Memorial University of Newfoundland Social Network(MUNSN)

Prepared By: Group D

Tyler Vey

Saahil Budhrani

Mark Hewitt

Allan Collins

MUNSN Module Description Document

March 17, 2017

TABLE OF CONTENTS

1.0 Overview....7
2.0 Modules....8
  2.1 Module List....8
  2.2 Module Descriptions....8
    2.2.1 Timeline Module....8
      2.2.1.1 User Interface....8
        Timeline Visibility Feature....8
        Timeline Post Visibility....8
        Timeline Posting Rights....8
      2.2.1.2 Service Calls....9
    2.2.2 Resume Module....9
      2.2.2.1 User Interface....9
    2.2.3 Private Messaging Module....9
      2.2.3.1 User Interface....9
    2.2.4 Lost and Found Module....9
      2.2.4.1 User Interface....9
    2.2.5 Polls Module....10
      2.2.5.1 User Interface....10
      2.2.5.2 Creating a Poll....10
      2.2.5.3 Voting on a Poll....10
    2.2.6 Friends Module....10
      2.2.6.1 User Interface....10
    2.2.7 Study Groups Module....11
      2.2.7.1 User Interface....11
    2.2.8 Account Module....11
      2.2.8.1 User Interface....11
    2.2.9 Schedule Module....11
      2.2.9.1 User Interface....11
      2.2.9.2 Add A New Course to Schedule....12
      2.2.9.3 Remove a Course from Schedule....12
3.0 Testing....12
  3.1 Software Development LifeCycle Architecture....12
    3.1.1 Unit Testing....13
    3.1.2 Integration Testing....13
    3.1.3 Regression Testing....13
  3.2 Test Cases....13
    3.2.1 Test Cases Overview....13
    3.2.1 Timeline Module....14
      3.2.1.0 TC-TL-00 the sample one. 14
        Actors....14
        Pre-Conditions....14

Test Case Steps....14
Comments....15
Overall result....15
3.2.1.1 TC-TL-01 View a timeline....15
Actors....15
Pre-Conditions....15
Steps....15
Comments....15
Overall Result....15
3.2.1.2 TC-TL-02 Create a Post on own Timeline....15
Actors....15
Pre-Conditions....15
Steps....16
Comments....16
Overall Result....16
3.2.1.3 TC-TL-03 Create a Post on someone else’s Timeline....16
Actors....16
Pre-Conditions....16
Steps....16
Comments....17
Overall Result....17
3.2.1.4 TC-TL-04 Attach Media to A Post....17
Actors....17
Pre-Conditions....17
Steps....17
Comments....17
Overall Result....17
3.2.1.5 TC-TL-05 Attach Media to A Post - Error Warnings....18
Actors....18
Pre-Conditions....18
Steps....18
Comments....18
Overall Result....18
3.2.1.6 TC-TL-06 Comment on a Post....18
Actors....18
Pre-Conditions....19
Steps....19
Comments....19
Overall Result....19
3.2.1.7 TC-TL-07 Edit a Post Comment....19
Actors....19
Pre-Conditions....19
Steps....19
Comments....20

Overall Result......20
3.2.1.8 View Post Comment History......20
Actors......20
Pre-Conditions......20
Steps......20
Comments......20
Overall Result......20
3.2.1.9 TC-TL-09 Timeline Visibility - Guest......20
Actors......20
Pre-Conditions......21
Steps......21
Comments......21
Overall Result......21
3.2.1.10 TC-TL-10 Timeline Visibility - Friend......21
Actors......21
Pre-Conditions......22
Steps......22
Comments......22
Overall Result......22
3.2.1.11 Individual Post Visibility - Guest......22
Actors......22
Pre-Conditions......22
Steps......22
Comments......23
Overall Result......23
3.2.1.12 Individual Post Visibility - Friend......23
Actors......23
Pre-Conditions......23
Steps......23
Comments......24
Overall Result......24
3.2.1.13 Individual Post Visibility does not get Overwritten by Timeline Post Visibility. 24
Actors......24
Pre-Conditions......24
Steps......24
Comments......24
Overall Result......24
3.2.2 Resume Module......24
3.2.2.1 Uploading a Resume......24
Actors......24
Pre-Conditions......24
Steps......25
Comments......25
Overall Result......25

3.2.2.2 Deleting a Resume....25
Actors....25
Pre-Conditions....25
Steps....25
Comments....26
Overall Result....26
3.2.3 Private Messaging Module....26
3.2.3.1 TC-PM-01....26
Actors....26
Pre-Conditions....26
Steps....26
Comments....26
Overall Result....26
3.2.4 Lost and Found Module....27
3.2.4.1 Create Lost and Found Post....27
Actors....27
Pre-Conditions....27
Steps....27
Comments....27
Overall Result....27
3.2.4.2 View Lost and Found Post....27
Actors....27
Pre-Conditions....27
Steps....28
Comments....28
Overall Result....28
3.2.5 Polls Module....28
3.2.5.1 Creating a New Poll....28
Actors....28
Pre-Conditions....28
Steps....28
Comments....28
Overall Result....29
3.2.5.2 Voting on a Poll....29
Actors....29
Pre-Conditions....29
Steps....29
Comments....29
Overall Result....29
3.2.6 Friends Module....29
3.2.6.1 TC-FR-01 View Friend Recommendations....29
Actors....29
Pre-Conditions....29
Steps....29

