 Advanced Software Engineering**

**Iteration I**

**Due Date: Mar. 12 (T) 2013, Midnight**

Team Members:

Geetalakshmi Javvadi

Jinal Patel

Supraja Raghavan

**Detail Design of Services - Iteration 1:**

**Database Schema:**

For our first iteration we have designed the Database Schema and implemented the Database tables. Similarly we have designed and implemented the Login and Registration Web services. For this iteration we will create the necessary backend processing. We have created the necessary, user, and group tables that will be used. We will populate these tables with the required data.

**Web Services – Registration:**

We have also designed and developed the Login and the Registration web services. The user who would like to have access to our accent web service will have to register for the service. The registration will have the details of the user’s username, password, name and email id registration details.

**Web Services – Login:**

The Login page will ask the registered user to enter their login username and password. The user’s authenticity will be verified and validated. A validated user can then access our mobile application. For this iteration we have tested our web services using the NUNIT tool.

**The various services and their design are as described below:**

**Registration:**

The design details of the Registration web service are as described below. In this web service the user registers for access to the accent mobile application. They provide the required details such as the username, password and group.etc. The users are thus registered to the system after which they can login to the system. The class diagram, sequence diagram and the service specification are as shown below.

**Class Diagram:**



**Service Specifications for Registration:**

* Description: Register new user
* Input: User information
* Output: Confirmation
* Precondition: User must be interested in the accent mobile application
* Post condition: User account is created
* Actors involved: User, IIS Server and Database.
* Object involved: User, Group, DBInterface, RegisterWebService
* Constraint/Exception: Username does not exist

**WSDL Definition:**

<wsdl:definitions targetNamespace="http://tempuri.org/"><wsdl:types><s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/"><s:element name="SHRegistration"><s:complexType><s:sequence><s:element minOccurs="0" maxOccurs="1" name="Username" type="s:string"/><s:element minOccurs="0" maxOccurs="1" name="Password" type="s:string"/><s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string"/><s:element minOccurs="0" maxOccurs="1" name="Emailid" type="s:string"/></s:sequence></s:complexType></s:element><s:element name="SHRegistrationResponse"><s:complexType><s:sequence><s:element minOccurs="0" maxOccurs="1" name="SHRegistrationResult" type="s:string"/></s:sequence></s:complexType></s:element></s:schema></wsdl:types><wsdl:message name="SHRegistrationSoapIn"><wsdl:part name="parameters" element="tns:SHRegistration"/></wsdl:message><wsdl:message name="SHRegistrationSoapOut"><wsdl:part name="parameters" element="tns:SHRegistrationResponse"/></wsdl:message><wsdl:portType name="RegistrationSoap"><wsdl:operation name="SHRegistration"><wsdl:input message="tns:SHRegistrationSoapIn"/><wsdl:output message="tns:SHRegistrationSoapOut"/></wsdl:operation></wsdl:portType><wsdl:binding name="RegistrationSoap" type="tns:RegistrationSoap"><soap:binding transport="http://schemas.xmlsoap.org/soap/http"/><wsdl:operation name="SHRegistration"><soap:operation soapAction="http://tempuri.org/SHRegistration" style="document"/><wsdl:input><soap:body use="literal"/></wsdl:input><wsdl:output><soap:body use="literal"/></wsdl:output></wsdl:operation></wsdl:binding><wsdl:binding name="RegistrationSoap12" type="tns:RegistrationSoap"><soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/><wsdl:operation name="SHRegistration"><soap12:operation soapAction="http://tempuri.org/SHRegistration" style="document"/><wsdl:input><soap12:body use="literal"/></wsdl:input><wsdl:output><soap12:body use="literal"/></wsdl:output></wsdl:operation></wsdl:binding><wsdl:service name="Registration"><wsdl:port name="RegistrationSoap" binding="tns:RegistrationSoap"><soap:address location="http://localhost/aspnet\_client/SpeechHelperLogin/Registration.asmx"/></wsdl:port><wsdl:port name="RegistrationSoap12" binding="tns:RegistrationSoap12"><soap12:address location="http://localhost/aspnet\_client/SpeechHelperLogin/Registration.asmx"/></wsdl:port></wsdl:service></wsdl:definitions>

**Sequence Diagram:**

****

**Login:**

The design details of the Login web service are as described below. After a user has registered for the mobile application, the user needs to Login to the application in order to access their group and the word list for their accent group. The user is authenticated and verified. If the user enters an authentic username and password he is allowed access to the application otherwise the access is denied. The class diagram, sequence diagram and the service specification are as shown below.

