

Intro/Preface.....	1
Glossary	1
Note on Prioritization.....	1
Use Cases by Stakeholder/Priority.....	2
Priority 1: Join/Leave Course	2
Priority 2: BBS + Database setup	4
Priority 3: File Upload	4
Priority 4: Study Groups.....	5
Priority 5: Misc. features	6
Requirements by Stakeholder.....	11
Acceptance Tests	12
Blah	12

Intro/Preface

Glossary

Note on Prioritization

- 1) Join Class (Join pre existing, Remove) Add for other iterations
Uni, Term, Class #
Class page - list of people, wall, ... Basic FB Group
- 2) Bill Board
Posting, Chronology, Comments
- 3) File Upload (Notes)
Stats, (Rating), Comments, Sort
- 4) Study Group
Invite Sys, Class Page w/ visibility
- 5) Ratings of Notes, Class Creating, Tags (searching), report abuse, tracking updates, calender
Need sink

Use Cases by Stakeholder/Priority

Priority 1: Join/Leave Course

Stakeholder's Need: 1.0.1

Use Case: Add application to the user's Facebook account.

Actor: Facebook user

Preconditions: The end user is registered with an active Facebook account.

Details: The user would click on a linked button that says 'Add application'.

The user would then be redirected to the application's main page, which lists of courses this user has joined.

Postconditions:

After confirmation to add the application, the application will appear to be located on the user's application page listings henceforward.

Stakeholder's Need: 1.0.2

Use Case: Add a course to the user's course membership list (Join a course).

Actor: Facebook user

Preconditions: The application has been added to the user's Facebook account.

Details: User either enters a search term to the an existing course, OR selects the university, session, and then course each from a combobox.

The user must click on the course to join it.

The user's information is sent to and added within our custom database, thus that user is now a member of the course.

Postconditions: After choosing the course, the user is sent to the main page, showing him all the courses that user has joined.

Stakeholder's Need: 1.0.10.1

Use Case: View the user's friends within a course.

Actor: Facebook user

Preconditions: The application has been added to the user's Facebook account.

Details: From the main page, the user must click on the link 'View Friend's courses'.

The user is sent to a page showing all of their friends, and the courses that their friends are currently a member of.

I don't see this as a requirement. The user should be able to view all the current members of the course. I propose on revising the requirement to allow the user to see ALL members of the course.

Stakeholder's Need: 1.0.10.2

Use Case: View user's friend's joined courses.

Actor: Facebook user

Preconditions: The application has been added to the user's Facebook account.

Details: From the main page, the user must click on the link 'View Friend's courses'.

The user is sent to a page showing all of their friends, and the courses that their friends are currently a member of.

Stakeholder's Need: **Unlisted**

Use Case: Leaving a course.

Actor: Facebook user

Preconditions: The application has been added to the user's Facebook account.

The user should have joined a course.

Details: From the application's main page (which lists of courses this user has joined), the user would click the 'X' next to the course name.

A message prompt would appear asking the user if this is a desired action.

If the user chooses accept from the prompt, information is sent to the database noting the user is no longer a member of the course.

Postconditions: The application's main page is updated-the user's joined course list is refreshed.

Stakeholder's Need: **Unlisted**

Use Case: Creating a course.

Actor: Facebook user, Administrator

Preconditions: The application has been added to the user's Facebook account.

Details: From the application's main page, the user must click the a link 'Create a new course'.

The user is sent to a blank form, and must input the university, session, course name, and course number.

When finished inputting the data, the user can ether cancel the request (and return to the applications main page) or submit it for approval.

An administrator would then be required to review the request and either accept it or deny it.

A message is sent the user notifying if the course was either accepted or denied.

Postconditions: Upon acceptance from an administrator, the course is now added to the selection of application courses available for all application users to join.

Stakeholder's Need: **Unlisted**

Use Case: Removing the application.

Actor: Facebook user

Preconditions: The application has been added to the user's Facebook account.

Details: The user must navigate to the 'Application Settings' page from the Settings title menu off the standard Facebook page.

The user clicks on the 'X' next to the application that is desired to be removed.

