Software Requirement Specification

Version No : 1.0

Date : 15 February 2011

Project Name : Practice Labs Social Network

Project Code :

Copyright Notice

The contents of this document are proprietary to HCL and shall not be disclosed, disseminated, copied, or used except for purposes expressly authorized in writing by HCL

Revision History

| Version No | Date | Prepared by / Modified by | Significant Changes |
| --- | --- | --- | --- |
| 1.0 | 15-feb-2011 | Santhosh Kumar Gandi | Purpose, Scope, Product Perspective were done |
| 1.0 | 21-feb-2011 | Sravani Thota | Added Overview and other parts of system context |
| 1.0 | 22-Feb-2011 | Sravani Thota | Added Specific Requirements to some extent |
| 1.0 | 23-Feb-2011 | Sravani Thota | Added remaining part of Specific Requirements |
| 1.0 | 24-Feb-2011 | Sravani Thota | Added Derived Requirements, Validation |
| 1.0 | 10-Mar-2011 | Sravani Thota | Added 4 new use-cases |

Glossary

|  |  |
| --- | --- |
| Abbreviation | Description |
|  |  |
|  |  |
|  |  |

**Table Of Contents**

[1 Introduction](#_Toc275360485) 5

[1.1 Purpose 5](#_Toc275360486)

[1.2 Scope 5](#_Toc275360487)

[2 Overview of the Document 5](#_Toc275360488)

[3 System Context 6](#_Toc275360489)

[3.1 Product Perspective 6](#_Toc275360490)

[3.1.1 System Overview & Context Diagram 6](#_Toc275360491)

[3.1.2 Operational Concepts and Scenarios 6](#_Toc275360492)

[3.1.3 System Interface 6](#_Toc275360493)

[3.1.4 Memory Constraints 6](#_Toc275360494)

[3.1.5 Operations 6](#_Toc275360495)

[3.1.6 Site Adaptation Requirements 6](#_Toc275360496)

[3.2 Product Functions 9](#_Toc275360497)

[3.3 Constraints 10](#_Toc275360498)

[3.4 Assumptions and Dependencies 10](#_Toc275360499)

[3.5 Apportioning of Requirements 10](#_Toc275360500)

[4 Specific Requirements 11](#_Toc275360501)

[4.1 Organizational Requirements 11](#_Toc275360502)

[4.2 Business requirements 11](#_Toc275360503)

[4.3 User Requirements 11](#_Toc275360504)

[4.4 External Interface Requirements 12](#_Toc275360505)

[4.5 Functional Requirements / System Features 18](#_Toc275360506)

[4.5.1 Use-case For Login 18](#_Toc275360491)

[4.5.2 Use-case For Registration 20](#_Toc275360491)

[4.5.3 Use-case For User Details 22](#_Toc275360491)

[4.5.4 Use-case For Search Result 24](#_Toc275360491)

[4.5.5 Use-case For SendFriendRequests 25](#_Toc275360491)

[4.5.6 Use-case For Accept/Deny Friend Requests 26](#_Toc275360491)

[4.5.7 Use-case For Verify Friends Added 28](#_Toc275360491)

[4.5.8 Use-case For Admin Login windows form 29](#_Toc275360491)

[4.5.9 Use-case For Admin Profile windows form 30](#_Toc275360491)

[4.5.10 Use-case For deleting a user from datagrid 31](#_Toc275360491)

[4.5.11 Use-case For viewing current list of users from database 32](#_Toc275360491)

[4.6 Performance Requirements 33](#_Toc275360507)

[4.7 Acceptance Criteria 33](#_Toc275360508)

[4.8 Logical Database Requirements 34](#_Toc275360509)

[4.9 Design Constraints 38](#_Toc275360510)

[4.10 Testing Requirements 38](#_Toc275360511)

[4.11 Compliance to Standards 38](#_Toc275360512)

[4.12 Software System Attributes 38](#_Toc275360513)

[4.13 Statutory and Regulatory Requirements 42](#_Toc275360514)

[4.14 Organizing Specific Requirements 43](#_Toc275360515)

[4.14.1 System Mode 43](#_Toc275360516)

[4.14.2 User Class 43](#_Toc275360517)

[4.14.3 Objects 43](#_Toc275360518)

[4.14.4 Features 43](#_Toc275360519)

[4.14.5 Implementation schedule (Optional) 43](#_Toc275360520)

[4.14.6 Acceptance 43](#_Toc275360521)

[5 Derived Requirements 44](#_Toc275360522)

[6 Validation 44](#_Toc275360523)

[6.1 Validation strategy 44](#_Toc275360524)

[6.2 Validation criteria 44](#_Toc275360525)

[6.3 Validation Constraints 45](#_Toc275360526)

[7 References 45](#_Toc275360527)

# Introduction

## Purpose

The purpose of this document is to inform the reader significant details and

back-ground of our project

This document will explain the purpose and features of the product, the interfaces, what the product will do, the constraints under which it must operate and how the product system will react to external stimuli. This document is intended for HCL Technologies Trainers, Trainees and Developers of the product. Throughout this document the term ‘product’ is used to indicate Practise-Labs Social Networking Application.

## Scope

This product is used in all HCL Technologies (India) training centers. This product is intentionally bugged, where trainees will debug this product in their practice labs at later stage.

# Overview of the Document

This document describes the social networking web site functionality. With this social networking web application you can navigate through a web-site where a new user can register with the web-site.

If the user is already registered then he/she can log in to the site, can view his/her profile, can search for his/her friends, can send friend request for any user, can receive friend requests from other users, can either accept/deny the friend requests received, can also chat with his/her friends by posting the scraps.

# System Context

## Product Perspective

The product is a Practice-Labs Web application which will be used as a training material for future trainees in HCL. This is a regular social networking application like orkut, face-book, my-space, etc., but not as vast as a those real time applications. Minimum common functionalities such as, a web user can register with the site, create his/her profile, search for friends in the network, send them friend requests, accept request and add him/her as a friend or deny requests, send messages, visit friend’s profile, post scraps.

### System Overview & Context Diagram

It would be just like that of a real-time website. i.e, “n” number of users accessing the website to perform their required actions

### Operational Concepts and Scenarios

There can be two types of users who will be accessing this application. They are Registered User and Un-registered user.

Un-registered user can view the home page of the website where he/she can access certain features like searching for friends where the name of the friend will be provided. Then a list of users with similar kind of names will be displayed along with the photo of the user. Generally if the user wants to view the profile of his friends he can do so by clicking on the photo of the user. But if the unregistered user wants to view the profile of the friends then he has to register first and log-in to his/her profile. Only then the user can view his friend’s profile.

Registered user is allowed to access all the features that are provided by the application. Once the user is logged in he/she can view his/her profile, can search for his/her friends, can send friend request for any user, can receive friend requests from other users, can either accept/deny the friend requests received, can also chat with his/her friends by posting the scraps.

### System Interface

The System interface is just a browser that allows the user to surf the site by following the website’s URL.

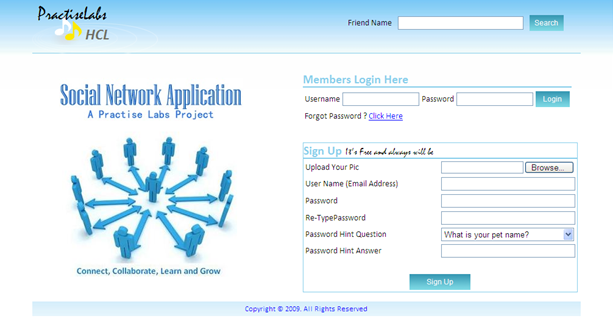
#### User Interface

Using this system is fairly simple and intuitive. A user familiar with basic browser navigation skills should be able to understand all functionality provided by the system.

This Product has two User Interfaces. One is a website for normal users and another is a windows forms application for administrator.

Website is completely designed and developed in ASP.Net and uses C-Sharp as coding language. Two Master Pages are used to maintain unique layout for all web pages. Design aspects such as layout of the page, pictures, header, footer and menu are all coded in a cascading style sheet and it is embedded in to Master Pages.