**Service Specifications for Login:**

* Description: Authenticate user
* Input: Username, password
* Output: authenticate pass
* Precondition: User has been registered
* Post condition: User is authenticated
* Actors involved: User, IIS Server, database
* Object involved: User, Group, DBInterface, LoginWebService
* Constraint/Exception: Username and password are correct

**WSDL Definition:**

<wsdl:definitions targetNamespace="http://tempuri.org/"><wsdl:types><s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/"><s:element name="SHLogin"><s:complexType><s:sequence><s:element minOccurs="0" maxOccurs="1" name="username" type="s:string"/><s:element minOccurs="0" maxOccurs="1" name="password" type="s:string"/></s:sequence></s:complexType></s:element><s:element name="SHLoginResponse"><s:complexType><s:sequence><s:element minOccurs="0" maxOccurs="1" name="SHLoginResult" type="s:string"/></s:sequence></s:complexType></s:element></s:schema></wsdl:types><wsdl:message name="SHLoginSoapIn"><wsdl:part name="parameters" element="tns:SHLogin"/></wsdl:message><wsdl:message name="SHLoginSoapOut"><wsdl:part name="parameters" element="tns:SHLoginResponse"/></wsdl:message><wsdl:portType name="LoginSoap"><wsdl:operation name="SHLogin"><wsdl:documentation>Aunthitication</wsdl:documentation><wsdl:input message="tns:SHLoginSoapIn"/><wsdl:output message="tns:SHLoginSoapOut"/></wsdl:operation></wsdl:portType><wsdl:binding name="LoginSoap" type="tns:LoginSoap"><soap:binding transport="http://schemas.xmlsoap.org/soap/http"/><wsdl:operation name="SHLogin"><soap:operation soapAction="http://tempuri.org/SHLogin" style="document"/><wsdl:input><soap:body use="literal"/></wsdl:input><wsdl:output><soap:body use="literal"/></wsdl:output></wsdl:operation></wsdl:binding><wsdl:binding name="LoginSoap12" type="tns:LoginSoap"><soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/><wsdl:operation name="SHLogin"><soap12:operation soapAction="http://tempuri.org/SHLogin" style="document"/><wsdl:input><soap12:body use="literal"/></wsdl:input><wsdl:output><soap12:body use="literal"/></wsdl:output></wsdl:operation></wsdl:binding><wsdl:service name="Login"><wsdl:port name="LoginSoap" binding="tns:LoginSoap"><soap:address location="http://localhost/aspnet\_client/SpeechHelperLogin/Login.asmx"/></wsdl:port><wsdl:port name="LoginSoap12" binding="tns:LoginSoap12"><soap12:address location="http://localhost/aspnet\_client/SpeechHelperLogin/Login.asmx"/></wsdl:port></wsdl:service></wsdl:definitions>

**Class Diagram:**



The sequence diagram below gives the sequence diagram for a user login. The diagram is for the user’s first time login and also for their subsequent login. For the first time login the sequence for RecentlyVieweWordList call will be not available.

**Sequence Diagram:**



**Design of Unit Test Cases:**

We use NUnit to design and test our code for the web services. We have created test cases that test the proper login and registration. For the registration we test our code by inserting the new user into the database. We also test to see if when a user name exists the user is not allowed to create a new account and is prompted to create a different username.