Comments......30
Overall Result......30
3.2.6.2 TC-FR-02 Send Friend Request......30
Actors......30
Pre-Conditions......30
Steps......30
Comments......30
Overall Result......30
3.2.6.2 TC-FR-03 View Friend List......30
Actors......30
Pre-Conditions......30
Steps......30
Comments......31
Overall Result......31
3.2.7 Study Groups Module......31
3.2.7.1 Create a Study Group - Public Group......31
Actors......31
Pre-Conditions......31
Steps......31
Comments......31
Overall Result......31
3.2.7.2 Create a Study Group - Private Group......32
Actors......32
Pre-Conditions......32
Steps......32
Comments......32
Overall Result......32
3.2.7.3 Invite to Study Group......32
Actors......32
Pre-Conditions......32
Steps......32
Comments......33
Overall Result......33
3.2.8 Account Module......33
3.2.8.1 TC-AC-01 Create and Confirm Account......33
Actors......33
Pre-Conditions......33
Steps......33
Comments......33
Overall Result......33
3.2.9 Schedule Module......34
3.2.9.1 Adding a Class to a Schedule via Suggested Courses......34
Actors......34
Pre-Conditions......34

Steps....34
Comments....34
Overall Result....34
3.2.9.2 Adding a Class to a Schedule via Searched Courses....34
Actors....34
Pre-Conditions....34
Steps....34
Comments....35
Overall Result....35
3.2.9.3 Removing a Class from the Schedule....35
Actors....35
Pre-Conditions....35
Steps....35
Comments....36
Overall Result....36
3.4 Web Service Call Testing....36
3.5 Non-Functional Testing....36
4.0 Document Responsibilities....36
5.0 Coding Effort and Priority....37
6.0 Schedule....37
Week 09 Feb-27 to Mar-05....37
Week 10 Mar-06 to Mar-12....38
Week 11 Mar-13 to Mar-19 (Planned)....38
Week 12 Mar-20 to Mar-26 (Planned)....38
Week 13 Mar-27 to Apr-02 (Planned)....38
Week 14 Apr-03 to Apr-05 (Planned)....38

1.0 Overview

This document seeks to provide an actionable description of each module to be included in the Mun Social Network project. It consists primarily of two major parts, the Modules and the Testing sections.

The Modules section will, in accordance with its name, provide a description of each module. Each description will cover both the functionality and the interface of that module.

The Testing section will cover the intended testing plan for the project. It will first provide background and details for the overall testing of the project then move on to providing the planned test cases for each module and its individual parts.

2.0 Modules

This section is intended to cover each module in the project, giving information on their functionality and interface.

2.1 Module List

1. Timeline Module
2. Resume Module
3. Private Messaging
4. Lost and Found
5. Polls
6. Friends
7. Study Groups
8. Account Module
9. Schedule

2.2 Module Descriptions

2.2.1 Timeline Module

2.2.1.1 User Interface

The timeline module contains a timeline where a the timeline owner can create or comment on or reply to comments on posts. When a post is commented on the comment is indented. Replies to comments are also indented. Posts are sorted chronologically with the most recent at top.

Timeline Visibility Feature

The timeline owner can set the timeline visibility by using the timeline visibility button located at the top of the timeline on the timeline menu bar. When selected a drop-down menu with the options ‘User’, ‘Friends’, ‘Everyone’ appears. The currently selected option is greyed out and unselectable. When a selection is made the user is informed that the change has been applied.

Timeline Post Visibility

The timeline owner can also set the individual post visibility as well by clicking the visibility button on the top menu of a timeline post. The options ‘User’, ‘Friends’, ‘Everyone’ appears. The currently selected option is greyed out and unselectable. When a selection is made the user is informed that the change has been applied.

Timeline Posting Rights

The timeline owner can change posting rights by clicking the ‘Posting Rights’ button on the timeline menu bar at the top of the timeline. The options ‘User’, ‘Friends’, ‘Everyone’ appears. The currently selected option is greyed out and unselectable. When a selection is made the user is informed that the change has been applied.

2.2.1.2 Service Calls

Exclude due to time

2.2.2 Resume Module

2.2.2.1 User Interface

The resume module consists of a single page. The page will begin as a mostly blank page not including the sites natural navigation bars and menu. If the page is empty this means that the current user does not have a resume attached to their account. If their is no resume the will be an option on the screen to upload new resume.

If a resume is currently attached to the account when the page is navigated to, the appropriate resume will be displayed on the screen. In addition to this there will be a new option on the bottom of the page offering the ability to delete the current resume.

2.2.3 Private Messaging Module

2.2.3.1 User Interface

The Private Messaging Module consists of a single page. The page displays a list of the current user’s friends along the left side and a chat box on the right. The list consists of the username for that friend. The chat box consists of a scrollable box containing all previously sent messages between the two users and a text field below for inputting new messages. Clicking on a friend in the list will open a conversation with them inside the chat box and display all past messages between that friend and the current user, replacing the past conversation.

2.2.4 Lost and Found Module

2.2.4.1 User Interface

The Lost and Found module will contain an option for the user to create a post just as he would in their timeline but instead in this case they will be presented with the option to add a marker on a map to pinpoint the location of the lost item. The user can set the privacy of the post as any post created. The user can be contacted after by phone number (if provided by the original poster) or by private messaging.