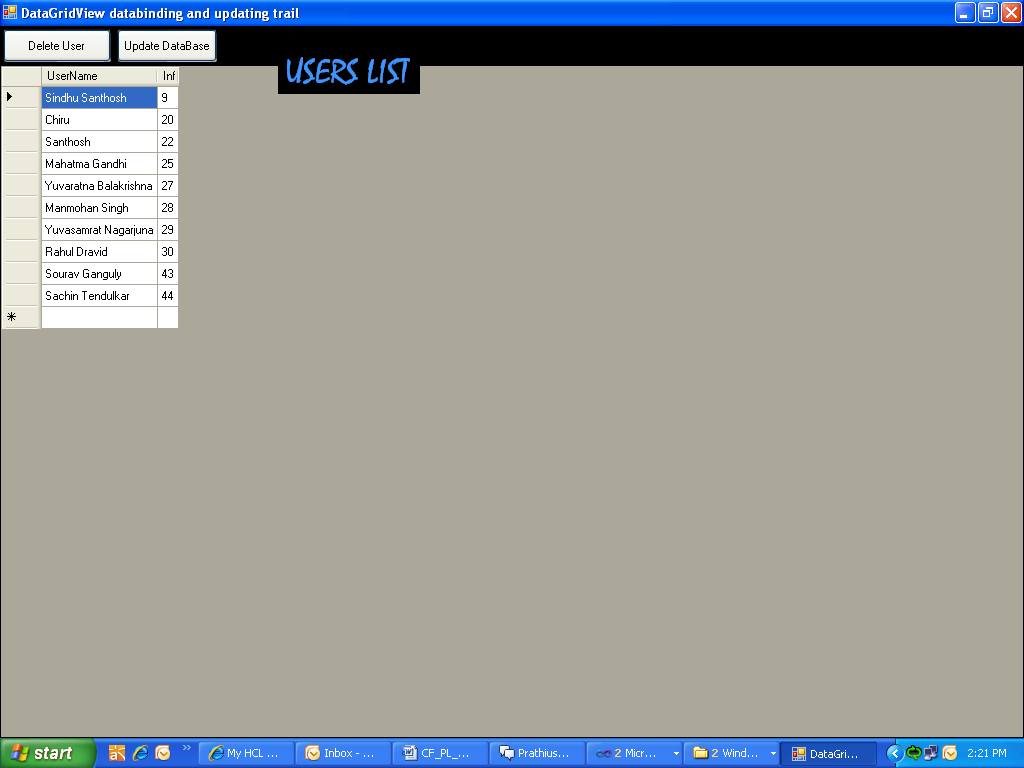
Here is a screen shot of website’s home page



Admin user interface is a windows forms application. Here is a screen shot of that



This is the screen shot of Admin’s profile



#### Hardware Interface

The system should work on most of the home desktop and laptop computers. This is the hardware that this application requires

#### Software Interfaces

The system is designed to run on any version and any type of browser preferably go for Firefox 3 and Internet Explorer 7.

#### Communication Interfaces

The HTTP protocol will be used to facilitate communications between the client and server.

System must have access to the included database. Other components of this Practice Labs application may require access to certain databases and web services; refer to those components documentation for details.

### Memory Constraints

It is not anticipated that Practise Labs Application will require any additional memory storage.

### Operations

The system is limited by its operating server in terms of the maximum number of users it can support at a given time

### Site Adaptation Requirements

No site adaptation is necessary. The product is portable and is self-contained. The entire system is transported to wherever it is needed. No external dependencies are in place and operation of the system will never change due to location.

## Product Functions

This product is developed in-order to support the following functions:

* + Allowing a user to get registered with the portal and thus become a member of the portal
  + Providing the user a search function such that the user can search for other users on the portal and can make friendship
  + A logged-in user can view the profiles of other users on the portal
  + A logged-in user can send friend requests to any user on the portal
  + A logged-in user can either accept/deny the friend requests received
  + Logged-in users can logout from the portal once his/her desired task is finished
  + All the above functionalities are injected with bugs so that this serves as a practice lab for the coming trainees of HCL

## Constraints

The major constraints in this project will be the safety and security constraints which were vital in protecting user’s privacy. They may include the following:

* Restricting anonymous users from accessing the site
* Restricting backward navigation to log-in page once after the user is logged-in
* Restricting backward navigation once after the user is logged-out
* Grant the access permissions for users depending up on their category like registered user and un-registered user.

## Assumptions and Dependencies

Although basic password authentication and other security mechanisms are used to protect this Practise Labs application from un-authorized access, a redundant database is setup as the role of backup Database Server whenever the primary database is failure.

The correct functioning of Practise Labs application will partly be dependent on the correctness of the data stored and managed as part of the database server; the event of the server failing due to an error with one of these applications might result in Practise Labs application becoming temporarily unavailable.

## Apportioning of Requirements

Following are the requirements that may be delayed until future versions of the system because of the project plan and hours available:

* Implementing a “logger” which will log all the actions that were done by the user in to a text document

# Specific Requirements

## Organizational Requirements

The organizational requirements may include the following:-

1. Paying attention to policy pertaining to logo, (trademark or service mark) usage.
2. Discouraging or not allowing any kind of anonymity on the site.

That is, any kind of misuse of the site by the posting of inappropriate content should not be tolerated. Only registered users should be given rights to carry out their functionalities.

1. Posting terms and conditions for the use of the site. Within the terms and conditions, users can also be notified that content will be monitored and may be removed by the moderator.
2. Using separate windows forms application for an Administrator to log on and view all the users on the portal.
3. Unit testing all the functionalities in the application using visual studio test tool.
4. Finally after developing the complete application, the application should be intentionally bugged, where trainees of HCL will debus this application as a part of practice during their training.

## Business requirements

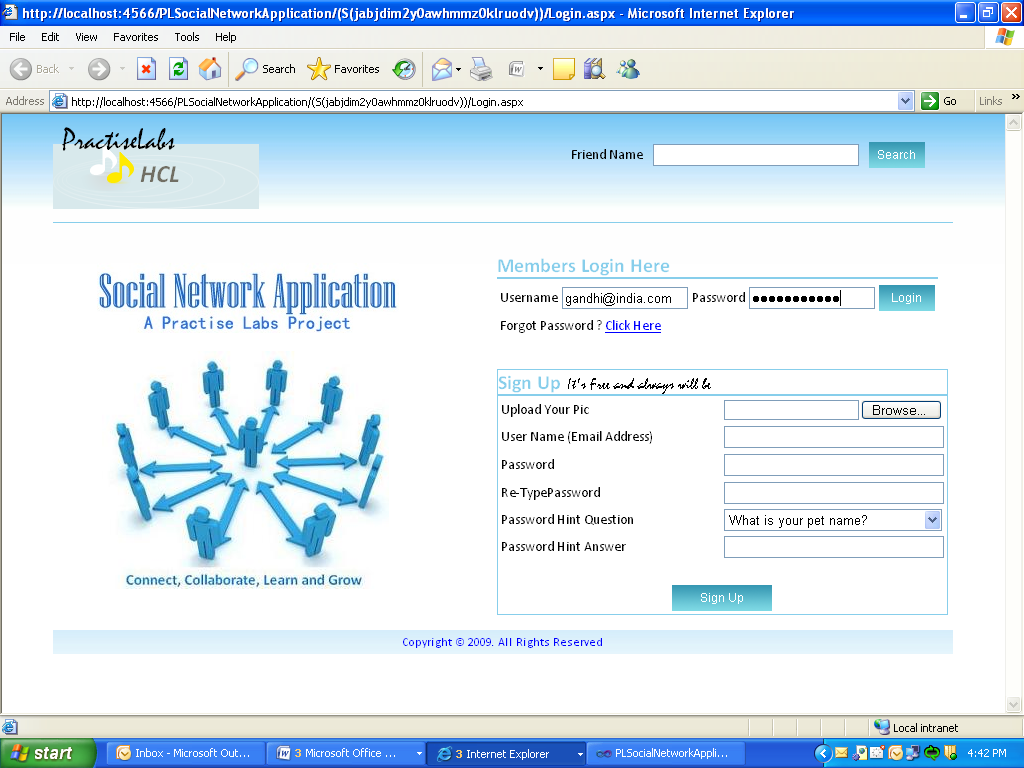
1. Making the site available for 24X7 hours.
2. To maintain compliance. Compliance is set up within an organization to ensure that proper controls are in place to protect the user’s data and comply with regulatory requirements.