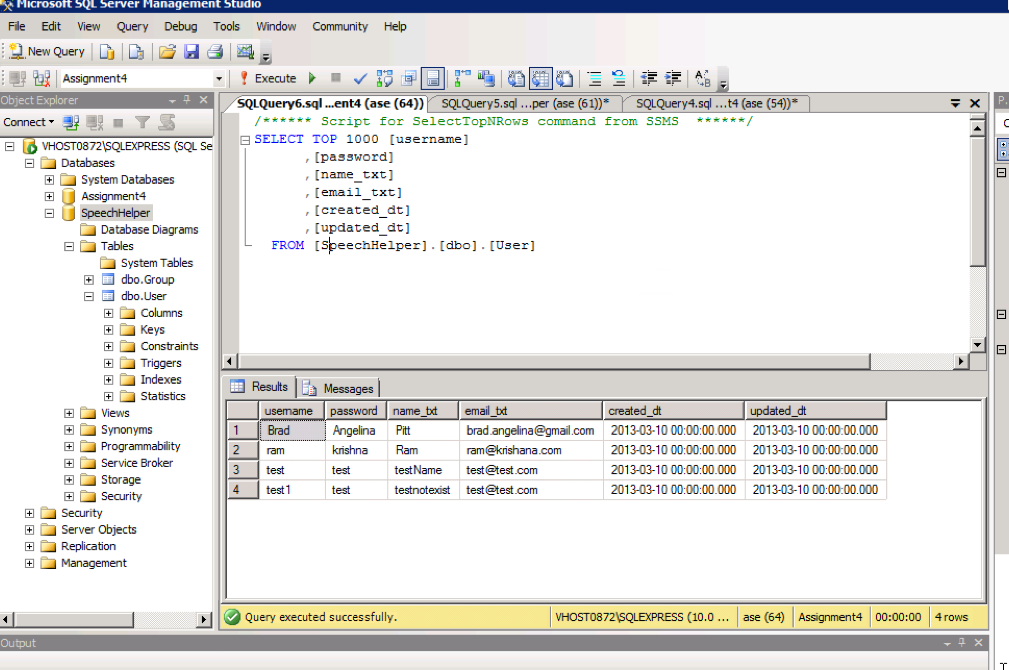
Similarly for the Login, we test the web service by creating test cases that check for a user’s username and password and authenticate and authorize the user. We also created test cases to check for wrong login details.

**Implementation**

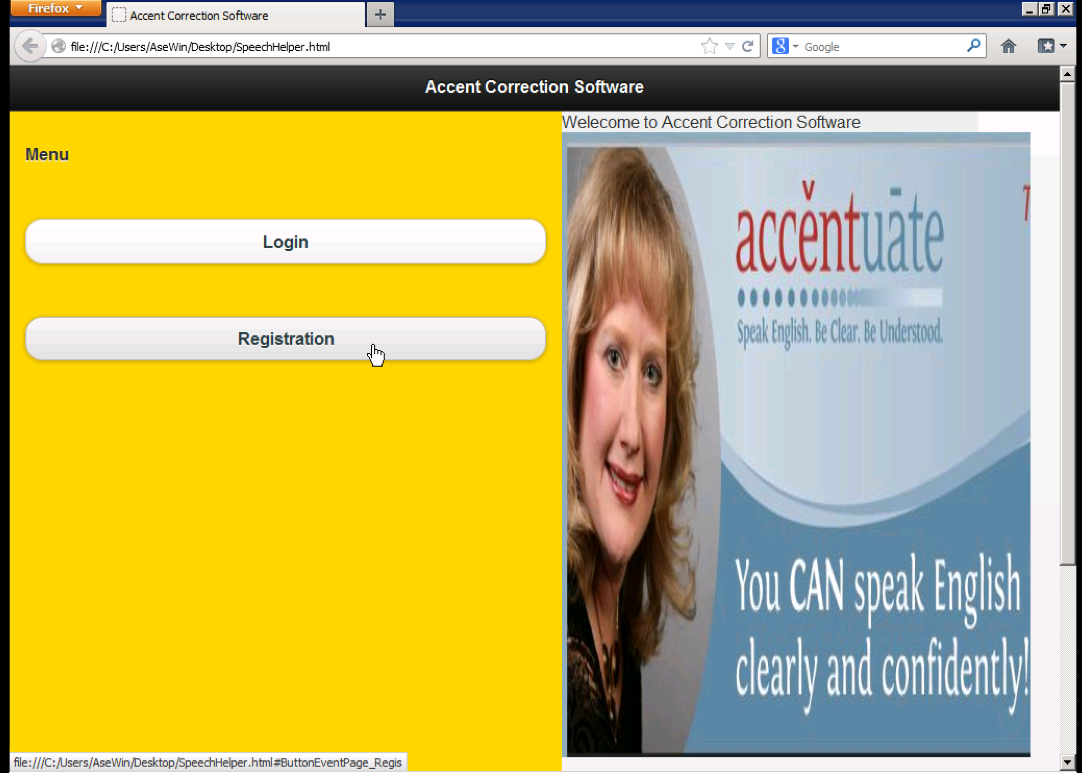
We have created the required database tables and have implemented the web service for user registration and Login. For this iteration we have created a web page using HTML5 that allows a new user to register for the Speech Helper software and also allows an existing user to Login to the application. We have implemented the database and the tables to allow users to register and login. We have implemented and tested the code for the web service using NUnit.