The user confirms to remove this application when prompted with a popup asking to verify this request.

Postconditions: After confirmation to remove the application, the application will no longer appear to be located on the user's application page listings henceforward.

Stakeholder's Need: **Unlisted**

Use Case: View course mates within a joined course.

Actor: Facebook user

Preconditions: The application has been added to the user's Facebook account.
The user should have joined a course.

Details: From the main application page, the user must click on a course link.

The user is sent to the course home page. This page has a section called 'members' that shows all of the current members of the course. It will display their Facebook name, and their default Facebook picture.

Priority 2: BBS + Database setup

x.x.x Use Case: Add a post to the class

Needs:

Actor: User

Preconditions: The user is viewing one of their class's class page

Description:

1. The user enters a message into a text field
2. The user can enter a due date for the message
3. The user submits the message by pressing a 'Share' button
4. The user is directed back to the class page where their posting is displayed

Req Note: Posts are displayed chronologically and threaded

x.x.x Use Case: Add a comment to a class post for threaded discussions

Needs:

Actor: User

Preconditions: The user is viewing one of their class's class page

Description:

1. The user is presented a comment link next to each starting post
2. Upon clicking the comment link, a text field is loaded and presented to the user
3. The user can enter their comment into the field
4. The user submits the message by pressing a 'Comment' button
5. The user is directed back to the class page where their comment can be viewed

~~DELETE following use case???~~

x.x.x Use Case: Sort postings by due date

Needs:

Actor: User

Preconditions: The user is viewing one of their class's class page

Description:

1. The user chooses a due date option from a combo-box
2. The options are the sorting options:
 - Ascending
 - Descending
 - Remove
3. Upon selecting one of the above options the postings are displayed in that order

Priority 3: File Upload

x.x.x Use Case: Uploading Notes

Needs:

Actor: User

Preconditions: The user is viewing one of their class's class page

Description:

1. The user chooses an 'Upload' button from the same form as used in posting
2. The user is presented a browser dependent file browsing window to select their file
3. The results of the file selection are then uploaded on the user hitting 'Share'
4. The user is directed back to the class page where the posting with the note's stub is displayed

x.x.x Use Case: Downloading Notes

Needs:

Actor: User

Preconditions: The user is viewing one of their class's class page

Description:

1. The user can select a note to download by clicking the 'Download' button in a posting's note stub
2. The user is presented a browser dependent file browsing window to select a location to download to
3. The note is downloaded to the folder selected

Priority 4: Study Groups

Use Case:

SN:

Actor:

Preconditions:

Details:

Postconditions:

Study groups

Use Case: Join a study-group

SN: 1.0.8, 1.3.3

Actor: NoteShare user

Preconditions: The user has added the course that has the associated study-group he is interested in joining.

Details: On the course page, the user must click on the name of the study-group he is interested in joining. A message giving a short summary of the study group and it's members is shown, along with a link to 'Add study-group'. If admission to the study-group is restricted, the link instead shows 'Request membership'. The user must click on this.

Postconditions: If unrestricted, the study-group now shows up as a clickable link on the user's list of joined study groups when he accesses the course home-page. If restricted, the group shows up similarly but is not clickable till his membership is approved by the study group.

Use Case: Leave a study group

SN: [!]

Actor: NoteShare user

Preconditions: The user is actually a member of a study-group

Details: The study-group member can leave a group in one of 4 ways:

1. By clicking the 'X' next to the name of the study-group on the course home page
2. By clicking on 'Leave study-group' on the study-group home page
3. Removing the associated course removes the user from all associated study groups
4. Removing the NoteShare application

In all cases, a confirmation message will pop-up to ask the user to confirm his action. [???] Additionally, a check box will be present to ask if all his related posts and uploads are to be deleted too.

Postconditions: The user will no longer be part of the study-group. Add note on [??]

Use Case: Allow a person to enter a study group [?]

Invite people to a study group [don't need?]

