Test Plan  
CSCI 4830  
  
6 Degrees

Team 3  
Josua Wiley  
Anna Buhman  
Douglas Martin  
Justin Hamilton

Table of Contents

[Revision History 3](#_Toc276919871)

[1 Requirements Traceability 4](#_Toc276919872)

[2 Tested Items 6](#_Toc276919873)

[3 Testing Schedule 6](#_Toc276919874)

[3.1 Test Strategy 6](#_Toc276919875)

[3.2 Test Priorities 6](#_Toc276919876)

[3.3 Integration Plan 6](#_Toc276919877)

[4 Test Recording Procedures 6](#_Toc276919878)

[5 Constraints 6](#_Toc276919879)

[6 Test Cases 7](#_Toc276919880)

[6.1 System Access 7](#_Toc276919881)

[6.2 User Management 10](#_Toc276919882)

[6.3 Friend Management 10](#_Toc276919883)

[6.4 Connection Path 11](#_Toc276919884)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| JW | 11/08/2010 | * Changed Input & Output * Changed Test Strategy * Changed Test Recording Procedures | 1.4 |
| JH | 11/07/2010 | * Reimplimented Constraints. * Added Implementation Traceability | 1.3 |
| JW | 11/07/2010 | * Reimplimented Test Cases * Added Table of Content * Formatted Document * Reimplimented Tested Cases * Reimplimented Requirement Tracebility | 1.2 |
| AB | 11/05/2010 | * Added initial Requirements Traceability * Added Tested Items * Added Testing Schedule * Added Test Recording Procedure * Added Test Cases | 1.1 |
| AB | 11/05/2010 | Initial Draft | 1.0 |

# 1 Requirements Traceability

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tester** | **User Requirement** | **System Requirement** | **Implementation** | **Test Case** |
| System Access | 3.1 | 6.1 |  | 6.1 |
| Access of Internal Page | 3.1 | 6.1.2-6.1.5 | index.html logon.html | 6.1.1 |
| Login | 3.1.6 | 6.1.1.2 | index.html login.html 6Degrees.js 6Degrees.css Logon.js Service.js layoutContainer.js user.js | 6.2.2 |
| Create Account | 3.1.1-3.1.3 | 6.1.1.1, 6.2.1 | index.html login.html 6Degrees.js 6Degrees.css Logon.js Service.js layoutContainer.js index.js auth.js user.js | 6.3.3 |
| User Management | 3.1.2 | 6.1.2 |  | 6.2 |
| View User Information | 3.1.2 | 6.2.1 | index.html login.html 6Degrees.js 6Degrees.css Logon.js Service.js layoutContainer.js index.js auth.js user.js | 6.2.1 |
| Friend Management | 3.1.4,3.1.6 | 6.1.4-6.1.5, 6.2.2-6.2.4 |  | 6.3 |
| View Friends | 3.1.4 | 6.1.4, 6.2.2 | index.html login.html 6Degrees.js 6Degrees.css Logon.js Service.js layoutContainer.js index.js auth.js user.js | 6.3.1 |
| User Search | 3.1.4 | 6.2.3 | index.html login.html 6Degrees.js 6Degrees.css Logon.js Service.js layoutContainer.js index.js auth.js user.js | 6.3.2 |
| Connection Path | 3.1.8-3.1.9 | 6.1.4-6.1.5 |  | 6.4 |
| Create Connection Path | 3.1.8 | 6.1.4-6.1.5 | graph.js index.html login.html 6Degrees.js 6Degrees.css Logon.js Service.js layoutContainer.js index.js user.js | 6.4.1 |
| Modify Connection Path | 3.1.9 | 6.1.4-6.1.5 | graph.js index.html login.html 6Degrees.js 6Degrees.css Logon.js Service.js layoutContainer.js index.js user.js | 6.4.2 |

# 2 Tested Items

These are the different cases that will be tested in 6 Degrees project:

* Access of Internal Page
* Login
* Create Account
* View User Information
* View Friends
* User Search
* Create Connection Path
* Modify Connection Path

# 3 Testing Schedule

## 3.1 Test Strategy

The tests will be performed with unit testing. To help automate the process, Selenium and Common.js will be implemented. These will allow, once all the files and situations are set up, to have all the cases automated.