**The various Screen Shots for the database, HTML page and the web services are attached below:**

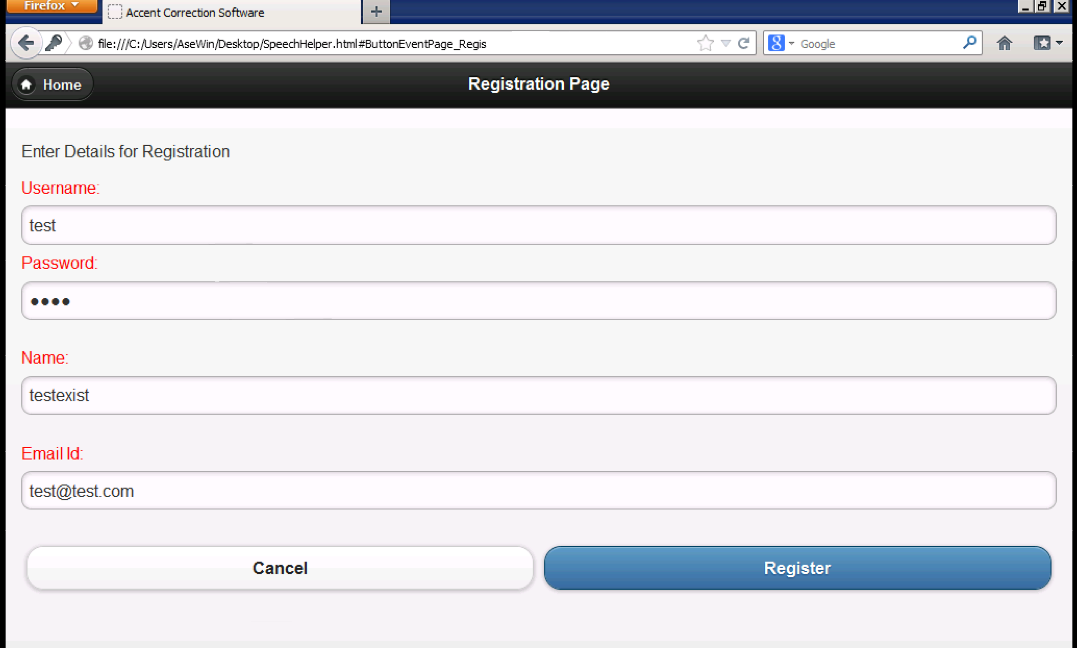
**Screenshot of the Database and the tables:**