Post a message, date on a study group BBS 1.0.9

Create a study group 1.3.1

- Limit membership 1.3.2

Remove a study group

(search study groups? 1.0.5.3)

Priority 5: Misc. features

x.x.x Rate an Uploaded Note	
Version #	1.0.0
Basic Events	1.) User sees list of notes. 2.) User selects plus or minus button. 3.) Selected button is highlighted. 4.) Note rating is updated by +/- 1.
Actors	User
Preconditions	List of notes is displayed on screen. List of notes is a subset of the complete note collection. List of notes is generated by user selected search criteria. List of notes displays plus and minus icons in-line with each note. List of notes displays title of each note.
Triggers	User clicks on plus or minus icons displayed in-line with the note title.
Postconditions	Icon the user selected is highlighted. Rating is updated internally for that note set.
Alternative Paths	User could arrive at a "list of notes" multiple ways: --Personal homepage --Community Billboard --Study group --Course page --Search Results
Stakeholder's Needs	1.0.6.1
Notes: Some sort of clickable rating system is preferred for convenience as this is less cumbersome than a combo box. Also, screen footprint is fairly small.	

Note rating system like this could potentially appear anywhere a note's link is displayed.

x.x.x Rate a Bulletin Board Post	
Version #	1.0.0
<i>Basic Events</i>	1.) User sees a bulletin board post. 2.) User selects plus or minus button. 3.) Selected button is highlighted. 4.) Post rating is updated by +/- 1.
<i>Actors</i>	User
<i>Preconditions</i>	At least one bulletin board post is visible.
<i>Triggers</i>	User clicks on plus or minus icons displayed in-line with the post.
<i>Postconditions</i>	Icon the user selected is highlighted. Rating is updated internally for that bulletin board post.
<i>Alternative Paths</i>	A user could see a bulletin board post in a variety of areas: --Personal Homepage --Community Bulletin Board Page --Course Page
<i>Stakeholder's Needs</i>	1.4.1.1
Notes: Should probably be implemented similar to x.x.x Rate an Upload Note . At the very least, the interface component should be similar for user familiarity and interface consistency.	

x.x.x Create a Course	
Version #	1.0.0
<i>Basic Events</i>	1.) User selects create course button on "add course" page. 2.) User fills in course information. 3.) User selects create course button on "create course" page. 4.) Course is added to database and user's course list.
<i>Actors</i>	User
<i>Preconditions</i>	Course by specific course number is not already present in the database.
<i>Triggers</i>	User selects create a course button on create a course page.
<i>Postconditions</i>	Course is created in database. Course page is created. Course is added to user course list. User is queried to goto new course page.
<i>Alternative Paths</i>	None
<i>Stakeholder's Needs</i>	1.0.2
Notes: The create course button on "add course" page redirects the user to the "create course" page. The create course button on the "create course" page actually	

creates the course.

The create a course page itself should have sufficient input fields to fill a course entry in the course database.

x.x.x Search for Note by Category	
Version #	1.0.0
Basic Events	1.) User selects search "for notes" option. 2.) User selects "by category" option and fills in search phrase. 3.) User hits search button.
Actors	User
Preconditions	Search box is visible. Search "for notes" option is selected. Search "by category" option is selected. Search phrase is filled in.
Triggers	User hits search button.
Postconditions	Search results are displayed on screen.
Alternative Paths	None
Stakeholder's Needs	1.0.5.1
Notes: A search box could be located anywhere on the site, perhaps to deliver context specific searches. A mockup would probably make this clearer.	

x.x.x Search for Bulletin Board Posts by Subject	
Version #	1.0.0
Basic Events	1.) User selects search "for Bulletin Board Posts" option. 2.) User selects "by category" option and fills in search phrase. 3.) User hits search button.
Actors	User
Preconditions	Search box is visible. Search "for Bulletin Board Posts" option is selected. Search "by category" option is selected. Search phrase is filled in.
Triggers	User hits search button.
Postconditions	Search results are displayed.
Alternative Paths	None
Stakeholder's Needs	1.0.5.2
Notes: Again, a search box could be located anywhere on the site. A mockup might make this clearer.	

