

## **OBJECTIVE**

My Book is a website that will allow families and close groups of friends to connect to each other through an integrated calendar, gallery, and organized place to pin notes. Currently people do not have an appropriate venue to facilitate real life events online in an organized manner. People often use Facebook as a way to announce events to all of their friends in a casual way. Additionally, Google calendar is used more often as a place to keep personal dates organized. My Book however, is much like Google calendar but it works more like a collaborative schedule organizer for a specific, predefined small groups. All members have access to the calendar and all members can post new events. It also allows the members to post notes and images in separate areas of the book. My Book will satisfy the user's need of online social interaction, while providing an organized layout for sharing daily tasks or schedules.

## **DOMAIN ANALYSIS**

The general domain in which this software will function is the social environments of small groups of people. More specifically, people who wish to coordinate schedules for in-person interaction with some online networking.

The users within the domain are close-knit groups of friends and family. These groups often socialize offline, likely on a semi-regular basis. The members of these groups are not in contact with each other in person on a regular basis, or at least not in quantities that make online interaction needless. The technological skill level of these groups will probably be low on average.

The technological environment of this domain is uncertain as there is no way to tell what systems the users will have. Many of the users in the domain will have access to some form of device with internet browsing capabilities.

The social environment of this domain would primarily be recreational interaction. It will be used by people who want to facilitate venues for socializing, and stay connected between such venues. This conceivably includes groups in certain business environments but this is not the primary domain.

Common tasks in this domain include electronic communication between members of the group, the communication of schedule information, and the sharing of other relevant data.

## **REQUIREMENTS**

1. The login page will accept a username and password, if they are found in the database.
2. The login page will navigate to the home page, if login was successful.
3. The login page will allow new users to navigate to the register page.
4. The register page will allow a new user to either create a new group or join an existing one if it is found in the database.
5. The system will store new usernames , passwords, and unique six digit group Ids in the database.
6. The register page will navigate to the home page, if registration was successful.
7. The home page will display a calendar and links to navigate to gallery, notes, and create-event pages.
8. The home page will display the three most recently uploaded notes, and the three most recently uploaded events.
9. The system will display a list of events happening in the current month on the homepage.

10. The system will allow user's to create events.
11. The create event page will accept an event name, date, time, description, and place.
12. The system will store the new data as an event in the database.
13. The create event page will display a list of all events created, and update the list upon successful creation of an event.
14. The system will allow users to cancel the creation of an event.
15. The system will allow users to delete an event via a delete button next to each listed event on the event creation page. This button will remove the event from the database.
16. The notes page accepts text entered into create new note field.
17. The system will store the new data as a note in the database
18. The notes page displays a list of all notes in the order they were added.
19. The notes page will allow users to delete a note through the use of a delete button placed next to each note. This will delete the note from the database.
20. The gallery page accepts images uploaded.
21. The system will store the new image data as a Binary Large Object in the database
22. The gallery page displays a list of all images, and will provide a delete button if the user viewing the image is the one who uploaded it which will remove it from the database.

## **USE CASES**

### **Use Case: Create a new account, new group, success**

1. User loads index.xhtml.
2. User selects "New User" button.
3. User is directed to register.xhtml.
4. System prompts user to enter:

**Username**  
**Password**

**Confirm Password**

**A check-box which if checked will indicate that the user is attempting to join a group.**

**Group ID (if user is attempting to join a group).**

5. User enters data leaving the checkbox unchecked and the Group ID field with the default value and selects "Submit".
6. System generates new six digit ID numbers until it finds one that is unique.
7. System stores the data user's data and the group ID data in the database.
8. System navigates to home.xhtml.
9. The user's username and group ID are displayed in the top right corner.

**Use Case: Create a new account, fail**

1. User loads index.xhtml.
2. User selects "New User" button.
3. User is directed to register.xhtml.
4. System prompts user to enter:

**Username**

**Password**

**Confirm Password**

**A check-box which if checked will indicate that the user is attempting to join a group.**

**Group ID (if user is attempting to join a group).**

5. User enters data, and selects "Submit".
6. System displays message.

**The username "\_\_\_" has already been used, try again.**

7. User enters new username.
8. User clicks "Submit".
9. System stores the data.
10. System navigates to home.xhtml.

**Use Case: Register, Join Group, Successful**

1. User loads index.xhtml.
2. User selects "New User" button.
3. User is directed to register.xhtml.
4. System prompts user to enter:

**Username**

**Password**

### **Confirm Password**

**A check-box which if checked will indicate that the user is attempting to join a group.**

**Group ID (if user is attempting to join a group).**

5. User enters data and checks the checkbox and enters a six digit number into the Group ID field and selects "Submit".
6. System checks the database to see if the group exists and finds that it does.
7. System stores the data user's data and the group ID data in the database.
8. System navigates to home.xhtml.
9. The user's username and group ID are displayed in the top right corner, as well as any content posted by group members.