## User Requirements

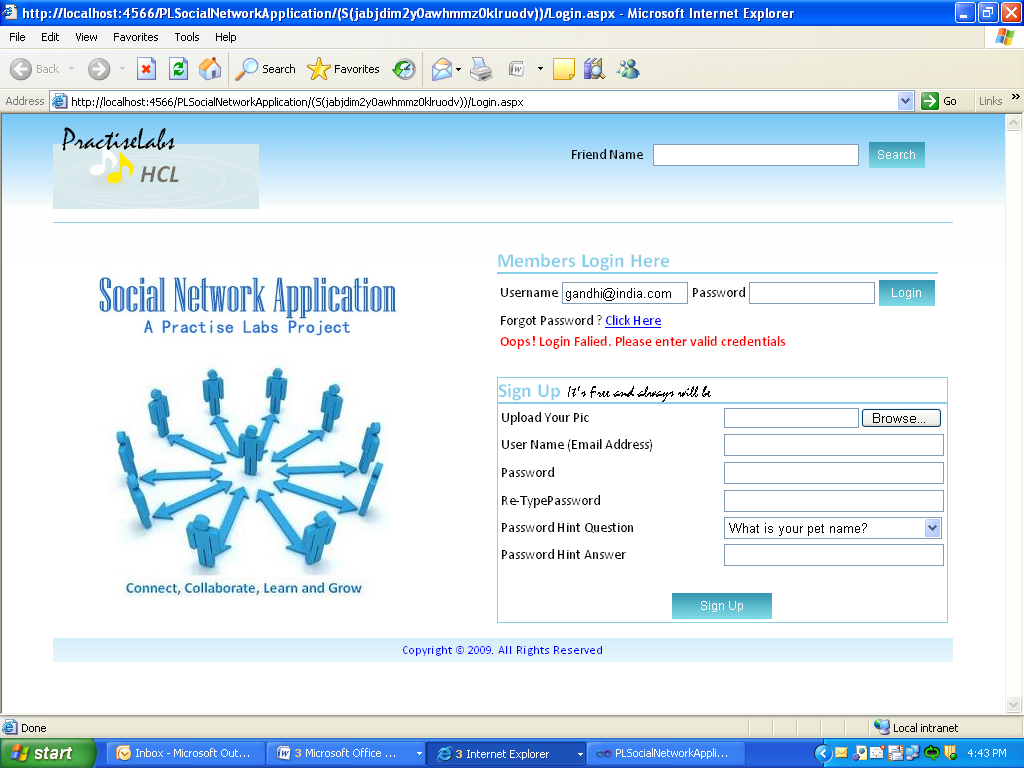
1. Primarily the user should be a registered user.
2. Anonymous users can also view the site but only with certain permissions
3. The user can send friend requests to anyone and can post scraps for any user on the portal once he/she is a registered user.

## External Interface Requirements

Page-1: This is the home page where the user can enter his/her credentials like username and password to log-in to the site.



Page-2: If in-case the user enters an invalid password or username a validation error is displayed as shown in the screen-shot below.



Page-3: If the user enters the valid credentials then the user profile page is displayed as shown in the screen-shot below.



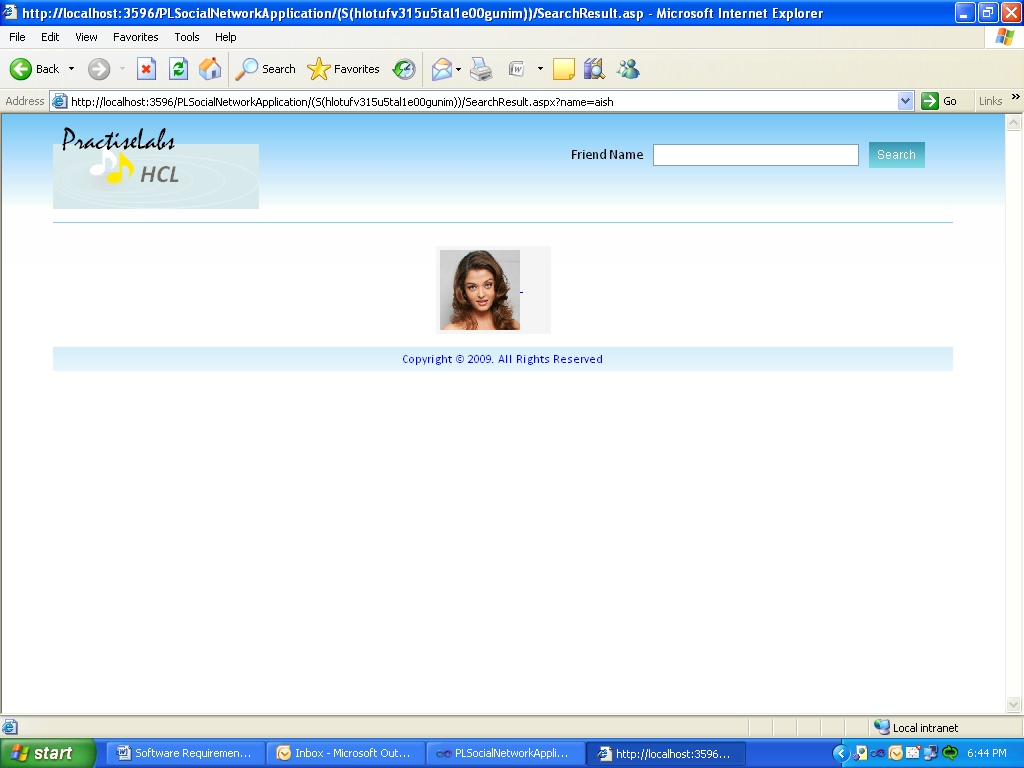
Page-4: If the user clicks on the image of any of his friends shown at the top-right corner then his friend’s profile page is displayed highlighting the “Delete Friend” button as shown below



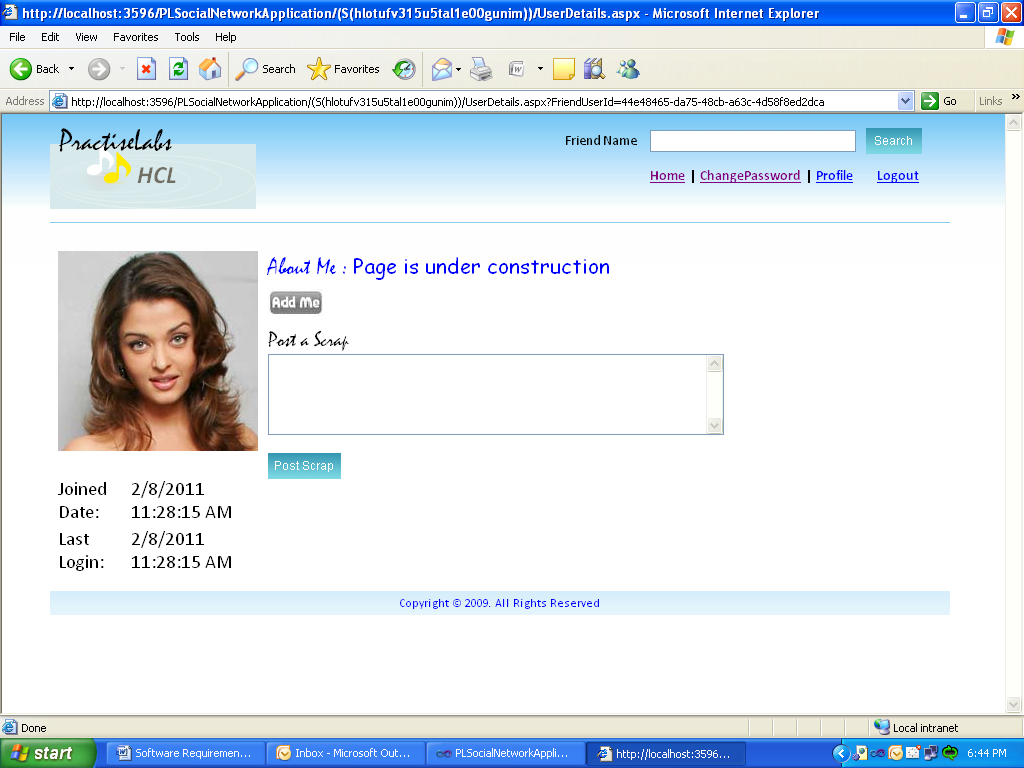
Page-5: If the user wants to search for a friend then he can do so by typing the some name e.g.,“aish” in the Friend Name text box at the top-right corner as shown below