2.2.5 Polls Module

2.2.5.1 User Interface

The poll module will be centralized around one page. Each user can navigate to the poll page via the menu/tab option on the over arching website UI. On this page the user will be able to see two sections, the first section will be located on the top of the page and will show the polls that were created by the user, also in this section there will be an option to create a new poll this will navigate to a new page. The second, lower section on the main poll hub will present all the polls that were created by others users but are relevant to the current user.

On the *create a poll page,* the user will be presented with a search bar that they can use to search for relevant courses to create a new poll about. The options will appear below the search bar after searching. After a topic has been selected the user will be presented with five fields that the user can fill in with voting options for the poll, not all fields need to be filled in to create a poll. On the bottom of this page there will be a button labeled create when the user is done creating the poll.

2.2.5.2 Creating a Poll

The creation of a new poll will be handled on a separate page from the main poll page. The user will be able to search for a course as a topic from a list of all courses. They will have the option to enter up to five voting choices for the poll but not all have to be used minimum of two option fields must be filed in for the system to allow the creation of the poll.

2.2.5.3 Voting on a Poll

Polls will only be visible to a user if the topic of the course matches one of courses they have on their schedule (see Schedule Module).The polls will be presented as bar graphs with radio buttons next to each bar that a user can click on to vote on that particular option. Once clicked the user can no longer vote on that poll and their vote cannot be changed.

2.2.6 Friends Module

2.2.6.1 User Interface

The Friends Module consists of two parts. The first part is a recommended friends list and the second is a list of current friends.

The recommended friends list consists of a list of users the system has provided. For each user a name and a smaller list of mutual friends is shown. In addition there is an add friend button beside each user on the list. Click this add friend button will result in a confirmation dialogue saying “Friend Request sent successfully”.

The latter list consists of a list of all users who the current user is friends with. For each user their name is shown.

2.2.7 Study Groups Module

2.2.7.1 User Interface

The Study Group module will contain the aspect of an area where users can communicate in a more group friendly environment. They will we allowed to create posts, upload attachments and comment on posts as they would be allowed throughout the interface. There can be a private and a public group. Private groups may have more than one Administrator and to be a part of a group every member must accept their invite.

2.2.8 Account Module

2.2.8.1 User Interface

The users interface for the Account Module consists of three distinct parts. The first is an account creation page, the second is an email verification page and the final part is a login page.

The account creation page consists of a single form with multiple fields. Those fields are username, email, password and reenter password. A submit button is located at the bottom for use when the form is filled out. If the form is erroneously done then clicking the submit button will return a dialogue informing the User of the error.

The email confirmation page consists of a single field and submit button. Upon inputting a given code into the field and submitting the user is either shown an error dialogue if the code was wrong, or sent to the user’s timeline page if the code was correct.

The final login page is like the others simply a form and button. The form consists of an email or username field and a password field. Filling out the form and clicking the login button will either return an error dialogue if the information is incorrect or redirect the user to their timeline page if the information entered is correct.

2.2.9 Schedule Module

2.2.9.1 User Interface

This module consists of three pages. When navigated to the page will initially be empty if the user has not previously added any classes to their schedule, not including the sites sidebar and top bar used for navigation. There will be an option on this page to add a new class to their schedule which will bring the user to a separate page.

The initial page, if the user has already added a course to their schedule, will display a timetable displaying their courses in the respective slots and the add new course option will be on the bottom of the page, in addition to this there will a new option to remove an existing course.

The add a new course age will present a search bar to the user to search for the desired course. The remove a course page will display a list of all courses the user currently has in their schedule with a button next to them labelled “remove”.

2.2.9.2 Add A New Course to Schedule

On the add a new course page the user will be able to search for the desired course, via search bar, through a database of available courses. In addition to this there will be generalized course topic that the user can choose from such as COMP or MATH that will bring the user to a list of courses in that subject. In either scenario each option in the list will have a button labelled “add” next to it. Clicking it will add the course to the schedule.

2.2.9.3 Remove a Course from Schedule

Clicking on the remove course option will bring the user to a list of their already selected courses, each option will have a remove button next to it. Clicking the button removes the course from the list and their schedule, no message to confirm will be presented to the user.

3.0 Testing

This section exists to detail the testing plan for the project. It begins with information about testing architecture, proceeds into different intended testing methods and follows up with a detailed list of test cases.

3.1 Software Development LifeCycle Architecture

For this project there are three main environments to handle the software development lifecycle. They are;

- Development (DEV)
- Staging (STAG)
- Production (PROD)

Development is for the development of code. Modules that have passed a set of unit tests are pushed to DEV. Staging is where Release Candidates are initially deployed and integration and regression testing are performed. Production is where a release is deployed for client use.

3.1.1 Unit Testing

The first stage in testing. Unit tests are run locally. All unit tests must be passed before any code is pushed to DEV. Ideally this handles bugs that are specific to a single part of the project.

3.1.2 Integration Testing

The second stage of testing. Here modules are tested as a group. This is intended to catch bugs in the interactions between disparate parts of the system.

3.1.3 Regression Testing

The final stage of testing. Previously tested and working code is tested again to ensure that the addition of other modules or any code changes do not cause any bug for the module in question.

3.2 Test Cases

3.2.1 Test Cases Overview

This section gives the test suite for each module. Ideally test cases should be as modular as possible. If a test case fails we should know the failure from the test case name alone. If not the individual test may be too broad.