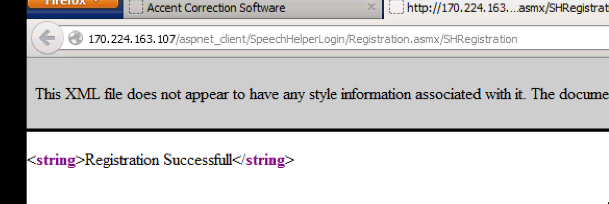
****

**Screen shot of the HTML Page:**

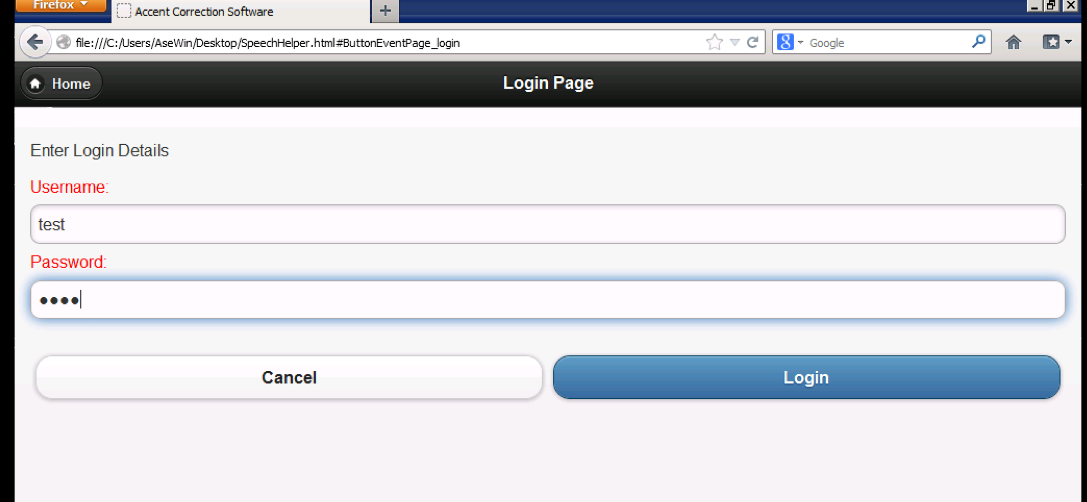
****

**Screen Shot for Registration and Registration Confirmation:**

****

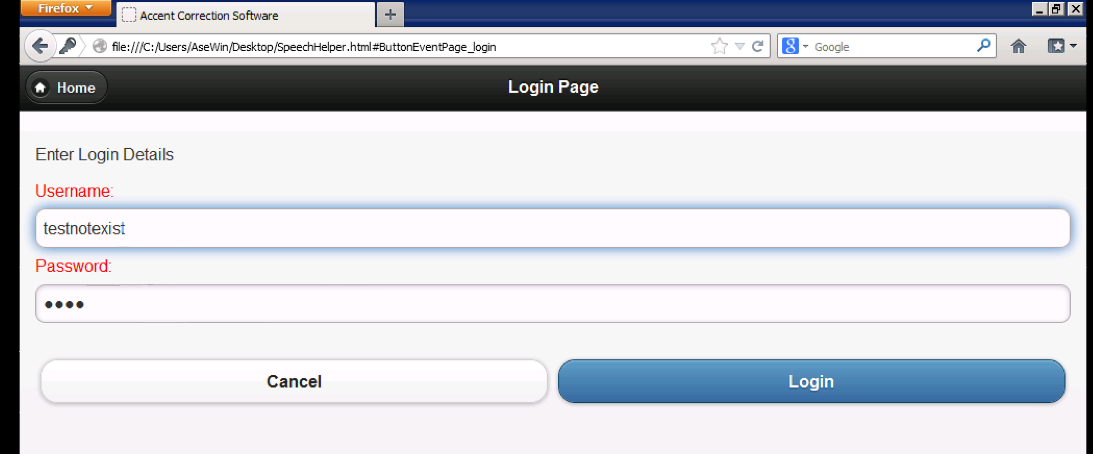
****

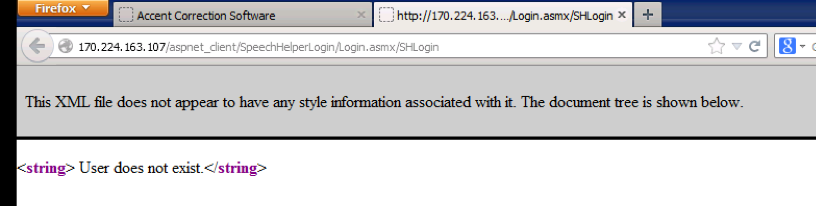
**Screen shot for Login – Success:**

****

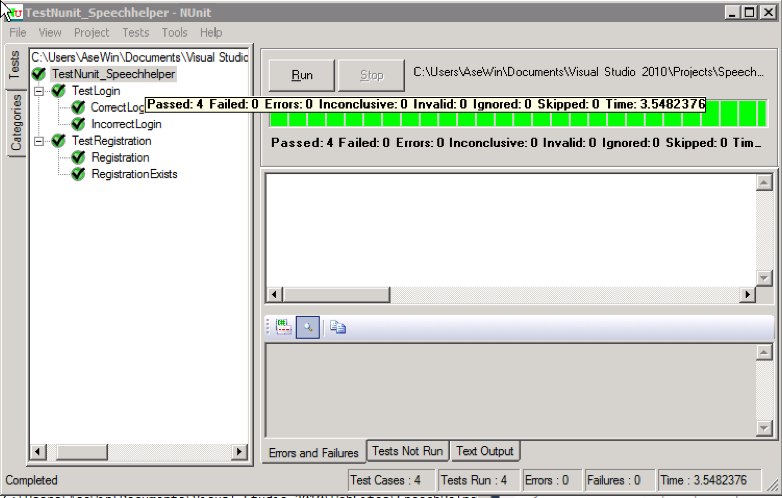