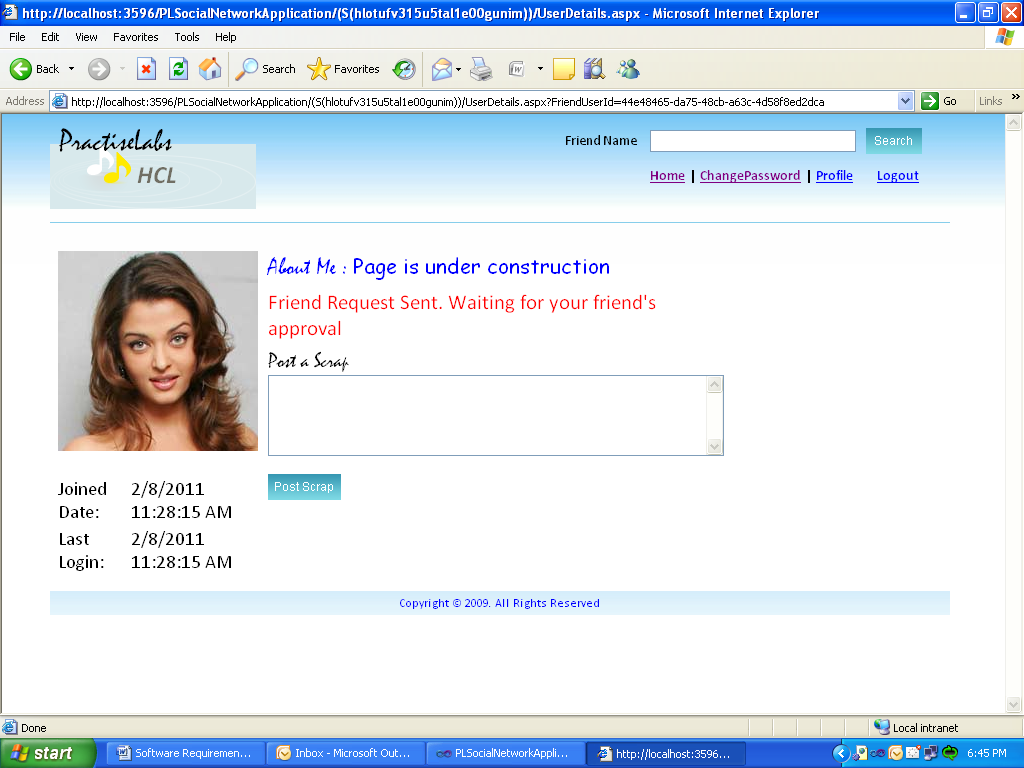
Page-6: After clicking on search button the SearchResult.aspx page is displayed with the user’s image as shown below:



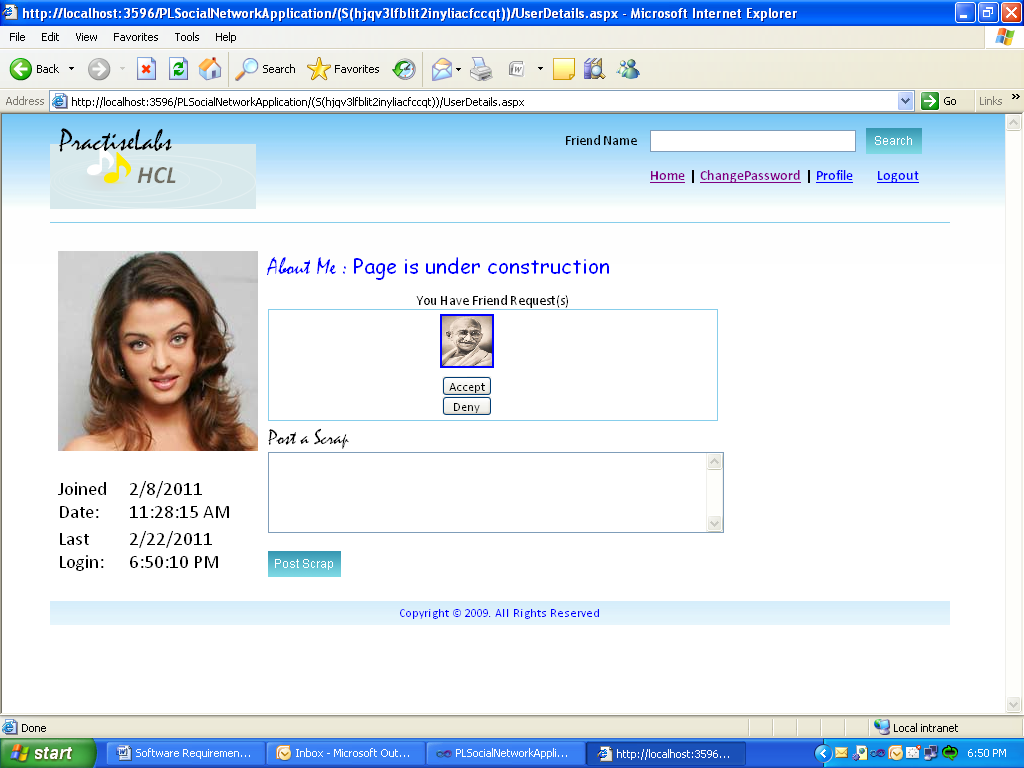
Page-7: By clicking on the users’ image the following screen is shown. If you wish to send a friend request for that user then click on “Add Me” button shown in the below screen shot



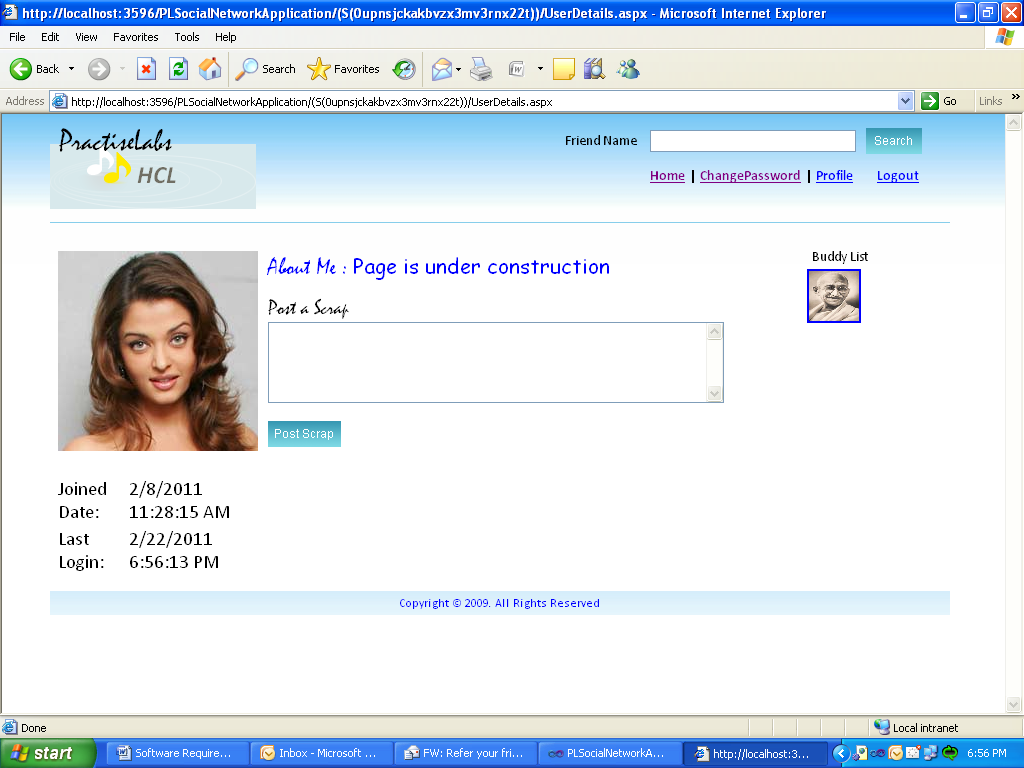
Page-8: After clicking on “Add Me” the following page is shown displaying a message “Friend Request Sent. Waiting for friend’s approval”