These test cases are built from the use cases we wrote in the SRS document. If any test fails it will be marked red.

Some test cases are left blank since they are intended to be used for actual testing. The fields along with descriptions are listed below.

- **Actors**: The actors involved in the test case. The actors involved in all of our test cases are listed and described in the table below. Each test case references this table and does not actually describe the actor.
    - Table 1.1 Test Case Actors

Actor Name	**Description**
<User>	a MUNSN user who is logged into their account
<Guest>	A person viewing the MUNSN site but not logged into any account.
<Friend>	Considered to be a Friend of <User>

- **Pre-conditions**: List any pre-conditions the system and testers environment must meet in order for the test to proceed. It is possible for the tester to not even be able to run the test if they cannot achieve the pre-conditions and the test will fail. If a test case fails due to not meeting the pre-conditions this should be noted in the comments and the overall result should be marked FAIL.
- **Steps**

    - **#**: the step number. Test case steps are to be run sequentially.
    - **Action**: describes to the tester what action to take take for this step.
    - **Expected Result**: Describes to the tester what the expected result of the action they took in this step is intended to be.
    - **Actual Result**: The tester records the actual result of the test case here. This is a mandatory field whether a step passes or fails. For the purposes of this document this field is left blank. It will be used during actual testing. For example if a tester if testing an expected result that a file picker appears and it does not, he would record in the actual result field ‘File picker does not appear.’.
    - **Pass Fail**: here a tester records whether a step passes or fails. This is a mandatory field during testing. A tester records PASS or FAIL.
- **Comments**: an optional field where a tester can record any comments.
- **Overall Result**: The overall result of the test. This is a mandatory field during testing. A tester should mark this field PASS or FAIL. If a single step fails then the entire test case fails.

Put the error warnings testing in a separate test case. Please see TC-TL-05.

The fields Actual Result and Pass/Fail are used for actual testing and reporting and logging of bugs. They will be left blank for this document.

If a step fails. Then an entire test case fails.

3.2.1 Timeline Module

3.2.1.0 TC-TL-00 the sample one.

Actors

1. <User>
2. <Guest>

Pre-Conditions

1. Precondition 1 - state none. If you have no pre-conditions
2. Precondition 2 - no need to list logged into account. Its actor 1.

Test Case Steps

#	Step Description	Expected Result	Actual Result	Pass Fail
1	Step 1 description	Expected result of step 1. The thing that must happen for this step to pass.	Actual result - used for actual testing	Pass or Fail
2	Step 2 descr	Expt result of step 2		

3	Step 3 descr	Expt result of step 3		
4	Step 4 descr	Epxpt		

Comments

Overall result

Pass or fail.

3.2.1.1 TC-TL-01 View a timeline

Actors

1. <User>

Pre-Conditions

1. <User> has visibility and posting rights on another <User>'s timeline.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> views their own timeline.	Options to change timeline visibility, change individual post visibility, comment on a post and change posting rights are visible and not greyed out.		
2	<User> views someone else's timeline	Options to change timeline visibility, change individual post visibility, comment on a post and posting rights are greyed out or not visible.		

Comments

<>

Overall Result

<>

3.2.1.2 TC-TL-02 Create a Post on own Timeline

Actors

1. <User>

Pre-Conditions

1. None

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks create post button on the timeline menu.	Create post dialog box pops up. Post button is initially greyed out. Attach media button is visible.		
2	<User> types text in the dialog box.	Post button is no longer greyed out		
3	<User> clicks post button	Post is created on the user's timeline. This new post appears at the top as posts are sorted chronologically. The poster name and post date is visible and correct. Post is present in database.		

Comments

<>

Overall Result

<>

3.2.1.3 TC-TL-03 Create a Post on someone else's Timeline

Actors

1. <User>

Pre-Conditions

1. <User> is viewing another <User>'s timeline and they have rights to post on this timeline.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks create post button on the timeline menu.	Create post dialog box pops up. Post button is initially greyed out. Attach media button is visible.		
2	<User> types text in the dialog box.	Post button is no longer greyed out		
3	<User> clicks post button	Post is created on the user's timeline. This new post appears at the top as posts are sorted chronologically. The poster name and post date is visible and correct.		

		Post is present in database.		

Comments

<>

Overall Result

<>

3.2.1.4 TC-TL-04 Attach Media to A Post

Actors

1. <User>

Pre-Conditions

1. <User> is logged into their own timeline and the create post dialog is open. Attach media button is visible.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks attach media button.	A file picker appears.	File picker does not appear	FAIL
2	<User> selects an image file type that is an allowable type ie. jpg and is less than 200 MB.	Image appears on create post dialog box. The Post button on the create post dialog is no longer greyed out.		
3	<User> selects a video with a size greater than 200 MB. Video file type is an allowable type ie avi.	Video Link appears on create post dialog box.		
4	<User> types some text in create post dialog box.	Text appears in create post dialog box.		
5	<User> hits post button in create post dialog box.	Post is created on timeline. The image and video link are visible. Post and media is present in database.		