## 3.2 Test Priorities

The first priority of this test will be to test that all of the functional requirements are met.  After that, the test is to make sure that the system is able to take multiple users and that it will validate the user information.

## 3.3 Integration Plan

Each part of the application will be tested as it is completed, and if it is shown to not meet the requirements, it will be fixed before the next part is made.  When the entire application is completed, it shall be tested again as a whole and with multiple users.

# 4 Test Recording Procedures

The main bold titles represent tests to be verified, with subtests within each test to confirm different successful cases that accomplish the objective and successful cases that need modifications to perform the objective.

# 5 Constraints

The application can only be tested with a moderate amount of users mostly through word-of-mouth or within the Software Engineering class itself. The application is currently only implementable using Facebook’s API, and although other social networks would like to be added in as well, Facebook is the base of 6 Degree’s test network. The obvious constraint (although also a feature) includes the fact that any network of friends will only be search within six degrees, meaning that there is a limit to the application, but this prevents search cases that are outside of reasonable time boundaries.

# 6 Test Cases

## 6.1 System Access

**6.1.1 Access of Internal Page**

6.1.1.1 Objective:  Verify the user can access login page.

Inputs:  None

Expected output:  Success without errors

Requirement being tested: **REQ 6.1.1.2**

6.1.1.2 Objective: Verify user cannot access internal page without correct session ID.

Inputs:  None

Expected output:  Success without errors

Requirement being tested: **REQ 6.1**

6.1.3.1 Objective: Verify user can view the create new account page.

Inputs: None

Expected Output: Success without errors

Requirement being tested: **REQ 6.1.1.1**

**6.1.2 Login**

6.1.2.1 Objective: Verify user cannot gain access with incorrect password but correct email.

Inputs: A valid 6 Degree account email and a password not associated with that account

Expected Output: Home page with error message to user

Requirement being tested: **REQ 6.1.1.2**

6.1.2.2 Objective: Verify user cannot gain access to system with blank email.

Inputs: A password not associated with that account

Expected Output: Home page with error message to user

Requirement being tested: **REQ 6.1.1.2**

6.1.2.3 Objective: Verify user cannot gain access to system with blank password and correct 6 Degrees account email

Inputs: A valid 6 Degrees account email

Expected Output: Success without errors

Requirement being tested: **REQ 6.1.1.2**

6.1.2.4 Objective: Verify user cannot gain access with blank password and blank 6 Degrees account email

Inputs: none

Expected Output: Success without errors

Requirement being tested: **REQ 6.1.1.2**

6.1.2.5 Objective: Verify user cannot gain access to system with blank 6 Degrees account email and a password

Inputs: a password

Expected Output: Success without errors

Requirement being tested: **REQ 6.1.1.2**

6.1.2.6 Objective: Verify user can have successful authentication

Inputs: A valid 6 Degrees account email and password

Expected Output: Success without errors

Requirement being tested: **REQ 6.1.1.2**

**6.1.3 Create Account**

6.1.3.2 Objective: Verify user cannot create account with invalid email

Inputs: an invalid email

Expected Output: create account page with error message

Requirement being tested: **REQ 6.1.1.1.1 & REQ 6.1.1.1.3**

6.1.3.3 Objective: Verify user cannot create account with valid email and blank password

Inputs: a valid email

Expected Output: create account page with error message

Requirement being tested: **REQ 6.1.1.1.1 & REQ 6.1.1.1.3**

6.1.3.4 Objective: Verify user cannot create account if email and verify email field are different.

Inputs: a valid email in email field, a valid email in verify email field that is not the same as email field

Expected Output: create account page with error message

Requirement being tested: **REQ 6.1.1.1.1 & REQ 6.1.1.1.3**

6.1.3.5 Objective: Verify user cannot create account if password and verify password field are different.

Inputs: a string in the password field, a different string in the verify password field