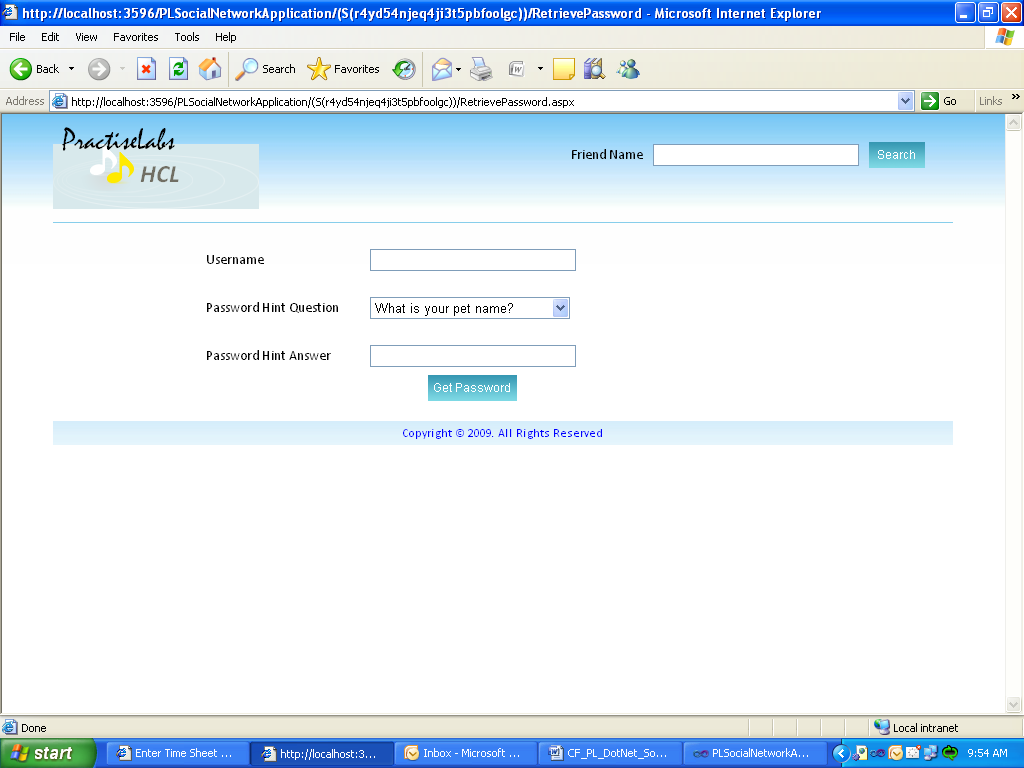
Page-9: The friend request sent is visible to the user once after she logs-in. It is as shown below:



Page-10: If the user clicks on “Accept” button the user gets added to the friends list of the current user. If denied, it is just rejected and friend will not be added.

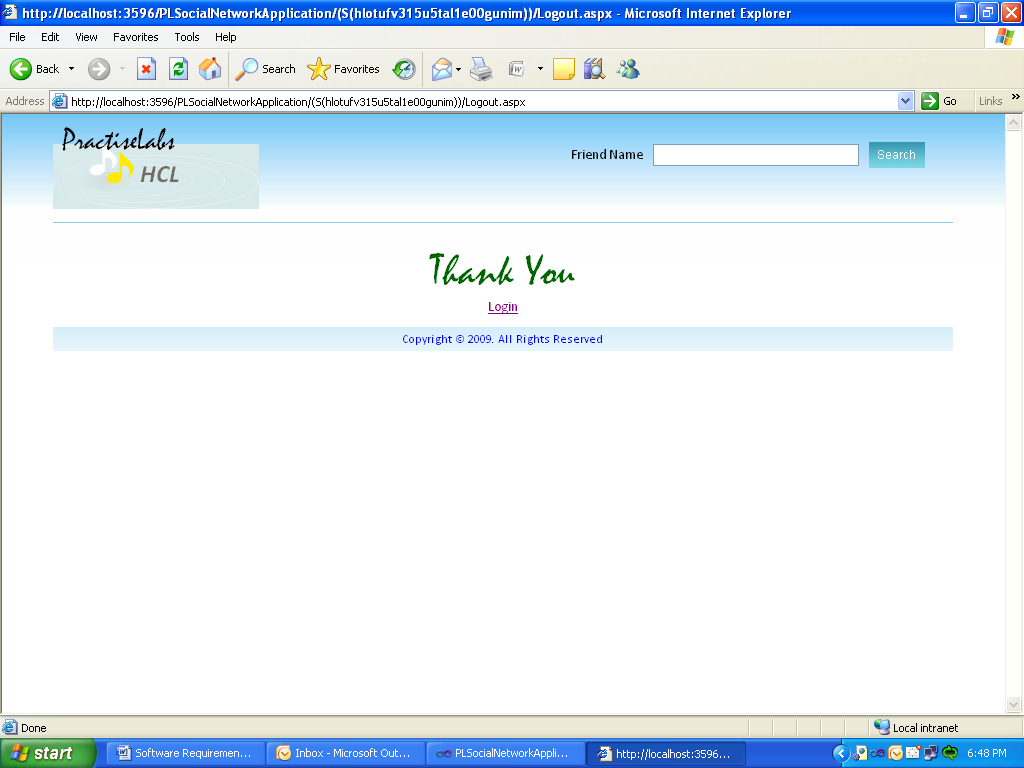


Page-11: If in case you forgot your password, you can get it back by clicking on “forgot password” on the home page which will display the below screen.



Page-12: After entering the valid details into above screen, the screen looks like below:

After clicking on logout link, then logout page is shown as below



## Functional Requirements / System Features

## Use-Case for Log-in

**Description**

This use-case describes the login page functionality.

**REQ-1.1**: The Login screen will have two fields, one is username and the other one is password

**REQ-1.2**: The User name field will be marked mandatory and the field will accept a maximum of 10 characters including characters and digits

**REQ-1.3**: The Password field will be marked mandatory and the password should be at-least 6 characters long

**REQ-1.4**: Once the registered user enters the valid username and password and clicks on the Log on button, the user will be logged in to his/her profile

**REQ-1.5**: This screen is also provided with an additional functionality of remembering the password of the user for the next time on-wards. This can be activated by the user by checking the checkbox shown besides the “Remember me next time” label

**REQ-1.6**: For a new user, the login screen also provides the option of registering an account with the web-site through the link “Not Registered Yet?” provided at the bottom of the log-in screen

**REQ-1.7**: This page also allows an anonymous user to search for his/her friends by providing the name of his/her friend in the search box. If friends with such name exist, it displays the images of those friends. Else it displays an error message for the user indicating that “no users with such name are found”

### Business Flow & User Actions

User enters the “username” and “password”

Logs-in to the respective user profile

Clicks on “Log on” Button

If validated

#### Business Rules & Validations

The business rules to be considered while entering this screen are that the user is present in the Practice Labs Login page.

The business rules to be considered while moving to the next page is that the username should be entered and is valid, the password should be entered and is valid. If they are valid and only after clicking the log on button the user will be navigated to the next page.