Comments

<>

Overall Result

<>

3.2.1.5 TC-TL-05 Attach Media to A Post - Error Warnings

Actors

1. <User>

Pre-Conditions

1. <User> is viewing their own timeline and the create post dialog is open. Attach media button is visible.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks attach media button.	A file picker appears.		
2	<User> selects a file type that is not an allowable type ie. .xml file.	Error message appears alerting <User> that .xml is not a valid file type and gives the list of valid file types for images and videos.		
3	<User> selects a video with a size greater than 200 MB.	Error message appears alerting <User> that the max file size for uploads is 200MB.		
4	<User> selects a image with a size greater than 200MB.	Error message appears alerting <User> that the max file size for uploads is 200MB.		
5	<User> selects 11 media files for upload.	Error message appears alerting <User> that the max number of media files per post is 10.		
6	<User> types more than 5000 characters.	Error message appears alerting <User> that the max number of characters is limited to 5000.		

Comments

<>

Overall Result

<>

3.2.1.6 TC-TL-06 Comment on a Post

Actors

1. <User>

Pre-Conditions

1. <User> is viewing their own timeline and their is at least at least one post.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks the comment button on a post.	The comment on post dialog box pops up. Post button is initially greyed out.		
2	<User> types some text.	Text appears in dialog box. The post button is no longer greyed out.		
3	<User> clicks the attach media button.	A file picker appears.		
4	<User> selects an image using the file picker.	Image appears in dialog box.		
5	<User> selects a video using the file picker.	Video link appears in the dialog box.		
6	<User> hits the post button	Comment appears in the timeline under the selected post. It is clear it is a comment since it is indented. Comment is present in database.		

Comments

<>

Overall Result

<>

3.2.1.7 TC-TL-07 Edit a Post Comment

Actors

1. <User>

Pre-Conditions

1. <User> is viewing their own timeline.
2. There is a least one post with a comment on the timeline. <User> made both the post and the comment.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks the edit button on a post comment.	Edit comment dialog box pops up. Post button is initially greyed out.		
2	<User> types some text in the dialog box.	Text appears in dialog box. Post button is no longer greyed out.		
3	<User> clicks the post button.	Dialog box disappears and		

		post comment text changes to reflect the new changes. View comment history button is no longer greyed out. Comment edition is present in database.		

Comments

<>

Overall Result

<>

3.2.1.8 View Post Comment History

Actors

1. <User>

Pre-Conditions

1. <User> is viewing their own timeline.
2. There is at least one post with a comment that has been edited on the <User>'s timeline. The <User> has made the post, the comment, and the comment edit.
3. The history button is visible and not greyed out.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks history button on a post comment.	A message box appears showing the revision history. Each new edition is indented. The text is correct as well as the poster name and date time stamp.		
2	<User> hits Close button on message box	Message box disappears and timeline is again visible.		

Comments

<>

Overall Result

<>

3.2.1.9 TC-TL-09 Timeline Visibility - Guest

Actors

1. <User>
2. <Guest>

Pre-Conditions

1. There is a <User> who has timeline visibility set to ‘Everyone’. There is at least one post one this user’s timeline.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<Guest> navigates to <User>’s timeline.	Timeline posts are visible to <Guest>.		
2	<User> clicks timeline post visibility on the timeline menu bar.	The following options are available on the drop down menu. 1. Everyone 2. Friends 3. User		
3	<User> selects ‘User’ option from the dropdown menu	<User> sees message confirming that a change to timeline visibility has been made. From a <Guest> perspective all posts disappear without a page refresh and only a blank timeline is visible. Visibility change is reflected in database.		
4	<User> clicks timeline post visibility	All options described in step 2 are again available.		
5	<User> selects ‘Everyone’ from the drop-down list.	<User> sees a message confirming that a change has been made to timeline visibility. From a <Guest> perspective all posts reappear without a page refresh. Visibility change is reflected in database.		

Comments

<>

Overall Result

<>

3.2.1.10 TC-TL-10 Timeline Visibility - Friend

Actors

1. <User>
2. <Friend>

Pre-Conditions

1. There is a <User> with timeline visibility set to ‘Friends’.
2. <User> has at least one post on his timeline.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<Friend> navigates to <User>’s timeline.	Timeline posts on <User>’s timeline are visible to <Friend>		
2	<User> sets timeline visibility to ‘User’	For <User> confirmation that timeline visibility has been changed is displayed. For <Friend>, timeline posts disappear without a page refresh.		
3	<User> sets timeline visibility to ‘Everyone’	For <User> confirmation that timeline visibility has been changed is displayed. For <Friend>, timeline posts reappear without a page refresh.		

Comments

<>

Overall Result

<>

3.2.1.11 Individual Post Visibility - Guest

Actors

1. <User>
2. <Guest>

Pre-Conditions

1. <User>’s timeline has at least one post.
2. The individual post visibility setting for a post on <User>’s timeline is set to ‘Everyone’ (the default setting).

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<Guest> navigates to <User>’s timeline.	<Guest> sees the post on <User>’s timeline with individual visibility set to ‘Everyone’.		
2	<User> clicks the ‘Visibility’ button on the post menu bar	The following options are available on the drop down menu. 1. Everyone 2. Friends 3. User		
3	<User> selects ‘User’ from	For <User> a confirmation message		

	the drop-down menu.	confirming that a change has been made to individual post visibility has been made. For <Guest> individual post referred to in step 1 disappears without a page refresh. The individual post visibility change is reflected in the database.		
4	<User> changes individual post visibility to ‘Everyone’.	For <User> a confirmation message confirming that a change has been made to individual post visibility has been made. For <Guest> individual post referred to in step 1 reappears without a page refresh. The individual post visibility change is reflected in the database.		