****

**Screen shot for Login – Failure:**

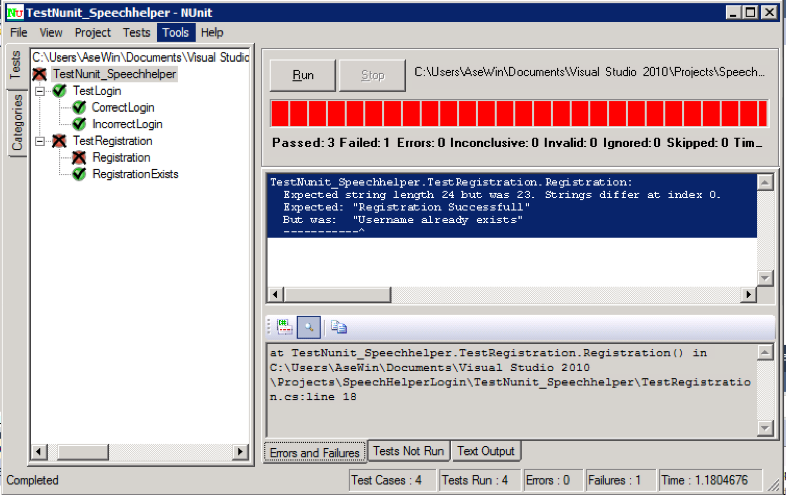
****

****

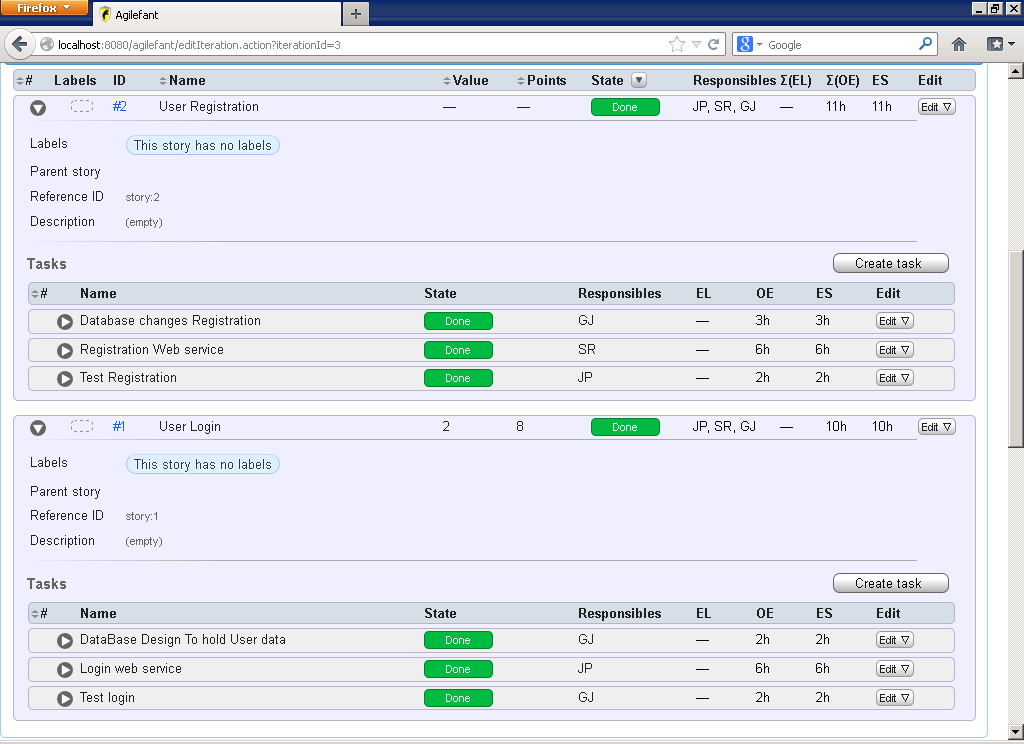
**Screen shot of NUnit – Test Pass:**

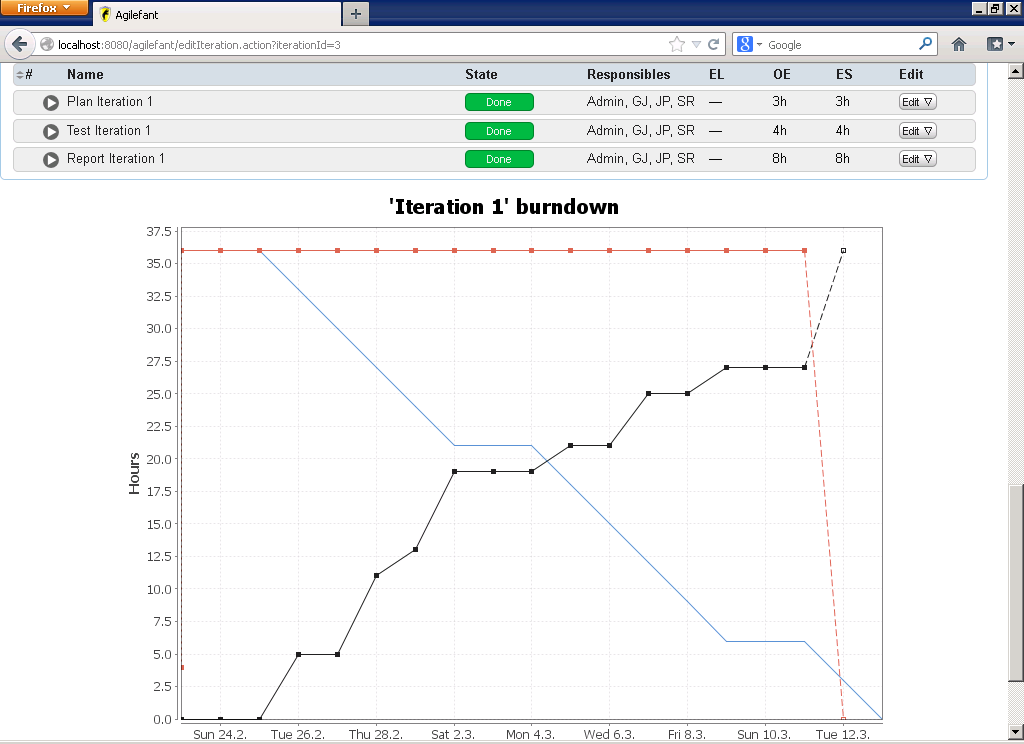
****

**Screen shot of NUnit – Test Fail:**

****

**Screen Shots of Agilefant:**





**Internet Access:**

* 1. Project Website (URL):

We have a website for the project on the cloud. The URL of the link is as given below.

<http://170.224.163.107/aspnet_client/website2/CS.aspx>

* 1. GitHub (URL):

We use GitHub for development version control. We have a GitHub account for the same. The URL is: <https://github.com/ASE-Javvadi-Patel-Raghavan/PronunciationHelper.git>