## 4.5.2 Use-Case for Registration

**Description**

This use-case describes the registration page functionality.

**REQ-2.1**: The Registration screen has 7fields of which 4(four) are optional and 3(three) are mandatory and are to be surely filled in by the user.

**REQ-2.2**: The Email field will be marked mandatory and the field will accept an email id in a particular format pre-defined by the owner of the site. For example, surya@gmail.com

**REQ-2.3**: The Password field will be marked mandatory and the password should be at-least 6 characters long

**REQ-2.4**: The Retype Password field will be marked mandatory and the retype password should match with the password typed earlier in REQ-003 and this retyped password should also be at-least 6 characters long

**REQ-2.5**: The Name field will be marked mandatory and the field will accept only letters, excluding digits and any other special characters. This field also accepts a maximum of 30characters only

**REQ-2.6**: The Country field is not a mandatory field and the field will accept only letters, excluding digits and any other special characters. This field also accepts a maximum of 20characters only

**REQ-2.7**: The About Yourself field is not a mandatory field and the field will accept any kind of characters. There is no limitation on the number of characters accepted by a field.

**REQ-2.8**: The Photo URL field is not a mandatory field and the field will accept the URL indication the location of the image from where you are accessing that image. This field is provided with a “Browse” option from where you can directly browse for the location where the image is located.

**REQ-2.9**: Once the new user enters the valid Email, Password, Retype Password, Name, Country (optional), About Yourself (optional), and Photo URL (optional) and as and when the “Register” button is clicked the user gets registered in the portal

### Business Flow & User Actions

User enters all the details required in the registration page

New User is registered and added to the database

Clicks on “Register” Button

Checks for validations

#### Business Rules & Validations

The business rules to be considered while entering this screen are that the user is present in the Practice Labs Registration page.

The business rules to be considered while moving to the next page is that all the required or mandatory field values are filled in by the user. If they are filled and only after the user clicks on the “Register” button user will be navigated to the next page.

## Use-Case for User Details

**Description**

This use-case describes the User Details page functionality.

**REQ-3.1**: Once after the user is logged-in to the portal, he/she can view his/her respective profile page.

**REQ-3.2**: This page also allows the logged-in user to search for his/her friends by providing the name of his/her friend in the search box. If friends with such name exist, it displays the images of those friends. Else it displays an error message for the user indicating that “no users with such name are found”

**REQ-3.3**: The Joined Date label displays the date on which the logged-in user was actually registered with the portal.

**REQ-3.4**: The Last Logged-in Date Label displays the most recent date on which the current logged-in user has accessed his/her account.

**REQ-3.5**: The Total Reviews Label displays the total number of visits to the currently logged-in user profile made by the users in the portal.

**REQ-3.6**: User Details page also allows the user to visualize the list of friend requests received present under the “Friend Requests” label.

**REQ-3.7**: User Details Page itself provides the user the lists of friend requests received and are awaiting for his/her response. He/she can either accept or deny the requests received.

**REQ-3.8**: User Details page also allows the user to view the scraps received from his friends and also allows him/her to send a reply to his/her friends scraps by clicking on “Post Scrap” button.

### Business Flow & User Actions

User enters all the details required in the registration page

New User is registered and added to the database

Clicks on “Register” Button

Checks for validations

#### Business Rules & Validations

The business rules to be considered while entering this screen are that the user should be logged-in first by giving valid credentials. Only then he/she will be allowed to navigate to next page.

The business rules to be considered while moving to the next page is that the User-Details page should contain all the required data in an accurate way and is according to the page layout

## Use-Case for Search Result

**Description**

This use-case describes the Friends search result page functionality.

**REQ-4.1**: Once the user is logged-in he/she can search for his/her friends by their name. If any user with similar sort of name exists in the database, then it gives the list of users images with those names.

**REQ-4.2**: On selecting the one whom the user wants as a friend and clicking on the images of that friend, it allows the user to go through his/her friend’s profile. If the user wants him/her as a friend, he/she can do so by sending a friend request.

**REQ-4.3**: The anonymous user can also search for friends from the home page itself without logging in. But the anonymous user is permitted only to search for friends; he/she cannot go through the friend’s profile until he/she is a registered and logged-in user like in REQ-**4.2** above.

### Business Flow & User Actions

User gets logged-in

Clicks on “Search” button

Enters the name of his/her friend in search box

User Image is displayed if the user exists

#### Business Rules & Validations

The business rules to be considered while entering this screen are that the user should be logged-in first. Only then he/she can search for and view the profile of his/her friends. Note that even the anonymous user can also search for friends but he/she will not be allowed to access the profile of his/her friends.

The business rules to be considered while moving to the next page are that the name that the user entered should already exist in the database. Only then the user can view his/her friend’s image. Clicking on this image allows the user to further navigate through his/her friend’s profile.

## Use-Case for Send Friend Requests

**Description**

By clicking on user image the page will be redirected to his/her user profile. You can send friend request to that user.

**REQ-5.1**: By clicking on the link Send Friend Request, Add Request should be sent to the user of that profile.

**REQ-5.2**: Status of the request whether success or failure should be displayed under the link.

### Business Flow & User Actions

Status is displayed

User Clicks on a profile picture

Friend Request is sent

Clicks Add Friend Button

#### Business Rules & Validations

The business rules to be considered while entering this screen are that the user should be logged-in first. Only then he/she can search for and view the profile of his/her friends. Note that even the anonymous user can also search for friends but he/she will not be allowed to access the profile of his/her friends.

The business rules to be considered while moving to the next page are that the name that the user entered should already exist in the database. Only then the user can view his/her friend’s image. Clicking on this image allows the user to further navigate through his/her friend’s profile.

## Use-Case for Accept OR Deny Friend Requests

**Description**

From here user can accept friend request and deny friend request. If user accepts request then this user will show in right side in friend section.

**REQ-6.1**: Request should be displayed with the profile picture

**REQ-6.2**: Underneath the Profile Picture there should be two buttons with names Accept and Deny respectively

**REQ-6.3**: When user Hits Accept button, Requested user is added to the friends list of current user.

**REQ-6.4**: Friend Request accepted message to be sent to the user who sent the request.

**REQ-6.5**: When User Hits Reject button, the request is rejected and the same is conveyed to requested user.

**REQ-6.6**: After Hitting Accept/Deny button user is redirected to User Details Page if no more requests are pending, otherwise control should stay with the same page and allow user to accept/deny other requests.

### Business Flow & User Actions

User enters the “username” and “password”

Logs-in to the respective user profile and can either accept/deny the friend request.

Clicks on “Log on” Button

If validated

#### Business Rules & Validations

The business rules to be considered while entering this screen are that the user should be logged-in first. Only then he/she can search for and view the profile of his/her friends. Note that even the anonymous user can also search for friends but he/she will not be allowed to access the profile of his/her friends.

The business rules to be considered while moving to the next page are that the name that the user entered should already exist in the database. Only then the user can view his/her friend’s image. Clicking on this image allows the user to further navigate through his/her friend’s profile.

## Use-Case for Verify Friends Added

**Description**

Here, user can see friends he added after approving the request. Friends list is shown at the left side of the screen.

**REQ-7.1**: Friends added should appear in the left panel of the user profile as Icons.

**REQ-7.2**: Each Icon should have its respective Profile Image.

### Business Flow & User Actions

Clicks Add or Deny Button

Logs-in to the respective user profile and can either accept/deny the friend request.

User’s profile picture is shown if his/her request Is accepted.

If validated

#### Business Rules & Validations

The business rules to be considered while entering this screen are that the user should be logged-in first. Only then he/she can search for and view the profile of his/her friends. Note that even the anonymous user can also search for friends but he/she will not be allowed to access the profile of his/her friends.

The business rules to be considered while moving to the next page are that the user has been added to your friend list.

## Use-Case for Admin Login Windows Form

**Description**

Administrator has separate windows forms functionality where the admin can login to view the list of users available in the practice-labs portal. Administrator is the only person who has rights to delete any user from the portal.