Comments

<>

Overall Result

<>

3.2.1.12 Individual Post Visibility - Friend

Actors

1. <User>
2. <Friend>

Pre-Conditions

1. There is at least one post on <User>’s timeline with individual visibility set to ‘Everyone’ (the default setting).

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<Friend> navigates to <User>’s timeline.	The post with individual visibility set to ‘Everyone’ is visible.		
2	<User> sets individual post visibility to ‘User’.	For <User> there is a message confirming that individual post visibility has been changed. For <Friend> the post referred to in step 1 disappears.		
3	<User> sets individual post visibility to ‘Friends’.	For <User> there is a message confirming that individual post visibility has been changed. For <Friend> the post referred to in step 1 reappears.		

Comments

<>

Overall Result

<>

3.2.1.13 Individual Post Visibility does not get Overwritten by Timeline Post Visibility

Actors

1. <User>
2. <Guest>

Pre-Conditions

1. <User> has at least one post on their timeline with individual post visibility set to ‘Everyone’ (the default setting).

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<Guest> navigates to <user>’s timeline.	<Guest> sees the post with individual post visibility set to ‘Everyone’ on <User>’s timeline.		
2	<User> sets timeline visibility to <User>	For <User> a message confirming that timeline visibility has been changed is displayed. For <Guest> there is no change as timeline visibility settings should not overwrite individual timeline post visibility settings.		

Comments

<>

Overall Result

<>

3.2.2 Resume Module

3.2.2.1 Uploading a Resume

Actors

1. <User>

Pre-Conditions

1. None

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> navigates to the resume section of their account via tab/menu option..	<User> is redirected to the appropriate page displaying their resume if it exists. If there is no resume provide a message prompting to upload new resume.		
2	<User> clicks on the upload new resume button on current page.	File browser opens for the <User> to select their resume file.		
3	<User> picks a .pdf file to upload	System accepts and receives .pdf file.		
4	<User> picks a non-.pdf file to upload.	System rejects file and sends message to user that the file type is not supported.		

Comments

1. Under unforseen circumstances the file may fail to upload, in this case error handlers will notify the <User> that the file failed to upload.

Overall Result

<>

3.2.2.2 Deleting a Resume

Actors

2. <User>

Pre-Conditions

2. <User> has an existing resume on their account.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> navigates to the resume section of their account via tab/menu option..	<User> is redirected to the appropriate page displaying their resume.		
2	<User> clicks on the delete resume button on current page.	Display message to the <User> asking if they are sure they want to delete the existing resume.		
3	<User> chooses “yes” to delete resume	System removes the resume file from the database, refreshes the page which no displays no resume.		
4	<User>chooses “cancel” to delete resume	Action is cancelled and <User> is returned to the resume page which displays the current resume.		

Comments

1. Under unforseen circumstances the file may fail to be removed, in this case error handlers will notify the <User> that the file failed to be deleted.

Overall Result

<>

3.2.3 Private Messaging Module

3.2.3.1 TC-PM-01

Actors

1. <User>
2. <Recipient User>

Pre-Conditions

1. Both users must be confirmed Friends

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> navigates to Messaging page view menu item.	A page with a list of Friends is shown.		
2	<User> selects <Recipient User> from list of Friends.	A text chat is shown. If there are any prior messages they are displayed, otherwise the chat is blank.		
3	<User> types a message in the chat text prompt and hits send.	The message is displayed in the chat box for both <User> and <Recipient User>.		
4	(Optional) <Recipient User> or <User> enter further messages into text prompt and send.	Any messages sent are added to the text chat and displayed accordingly for both users.		

Comments

1. Steps can be repeated infinitely to have complete conversations between users.

Overall Result

<>

3.2.4 Lost and Found Module

3.2.4.1 Create Lost and Found Post

Actors

<User>

Pre-Conditions

<User> must have found a lost item which requires a Lost and Found Post.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks create post button on the timeline menu.	Create post dialog box pops up. Post button is initially greyed out.		
2	<User> types text in the dialog box.	Post button is no longer greyed out		
3	<User> clicks on ‘Map’	This will open a small map with its current location.		
4	<User> selects Location of lost item	This will create a marker on the location		
5	<User> clicks post button	Post is created on the user’s timeline. This new post appears at the top as posts are sorted chronologically. The poster name and post date is visible and correct. Post is present in database.		

Comments

A <User> can create a lost and found post and set the privacy to “User” and only he’d be able to see the post.

Overall Result

<>

3.2.4.2 View Lost and Found Post

Actors

<User>

Pre-Conditions

<User> must have a Lost and Found Post or friends who created a Lost and Found Post

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks “Lost and Found” button on the Navigation Bar.	This will direct the User to a page with all the Lost and Found Posts created by his friends.		

Comments

<>

Overall Result

<>

3.2.5 Polls Module

3.2.5.1 Creating a New Poll

Actors

1. <User>

Pre-Conditions

1. None

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> navigates to the poll section of the website via poll tab/menu option.	A page dedicated to polls relevant to the <User> will be displayed.		
2	<User> chooses option to create new poll located at the top of the page.	The <User> will be brought to a separate page dedicated to creating new polls including a list of classes they can make the topic of the poll.		
3	<User> chooses the topic (class) the poll will be about and enters the voting options into the poll then pushes the create button.	The system accepts the information and constructs the poll. The <User> will be returned to the previous page displaying polls relevant to them with the new poll under a section labeled as “My Polls”.		