Expected Output: create account page with error message

Requirement being tested: **REQ 6.1.1.1.1 & REQ 6.1.1.1.3**

6.1.3.6 Objective: Verify user cannot create account if first name is blank.

Inputs: none

Expected Output: create account page with error message

Requirement being tested: **REQ 6.1.1.1.1 & REQ 6.1.1.1.3**

6.1.3.7 Objective: Verify user cannot create account if last name is blank.

Inputs: none

Expected Output: create account page with error message

Requirement being tested: **REQ 6.1.1.1.1 & REQ 6.1.1.1.3**

6.1.3.8 Objective: Verify user cannot create account if date of birth is blank.

Inputs: none

Expected Output: create account page with error message

Requirement being tested: **REQ 6.1.1.1.1 & REQ 6.1.1.1.3**

6.1.3.9 Objective: Verify user cannot create account if date of birth is incorrect date.

Inputs: A string in the date of birth that is not a possible date of birth

Expected Output: Create account page with error message

Requirement being tested: **REQ 6.1.1.1.1 & REQ 6.1.1.1.3**

6.3.2.10 Objective: Verify adds friend from Facebook if link Facebook account with 6 Degree account.

Inputs: Facebook Cookie

Expected Output: User Home Screen with welcome message

Requirement being tested: **REQ 6.1.1.1.2 & 6.2.1**

6.3.2.12 Objective: Verify adds friend from Facebook if link Facebook account with 6 Degree account.

Inputs: Facebook cookie and user account

Expected Output: Success without errors

Requirement being tested: **REQ 6.1.1.1.2 & 6.2.1**

## 6.2 User Management

**6.2.1 View User Information**

6.3.1.1 Objective: Verify the user can view his/her account information.

Inputs: Successful authentication

Expected Output: Display user metadata

Requirement being tested: **REQ 6.1.2**

## 6.3 Friend Management

**6.3.1 View Friends**

6.3.1.1 Objective: Verify display no friends if no friends in 6 Degree database.

Inputs: None

Expected Output: Display view friend page with "No current 6 Degree friends" message

Requirement being tested: **REQ 6.1.2 & 6.1.4 & 6.2.3**

6.3.1.2 Objective: Verify displays all friends on 6 Degree database.

Inputs: None

Expected Output: Display view friend page with each 6 Degree friend of that person listed

Requirement being tested: **REQ 6.1.2 & 6.1.4 & 6.2.3**

**6.3.2 User Search**

6.3.2.1 Objective: Verify cannot search with blank string for a query.

Inputs: the user has selected to search for the query which is blank

Expected Output: search form with error message

Requirement being tested: **REQ 6.1.4 & 6.2.3**

6.3.2.2 Objective: Verify search displays result for one word.

Inputs: a string with only one word

Expected Output: a collection of related user objects

Requirement being tested: **REQ 6.1.4 & 6.2.3**

6.3.2.1 Objective: Verify search displays results for multiple words.

Inputs: a string with multiple words

Expected Output: : a collection of related user objects

Requirement being tested: **REQ 6.1.4 & 6.2.3**

## 6.4 Connection Path

**6.4.1 create connection path**

6.4.1.1 Objective: Verify displays no connection message if no connection exists.

Inputs: user has selected to see a connection between himself/herself and another user

Expected Output: Form connection page with message stating no such connection can be made at this time

Requirement being tested: **REQ 6.1.5 & 6.2.3**

6.4.1.2 Objective: Verify displays loading message while calculating connection path.

Inputs: user has selected to see a connection between himself/herself and another user

Expected Output: Form connection page with message telling user to wait while it calculates

Requirement being tested: **REQ 6.1.5 & 6.2.3**

6.4.1.3 Objective: Verify displays connection once calculated.

Inputs: user has selected to see a connection between himself/herself and another user

Expected Output: Form connection page graphical connection path

Requirement being tested: **REQ 6.1.5 & 6.2.3**

6.4.1.4 Objective: Verify displays connection once calculated with selected 6 Degree account holders not included in the connection path.

Inputs: user has selected to see a connection between himself/herself and another user, the user has selected 6 Degree account holders to be ignored in the path

Expected Output: Form connection page graphical connection path

Requirement being tested: **REQ 6.1.5 & 6.2.3**

6.4.1.5 Objective: Verify user can select a user to be ignored in developing a connection path.

Inputs: the user selects one of the users in the connection path to be avoided in the connection path

Expected Output: the same page the user is on displays a messaging stating the selected 6 Degree account will not be implemented in the next connection path in this session

Requirement being tested: **REQ 6.1.5 & 6.2.3**