**REQ-8.1**: The Admin Login windows form will have two fields, one is username and the other one is password

**REQ-8.2**: The User name field will be marked mandatory and the field will accept a maximum of 10 characters including characters and digits

**REQ-8.3**: The Password field will be marked mandatory and the password should be at-least 6 characters long

**REQ-8.4**: Once administrator enters the valid username and password and clicks on the Log on button, the admin will be logged in to his/her profile

**Business Flow & User Actions**

Admin enters the “username” and “password”

Logs-in to the Admin profile

Clicks on “Log on” Button

If validated

### Business Rules & Validations

The business rules to be considered while entering this screen are that the user is present in the Practice Labs Admin Login windows form.

The business rules to be considered while moving to the next page is that the username should be entered and is valid, the password should be entered and is valid. If they are valid and only after clicking the log on button the administrator will be navigated to the next page.

## Use-Case for Admin Profile Windows Form

**Description**

Administrator has profile windows form where he/she can view the list of users available in the practice-labs portal. Administrator is the only person who has rights to delete any user from the portal.

**REQ-9.1**: The Admin Profile windows form should have a data-grid showing the list of users available on the portal

**REQ-9.2**: The Admin Profile windows form should also have a “Delete” and “Update” button being shown at the bottom of the windows form

**Business Flow & User Actions**

Admin Logs on

Admin Profile is shown with list of all users and also has “Delete” and “Update” button at the bottom

Views his/her profile

### Business Rules & Validations

The business rules to be considered while entering this screen are that the user is present in the Practice Labs Admin Login windows form.

The business rules to be considered while moving to the next page is that the username should be entered and is valid, the password should be entered and is valid. If they are valid and only after clicking the log on button the administrator will be navigated to the next page.

## Use-Case for deleting a user from data grid

**Description**

Administrator has profile windows form where he/she can view the list of users available in the practice-labs portal. Admin has the rights to delete any user from the database. If the admin selects a particular user from the grid and hits on “Delete” button then that particular user should be deleted from the grid as well as database

**REQ-10.1**: Admin selects a particular user by clicking on an associated row of that user.

**REQ-10.2**: If the admin hits on “delete” button then that selected user row should be deleted from the grid as well as database

**Business Flow & User Actions**

Admin is in profile page

Clicks on “Delete” button on the form

Admin selects any particular user

Admin can see the user deleted from the grid

#### Business Rules & Validations

The business rules to be considered while entering this screen are that the user is present in the Practice Labs Admin Login windows form.

The business rules to be considered while moving to the next page is that the username should be entered and is valid, the password should be entered and is valid. If they are valid and only after clicking the log on button the administrator will be navigated to the next page.

## Use-Case for viewing current list of users in database

**Description**

Admin has the rights to delete any user from the database. If the admin selects a particular user from the grid and hits on “Delete” button then that particular user should be deleted from the grid as well as database. This “Update” button helps the admin to retrieve the current list of users from the database so that he/she can check whether the users deleted from grid were also deleted from database or not

**REQ-11.1**: Admin clicks on “Update” button which retrieves list of users from database

**REQ-10.1**: Now the admin selects a particular user (for example “Rahul”) by clicking on an associated row of that user.

**REQ-10.2**: Then the admin hits on “Delete” button then that selected user row (Rahul) should be deleted from the grid as well as database

**REQ-11.1:** Now theadmin again clicks on “Update” button which retrieves current and updated list of users from database (it won’t have the user “Rahul” in the retrieved list)

**Business Flow & User Actions**

Admin is in profile page

Admin selects any particular user

Clicks on “Delete” button on the form

Admin can see the user deleted from the grid

Clicks on “Update” button on the form

Admin can see the user deleted from DB and the updated list is retrieved

#### Business Rules & Validations

The business rules to be considered while entering this screen are that the user is present in the Practice Labs Admin Login windows form.

The business rules to be considered while moving to the next page is that the username should be entered and is valid, the password should be entered and is valid. If they are valid and only after clicking the log on button the administrator will be navigated to the next page.

## Performance Requirements

The performance requirements are listed below:

* The “Response time” of the web-site depends on number of users accessing the web-site at a single point of time.
* The system “priorities” varies depending upon the type of user accessing the application like registered user, un-registered user or administrator
* Fault Recovery is not considered here as no such issues were found.
* The system is available for 24X7 hours and can be accessed at anytime if the user has internet connection available.

## 4.7 Acceptance Criteria

The application developed will be well accepted only if:

* It meets all the needs of the customers as specified in the requirements document.
* All the functionalities should be met and they should be verifiable

## Logical Database Requirements

**Table Name: User**

Primary Key: InfoId

Description: It contains details of the entire user’s who got registered with the portal and who were the members of the portal. As and when a new user gets registered with this web-site, the user will be added to this table.

User here always refers to a user who is getting registered now. Because the details that get stored into this table are the details which are provided by the user during registering with the portal.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Description | Key Type |
| InfoId | biginteger | UserInfo Identifier for each of the registered user | Primary key |
| UserId | Unique identifier | Unique User Id of the registered user from the asp.net membership | - |
| RegisteredDate | datetime | Date on which the user was registered | - |
| LastLogin | datetime | Date on which the user was last logged-in | - |
| Description | nvarchar(50) | Text describing the user | - |
| ProfileImage | nvarchar(50) | The user’s profile image url is stored here | - |
| HereFor | nvarchar(50) | Text describing the intention of the user to join in the portal | - |
| Name | nvarchar(50) | User’s Name | - |

**Table Name: UserOtherDetails**

Primary Key: No

Foreign Key: InfoId, Occupation Id

Description: It contains profile information of the user. That is, all the other personal details of the users who get registered with the portal and who were the members of the portal. As and when a new user gets registered with this web-site, the user will be added to this table.

User here always refers to a user who is getting registered now. Because the details that get stored into this table are the details which are provided by the user during registering with the portal.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Description | Key Type |
| InfoId | biginteger | Identifier that refers to InfoId of UserInfo table | Foreign key |
| OccupationId | integer | Refers to OccupationId field of Users table | Foreign key |
| Name | nvarchar(50) | User’s Name | - |
| DateOfBirth | date | User’s Date of Birth | - |
| Address | nvarchar(50) | Address of the user | - |
| City | nvarchar(50) | City of the user | - |
| State | nvarchar(50) | User’s State | - |
| Country | nvarchar(50) | Country of the user |  |
| Sex | nvarchar(8) | User’s Gender |  |

**Table Name: FriendRequests**

Primary Key: No

Foreign Key: FromUserId, ToUserId

Description: It contains details of users who has sent the friend requests to other users on the portal and stores both the userid of the user who has sent the request and the userid of the user to whom the request was sent.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Description | Key Type |
| FromUserId | uniqueidentifier | Refers to the user ID of aspnet\_users table | Foreign key |
| ToUserId | uniqueidentifier | Refers to the user Id of aspnet\_users table | Foreign key |

**Table Name: RelationShip**

Primary Key: RelationShipId

Foreign Key: No

Description: It contains the different types of relationships that the user may have with the other users on the portal.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Description | Key Type |
| RelationShipId | integer | The primary key that increments automatically | Primary key |
| RelationShip | nvarchar(50) | Refers to the name of the relationship | - |

**Table Name: Occupation**

Primary Key: OccupationId

Foreign Key: No

Description: It contains the different types of occupations that the users on the portal may have in their profile information

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Description | Key Type |
| OccupationId | integer | The key that increments automatically | Primary key |
| Occupation | nvarchar(50) | Refers to the name of the occupation | - |

**Table Name: ScrapsTable**

Primary Key: No

Foreign Key: FromUserId, ToUserId

Description: It contains details of the scraps that the users on the portal send as a reply to their friend’s message. As and when any user posts/sends a scrap to any other user on this portal, it gets recorded into this Scraps table.