Comments

The topic of the poll is limited to a list of classes the <User> has in their schedule. If no classes are in the schedule they will be able to only create a poll with a “general” topic.

Overall Result

3.2.5.2 Voting on a Poll

Actors

2. <User>

Pre-Conditions

2. None

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> navigates to the poll section of the website via poll tab/menu option.	A page dedicated to polls relevant to the <User> will be displayed in a manner similar to the timeline interface with two section being “My Polls” and “Others Polls”.		
2	<User> votes on a poll via radio buttons on the interface.	The system accepts the vote and updates the result of the poll which will change the visual representation of the poll.(i.e bar graph, pie chart etc…)		

Comments

Overall Result

3.2.6 Friends Module

3.2.6.1 TC-FR-01 View Friend Recommendations

Actors

1. <User>

Pre-Conditions

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> navigates to the Friend Recommendations page via menu option.	A page containing a list of recomended users for the <User> is displayed.		

Comments

Overall Result

3.2.6.2 TC-FR-02 Send Friend Request

Actors

1. <User>
2. <Recipient User>

Pre-Conditions

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> selects “add friend” option on another user’s profile page.	Confirmation dialogue for sent friend request is shown. For <Recipient User> a prompt for confirmation is shown.		
2	<Recipient User> accepts or declines “friend request”.	Confirmation dialogue for accept or decline is shown for both <Recipient User> and <User>. If accept then both user will be added to the other’s “friend list”.		

Comments

<>

Overall Result

<>

3.2.6.2 TC-FR-03 View Friend List

Actors

1. <User>

Pre-Conditions

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> navigates to Friend List page via menu option.	<User> is shown page with list of all other users who have mutually accepted a Friend Request.		

Comments

<>

Overall Result

<>

3.2.7 Study Groups Module

3.2.7.1 Create a Study Group - Public Group

Actors

<User>

Pre-Conditions

None.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks on ‘Create Group’ button	This will lead the user to another page where he is presented with what to do next.		
2	<User> Names the group	The name of the group is created by the user.		
3	<User> selects privacy of group	The user can selected a public group.		
4	<User> invites friends to join group	The users selected friends will be sent an invitation to the group.		
5	<User> can add a description to the group	The group description is optional to but if added by user it will describe the purpose of the group.		
6	<User> can select if specific members are administrators	The user can chose to make specific friends as administrators only if the group is private.		
7	<User> clicks done	The group is now created.		

Comments

<>

Overall Result

<>

3.2.7.2 Create a Study Group - Private Group

Actors

<User>

Pre-Conditions

None.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks on 'Create Group' button	This will lead the user to another page where he is presented with what to do next.		
	<User> Names the group	The name of the group is created by the user.		
2	<User> selects privacy of group	The user can select between a private group.		
3	<User> invites friends to join group	The users selected friends will be sent an invitation to the group.		
	<User> can add a description to the group	The group description is optional to but if added by user it will describe the purpose of the group.		
4	<User> can select if specific members are administrators	The user can chose to make specific friends as administrators of the group.		
5	<User> clicks done	The group is now created.		

Comments

<>

Overall Result

<>

3.2.7.3 Invite to Study Group

Actors

<User>

Pre-Conditions

<User> is friends with <User1> who created study group.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> clicks on Invite	This will show the user what exactly the		

	Notification	invite is about.		
2	<User> accepts invitation	The user is now a part of that group.		
3	<User> declines invitation	The user is not a part of the group		

Comments

<>

Overall Result

<>

3.2.8 Account Module

3.2.8.1 TC-AC-01 Create and Confirm Account

Actors

1. <User>

Pre-Conditions

1. None

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> inputs account information on account creation page.	The inputted data is verified as it is entered. Incorrectly filled fields show a notification of any errors.		
2	<User> submits account form by clicking the submit button.	A confirmation page is shown and a verification email is sent to the email entered by the user in step 1.		
3	<User> checks email and navigates to the link given within.	A account verification page is shown.		
4	<User> user confirms email on account verification page.	A confirmation is shown and the user is brought to the login page.		

Comments

<>

Overall Result

<>

3.2.9 Schedule Module

3.2.9.1 Adding a Class to a Schedule via Suggested Courses

Actors

1. <User>

Pre-Conditions

1. None

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	<User> navigates to the schedule section of the site via tab/menu option.	The schedule page will be displayed to the <User>, if a schedule already exists it will be displayed on this page otherwise a blank schedule will be displayed.		
2	<User> clicks on “add a new class” button on the bottom of the page.	The <User> will be redirected to a new page with a search bar and a list of course subjects.		
3	<User> clicks on one of the suggested course subjects.	A list of classes of that course subject are displayed to the <User>. A button next to the listing will say “Add”.		
4	<User> clicks on the add button next to a listing.	The system will add that course to the schedule of the <User> and redirects them back to the original schedule screen which now displays the updated schedule.		