### **Use Case: Login, success**

1. User loads index.xhtml.
2. System prompts user to enter:  
**Username**  
**Password**
3. User enters data, and selects "Login".
4. System navigates to home.xhtml.

### **Use Case: Login, fail**

1. User loads index.xhtml.
2. System prompts user to enter:  
**Username**  
**Password**
3. User enters incorrect data, and selects "Login"
4. System displays message:  
**The username or password is incorrect, please try again.**
5. User enters correct data.
6. System navigates to home.xhtml.

### **Use Case: Create a new note**

1. User carries out **Login, success**.
2. User clicks on "new note".
3. System navigates to note.xhtml.

4. System prompts user to enter new note.
5. User types in details of new note including title and body.
6. User selects "Submit".
7. System stores the note data in the database.
8. System navigates to note.xhtml and displays new note into the note list.

**Use Case: Create a new event, success**

1. User carries out **Login, success**.
2. User clicks on the "Create Event" button.
3. System navigates to createEvent.xhtml
4. System prompts user to enter event information.  
User fills out all required fields.
5. User clicks on the "Create Event" button.
6. System stores the data in the Database.
7. System navigates to createEvent.xhtml, which displays the updated event list.

**Use Case: Create a new event, fail**

1. User carries out **Login, success**.
2. User clicks on the "Create Event" button.
3. System navigates to createEvent.xhtml.
4. System prompts user to enter information.
5. User does not fill all required fields.
6. User clicks on the "Create Event" button.
7. System displays the message for each required field which does not have input:

**Please enter information in (fieldname).**

8. User fills out all required fields.
9. User clicks the "Create Event" button.
10. System stores data in the database.
11. System navigates to createEvent.xhtml, which displays the updated calendar.

**Use Case: Upload image to gallery, success**

1. User carries out **Login, success**.
2. User clicks the “Gallery” button on the main page.
3. System navigates to gallery.xhtml.
4. System loads and displays images in sets of nine from the database uploaded by other users in the current user's group (if any) in a list of thumbnail scrollable via buttons.
5. User clicks the “Browse” button
6. System presents user with a file selection dialogue.
7. Users selects image file they want to upload though the file selection dialogue.
8. User clicks the “submit” button on the dialogue.
9. System stores the file in the database.
10. System refreshes gallery.xhtml, presenting the user with an updated view of the uploaded images

#### **Use Case: Upload image to gallery, failure**

1. User carries out **Login, success**
2. User clicks the “Gallery” button on the main page.
3. System navigates to gallery.xhtml.
4. System loads and displays images in sets of nine from the database uploaded by other users in the current user's group (if any) in a list of thumbnail scrollable via buttons.
5. User clicks the “Browse” button.
6. System presents user with a file selection dialogue.
7. Users selects image file they want to upload though the file selection dialogue.
8. User clicks the “Upload” button.
9. System detects that the file is not an image file.
10. System displays the message:

#### **Filetype not supported**

11. User selects another file through the dialogue
12. User clicks the “submit” button

13. System stores the file in the database
14. System refreshes gallery.xhtml, presenting the user with an updated view of the uploaded images

**Use Case: Delete a Note**

1. User carries out **Login, success.**
2. User is presented with a view of the main page.
3. User clicks on the create note link which navigates to that page.
4. User finds the note in the the list of notes presented there they wish to delete, and clicks the delete button next to it.
5. System removes the corresponding note from the database and refreshes the page, displaying an updated list of notes.

**Use Case: Delete an Image**

1. User carries out **Login, success.**
2. User is presented with a view of the main page.
3. User clicks on the gallery link which navigates to that page.
4. User finds the picture that they uploaded, and clicks the delete button.
5. System removes the corresponding picture from the database and refreshes the page, giving an updated view of the images.

**Use Case: Delete an Event**

1. User carries out **Login, success.**
2. User is presented with a view of the main page.
3. User clicks on the create event link which navigates to that page.
4. User finds the event in the the list of events presented there they wish to delete, and clicks the delete button next to it.
5. System removes the corresponding event from the database and refreshes the page, displaying an updated list of events.