User here always refers to a registered user. Because only a registered user can send/post a scrap as a reply to the other user’s on the portal. An anonymous user cannot

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Description | Key Type |
| FromUserId | uniqueidentifier | Refers to the user ID of aspnet\_users table | - |
| ToUserId | uniqueidentifier | Refers to the user ID of aspnet\_users table | - |
| Message | nvarchar(50) | Scrap/Message sent by user to his/her friend | - |
| PostedDate | Datetime | The date on which the scrap was posted | - |

## Design Constraints

The database is completely designed and developed to run on Microsoft SQL Server 2008. But it can be run on almost any version of SQL Server

* Naming conventions and packaging of classes should be done properly

## Testing Requirements

For this application to be tested, the system should have a browser available on it. Then the testing can be done by individually testing each module’s functionality as per the test-cases defined in the test-cases document.

## Compliance to Standards

There are no standards or regulation pertaining to this application

## Software System Attributes

**Functionality**

This application primarily allows the user to register with the portal so that his/her profile is visible to other users on the portal and they can communicate with each other as friends. The user is also allowed to search for new friends, make them his/her friends, and post scraps for them

**Suitability**

This application is more suitable for those who have their like for social networking sort of sites and whoever is interested in making more and more friends.

**Accuracy**

This application is accurate enough as it displays the correct login and profile information for every logged-in user

* Sends the friend requests exactly to your desired ones
* Routes promptly to your friend’s profile when you click on your friend’s profile image

**Interoperability**

The application is interoperable as it can be run from any remote system once it is hosted on to a centralized web-server. Or else the only thing it requires was a visual studio and SQL server database to be installed on to the other system, so that the application could be run on any new system.

**Security**

The application provides the user with a high-level of security. The user profile details will be shared only with the other users registered with the portal. No anonymous user can have access to the registered users profile information. Secure information like users credentials (username and password) are not shared with anyone other than administrator him-self.

Input data validation

The data that the user enters to log-in to the portal needs to be validated against the registered users’ data. Only then the user should be allowed to log-on to the site.

**Functionality compliance**

The functionality of the application complies with the standards defined which can be examined from the below conditions.

* Once after logging in, the user cannot navigate back to login page
* After logging out, the user cannot come back to his/her profile page

Thus the web-site’s functionality complies with the basic standards defined.

**Reliability**

Not Applicable

**Maturity**

Not Applicable

**Fault tolerance**

Not Applicable

**Recoverability**

If the system or browser gets abruptly closed while the user was logged-in, then the user needs to log-in again which shows security. Hence recoverability was not considered here as we need to take care of security.

**Usability**

The web-site can be used once if the user is registered with the portal. The user is allowed to create a profile of his own, find friends, send friend requests, view friend’s profile, post scraps and he can even delete an existing friend if he no more wants him as a friend

**Understandability**

The user can understand what the site is about from the tag on the home page “connect, collaborate, learn and grow” which states that the user can find new friends of different occupations, can start discussions using scraps, learn from the discussions and can grow with what you have learnt

**Learn ability**

Users with different types of occupations are available on the portal. So communicating with those users who were experts in some domain helps the other users in learning. So even social networking sites make the users learn something from other users if used in a proper way

**Operability**

Any user who has the basic knowledge of using internet can join this portal and can start operating his profile.

**Attractiveness**

The application is very attractive in its user interface and also grabs the attention of the users as it is more on making friends on-line.

**Efficiency**

As the web-site works implementing all the functionalities as expected it is said to be efficient enough. It works for 24X7 hours.

**Time Behavior**

Not Applicable

**Resource Utilization**

The application requires the following resources to develop and design it:

* Microsoft Visual Studio
* Microsoft SQL server 2008 (or other versions)
* Microsoft IIS server
* Visual studio test tool for testing

**Maintainability**

Maintaining such an application is pretty easy as the database it-self handles all of the issues for most of the time. But depending upon the number of users accessing the application the maintenance may be slow sometimes because of heavy load on server.

**Analyzability**

The application is very easy to analyze as the application is split in to three tired architecture and only web-tier is what that is visible to the user. Once the user gets registered with the portal and test the functionality the user can analyze the complete application in a single step by concentrating more on how it works.

**Changeability**

If the application requires any changes with the database configuration or etc., it is very easy to do so as the only changes we have to do are mostly in the configuration file alone.

**Stability**

Not Applicable

**Testability**

The testing is done based on the functionality of the application and the testing is done almost on every field entered by the user which is then validated to provide security and not allowing the un-authenticated users

**Maintainability Compliance**

Not Applicable

**Portability**

The application is easily portable on to any platform

**Adaptability**

Not Applicable

**Replaceability**

Not Applicable

**Availability**

Once hosted onto the World Wide Web the application will be available for 24X7 hours or may change depending upon the vendor’s requirement

**Safety**

The application is safe enough as it doesn’t expose the user details to anonymous users

**Environmental**

The application is configured to run either on Microsoft SQL server 2008 environment. However changing the application to run on any-other version of SQL server doesn’t require lot many changes.

**Other Requirements**

For further extending, we can modify the application so that a new user registration is able to send notification by SMS.

## Statutory and Regulatory Requirements

The application does not have any statutory and regulatory requirements because this doesn’t concern with preparing such financial statements and all

## Organizing Specific Requirements

The organizations of requirements are listed as the following sub-classes and are described as shown below:

### System Mode

The system can either be operated by a user or an administrator. The system will be performing better when the mode of operation is under admin as the application is designed specially in a windows form application and there are no simultaneous users who can use that application. Hence the performance of system depends on its mode

### User Class

Users are categorized into two classes which can be either a registered user or an un-registered user.

* Registered users can search for friends, view the profile of searched users, if interested can send a friend request, post scraps or delete the friend already there in his profile
* Un-registered users can only search for friends, but cannot view the profile of searched users. For all other actions they need to log-in first

### Objects

Sending Friend Request, Adding a friend, Deleting an existing friend, Searching for a friend, viewing the profile of a friend, posting scraps are the different objects of this system/application.

### Features

The primary thing that is required to start this application is the valid username and password to log-on to the portal.

# Derived Requirements

The lists of requirements that are derived as a matter of security are:

* Configuring the application to use aspnet\_membership so that every user can have a unique-identifier which differentiates one user from the other users on the portal and which will be more secure.
* Configuring the application to have a session time-out for the user, if the page was idle for 5minutes or more.
* Restricting the backward navigation on the page appearing after immediate log-in of a user
* Restricting the backward navigation on the page appearing after immediate log-out of a user
* Allowing the user to navigate between the pages after he logs-on to until he logs-out

# Validation

## Validation strategy

Validation strategy here is a two-phase task. In the first phase, the application is tested to ensure that code is developed properly and performs reliably, efficiently, and concisely. It is usually done using “Visual Studio Test Document” it-self.

The second phase involves operational qualification that is “GUI Testing” which was performed at the user’s site. For operational qualification, the application is tested to know whether the specific requirements for which it is designed were met or not. This is done by testing the whole functionalities of the web-site from user interface

Thus Visual Studio testing and GUI testing are done as validation strategies.

## Validation criteria

* *Accuracy* of the desired application
* Validation Strategy chosen should be *robust*.
* *Awareness –* which requiresacquaintance with general concepts, topics, rules, methods or procedures, without necessarily being able to paraphrase or summarize information and also referring to source material for more in-depth knowledge.
* *Knowledge -* familiarity with specific information, including facts, definitions, rules, methods, process and functionalities without necessarily being able to see its fullest application
* *Understanding -* identification, assimilation and comprehension of information and relating it to its practical application.
* *Ability -* skill in relating specific information to the accomplishment of tasks i.e., correctly selecting the information that is appropriate to a situation and applying it to the solution of specific problems.

## Validation Constraints

There are several validation constraints that are supposed to be checked depending on the type of field that is to be validated:

* The Email field specifies a constraint in such a way that the desired e-mail id of the user should be in a specific format only (ex:hcl-users@hcl.com)
* The password field should have more than 6characters with at-least one alpha-numeric character and special character
* The password field should be of type “password” not of “text”

# References

1. CF\_PL\_DotNet\_SocialNetworking\_TestCases.xls
2. Referring real-time social network sites like orkut, face-book and etc., for a better understanding the application.