Comments

<>

Overall Result

<>

3.2.9.2 Adding a Class to a Schedule via Searched Courses

Actors

2. <User>

Pre-Conditions

2. None

Steps

#	Action	Expected Result	Actual	Pass

			Result	Fail
1	,<User> navigates to the schedule section of the site via tab/menu option.	The schedule page will be displayed to the <User>, if a schedule already exists it will be displayed on this page otherwise a blank schedule will be displayed.		
2	<User> clicks on “add a new class” button on the bottom of the page.	The <User> will be redirected to a new page with a search bar and a list of course subjects.		
3	<User> types the name of a subject or course into the search bar	A list of classes of that match the search criteria are displayed to the <User>. A button next to the listing will say “Add”.		
4	<User> clicks on the add button next to a listing.	The system will add that course to the schedule of the <User> and redirects them back to the original schedule screen which now displays the updated schedule.		

Comments

<>

Overall Result

<>

3.2.9.3 Removing a Class from the Schedule

Actors

3. <User>

Pre-Conditions

3. The <User> has a non-empty schedule.

Steps

#	Action	Expected Result	Actual Result	Pass Fail
1	,<User> navigates to the schedule section of the site via tab/menu option.	The schedule page will be displayed to the <User>, if a schedule exists it will be displayed here.		
2	<User> clicks on the remove a class from the bottom of the page.	The <User> will be redirected to a new page with a list of courses they have already chosen to be on their schedule presented as top to bottom alphabetically. Each will have a button next to it labelled “remove”		
3	<User> clicks on one of the remove buttons next to one of the courses.	The system removes that course from the <User’s> list of courses and updates the list being displayed to the user. The <User> remains on this page.		

4	,<User> navigates to the schedule section of the site via tab/menu option.	The schedule page will be displayed to the <User>, with an updated schedule reflecting the changes the <User> just made.		

Comments

<>

Overall Result

<>

3.4 Web Service Call Testing

There will be a back-end testing page on the DEV and STAGE servers, but not on the PROD server. This is a simple html page that takes user input and sends a url to the web service. The service sends back the JSON response which is displayed to the user. It takes GET, PUT and DELETE requests and displays the JSON response for all service calls.
This is ideal for testing the back-end in total isolation from the front-end. When there is a bug with the data being received from the back-end, by examining the response from the back-end test page you can determine if the problem is with the back-end itself or an issue with parsing on the front-end.

3.5 Non-Functional Testing

Think we have these in the test cases section. Probably easier just to roll it all into one.

4.0 Document Responsibilities

- Allan
  - Document layout and structure
  - 2.2.1 Timeline Module Interface Description
  - 3.2.1 Timeline Module Test Cases
  - 4.0 Document Responsibilities
  - 6.0 Schedule
  - 3.2.1 Test Case Overview
  - 3.2 Software Development Lifecycle
  - 3.4 Web Service Call Testing
- Saahil
  - 2.2.4 Lost and Found Interface Description
  - 2.2.7 Study Group Interface Description
  - 3.2.4 Lost and Found Test Cases
  - 3.2.7 Study Group Test Cases
- Mark
  - 2.2.2 Resume Interface Description
  - 2.2.5 Polls Interface Description

        - 2.2.9 Schedule Interface Description
        - 3.2.2 Resume Test Cases
        - 3.2.5 Polls Test Cases
        - 3.2.9 Schedule Test Cases
- Tyler
    - Document section introductions
    - 2.2.3 Private Messaging Interface Description
    - 2.2.6 Friends Interface Description
    - 2.2.8 Account Interface Description
    - 3.2.3 Private Messaging Test Cases
    - 3.2.6 Friends Interface Test Cases
    - 3.2.8 Account Interface Test Cases
    - 5.0 Coding Effort and Priority

5.0 Coding Effort and Priority

This section consists of a table outlining the full list of modules and ranking them based on priority and effort required. Order is ascending, with the most effort and highest priority starting at 1.

Module Name	Coding Effort	Priority Rank
Timeline Module	1	2
Resume Module	8	8
Private Messaging	2	9
Lost and Found	4	7
Polls	7	6
Friends	6	4
Study Groups	5	3
Account Module	9	1
Schedule	3	5

6.0 Schedule

Week 09 Feb-27 to Mar-05

- Document work
    - structure and layout.
- Back-end work
    - Nodejs-Express web server can serve pages and media, respond to GET and POST requests, route web-server requests.

Week 10 Mar-06 to Mar-12

- Document work
    - Test case section.

Week 11 Mar-13 to Mar-19 (Planned)

- Deadlines
    - Milestone 3 - Module Documentation Document due Mar-17
- Front-end work
    - Master MVC model is set up.
    - Html skeletons coded for all modules. (No service calls - buttons do not do anything)

Week 12 Mar-20 to Mar-26 (Planned)

- Back-end
    - Set up Mongodb database
    - Code all service calls
    - Code calls to populate front-end components like drop-down menus, list of timeslots, etc.
- Front-end
    - Set up service calls.
- Testing
    - Code is peer reviewed.
    - First release candidate is deployed to STAGE.
    - Unit tests, Regression Testing, Integration Testing for Release candidate one.
    - Bugs are logged and bug fixes are assigned. Each bug fix must pass all unit tests before being deployed to DEV.
    - Another round of unit tests, Integration Testing, Regression Testing for release candidate two.
    - Repeat as needed.

Week 13 Mar-27 to Apr-02 (Planned)

- Testing
    - further testing, bug fixes as required.
- Deployment
    - Code is tested deployed and working.

Week 14 Apr-03 to Apr-05 (Planned)

- Deadlines
    - Milestone 4 - Final Presentation