x.x.x Search for Study Group by Category
--

Version #	1.0.0
Basic Events	1.) User selects search "for Study Group" option. 2.) User selects "by category" option and fills in search phrase. 3.) User hits search button.
Actors	User
Preconditions	Search bar is visible. Search "for Study Group" option is selected. Search "by category" option is selected. Search phrase is filled in.
Triggers	User clicks on search button.
Postconditions	Search results are displayed on screen
Alternative Paths	None
Stakeholder's Needs	1.0.5.1
Notes:	

x.x.x Upload Note by Category	
Version #	1.0.0
Basic Events	1.) User selects upload note button from course page. 2.) User fills in meta information about note set. 3.) User enters tags for note set. 4.) User hits upload note set.
Actors	User
Preconditions	User has selected note set file. Note set file exists on user disk. User has entered at least title for note set. User has entered at least one category tag for note set.
Triggers	User clicks upload note set button.
Postconditions	On screen confirmation of successful upload is displayed. Notes are uploaded into system.
Alternative Paths	None
Stakeholder's Needs	1.0.3
Notes: Potentially, notes could be uploaded from anywhere within the system in a context dependent manner.	

x.x.x Report Abuse	
Version #	1.0.0
Basic Events	1.) User sees list of notes. 2.) User downloads a set of notes for review. 3.) User determines that the note set violates the Terms of Service. 4.) User selected "Report Abuse" button.

	5.) The button is highlighted. 6.) Note set removed after five reports by different users.
<i>Actors</i>	User
<i>Preconditions</i>	List of notes is displayed on screen. List of notes displays report abuse icon in-line with each note. List of notes displays title of each note.
<i>Triggers</i>	User clicks on report abuse icon displayed in-line with the note title.
<i>Postconditions</i>	Icon the user selected is highlighted. Abuse field of note is incremented.
<i>Alternative Paths</i>	User could arrive at a "list of notes" multiple ways: --Personal homepage --Community Billboard --Study group --Course page --Search Results
<i>Stakeholder's Needs</i>	2.1 2.2 2.3 3.1 3.3 3.8
Notes:	

x.x.x Track Updates	
<i>Version #</i>	1.0.0
<i>Basic Events</i>	1.) User sees list of notes/bulliten board posts. 2.) User selects the "Follow" button. 3.) The button is highlighted. 4.) Note set/bulliten board post is added to users homepage.
<i>Actors</i>	User
<i>Preconditions</i>	List of notes/bulliten board is displayed on screen. List of notes/bulliten board post displays follow icon in-line with each note. List of notes displays title of each note.
<i>Triggers</i>	User clicks on follow icon displayed in-line with the note title/ bulliten board post.
<i>Postconditions</i>	Icon the user selected is highlighted. Note/bulliten board post is added to users homepage.
<i>Alternative Paths</i>	User could arrive at a "list of notes" multiple ways: --Personal homepage --Community Billboard --Study group

	--Course page --Search Results
<i>Stakeholder's Needs</i>	1.2.2 1.4.1.5 1.4.1.6
Notes: May be useful to highlight updated sets in some way. Not sure if we are doing email notification.	

x.x.x Calendar	
Version #	1.0.0
<i>Basic Events</i>	1.) User logs into Facebook. 2.) User navigates to NoteShare application. 3.) User is taken to their personal homepage. 4.) User sees a calendar with important deadlines for their class.
<i>Actors</i>	User
<i>Preconditions</i>	User is logged into Facebook. User is within the NoteShare application. User is on their personal homepage. User adds a class.
<i>Triggers</i>	User is on their personal homepage.
<i>Postconditions</i>	N/A
Alternative Paths	User could arrive at a their personal homepage multiple ways: --Initial startup --Clicking home at anytime from any screen
<i>Stakeholder's Needs</i>	1.0.10 1.2.3 1.2.3.5
Notes: Users are also allowed to add their own personal deadlines to their calendar.	

Requirements by Stakeholder

Format

x.x.x Priority	Need y.y.y
<i>Informal Description</i>	
<i>Formal Description</i>	

<i>Pre-Conditions</i>	
<i>Post-Conditions</i>	
<i>Test</i>	

Acceptance Tests

Blah