



# Chapter 1

## Use Case Descriptions

### 1.1 Authentication (mora5651)

#### 1.1.1 Sign up (mora5651)

---

*Actors:* User.

*Goals:* To register and create an account in sQuire.

*Pre-conditions:* None.

*Summary:* The user signs up and creates an account using their email address and, creates a username and password.

*Related use cases:* None.

*Steps:*

1. User is prompted to enter email, username and password.
2. System sends confirmation email.
3. User verifies email.
4. System saves information, and redirectes user to sign in page.

*Alternative 1:* User already has an account.

*Alternative 2:* User doesn't confirm email. Delete request after timeout period.

### 1.1.2 Sign in (mora5651)

---

*Actors:* Users.

*Goals:* Pre-existing user signs into profile.

*Precondition:* User must already have an account

*Summary:* User wishes to access their account, projects and info.

*Steps:*

1. User is prompted to enter username/e-mail, and password.
2. System verifies information
3. Correct information prompts user to their home page.
4. If user forgot their user name, they can click "Forgot Username button. They then input their email address.
5. System validates their email address with an account, and sends an email with the username
6. If user forgot their password, they click "Forgot password" button. Then they input their email address.
7. System validates their email address with an account, and sends an email to them with a temporary password.

*Alternatives:* Information is incorrect, user tries again. Or makes a new account

---

### 1.1.3 Logout (mora5651)

---

*Actors:* Users.

*Goal:* Existing user logs out

*Precondition:* User must be logged in

*Summary:* The user can log-out of the program at any time.

*Steps:*

1. User clicks the "log-out" button.
2. System prompts user to ensure all unsaved work has been saved.
3. User verifies.
4. System logs user out.

*Alternative 1:* The program will send notification to ask if the user is sure to sign out.

*Alternative 2:* User cancels on step two. Return to home page.

---

#### 1.1.4 Forgotten Account Information (mora5651)

---

*Actors:* Users.

*Goal:* Recover forgotten account information.

*Precondition:* User must already have an account.

*Summary:* User has forgotten their account information, and wishes to recover information.

*Steps:*

1. User clicks the "Forgotten username/password" button.
2. User inputs their email address.
3. System validates their email address with an account, and sends an email with the username or password reset.

System logs  
user out.

*Alternative 1:* Information is incorrect, user tries again. Or makes a new account.

---

## 1.2 Project Ideas (dani2918)

### 1.2.1 Browse Project Ideas (dani2918)

---

*Actors:* User

*Goals:* Examine a list of available project ideas

*Pre-conditions:* User is signed in

*Summary:* User looks through posted project ideas to find projects to work on/discuss

*Related use cases:* Comment on Project Idea, Vote on Project Idea, Share Project Idea

*Steps:*

1. Actor selects Browse Project Ideas button
2. Actor refines search by selecting from list of project categories as desired
3. Actor enters terms into search field as desired and views a list of top projects
4. Actor selects desired project
5. System displays detailed project information

*Alternatives:* None.

*Post-conditions:* None.

---

### 1.2.2 Create Project Idea Thread (dani2918)

---

*Actors:* Project Administrator

*Goals:* Generate public interest in project idea

*Pre-conditions:* Prospective project administrator is signed in

*Summary:* A user with interest in heading up own project can post ideas to get feedback and/or recruit collaborators

*Related use cases:* Manage Project Idea Thread

*Steps:*

1. Actor selects Create Project Idea button
2. Actor enters prospective project title and thoughts and ideas as a description
3. Actor selects Submit button

*Alternatives:* None.

*Post-conditions:* None.

---

### 1.2.3 Manage Project Idea Thread (dani2918)

---

*Actors:* Project Administrator

*Goals:* Respond to feedback and manage project idea

*Pre-conditions:* Prospective project administrator is signed in and has navigated to one of his or her own project threads

*Summary:* A prospective project administrator responds to others' questions/comments

*Related use cases:* Create Project Idea Thread

*Steps:*

1. Actor selects Reply button on a comment
2. Actor types feedback to other user
3. Actor selects Submit button
4. System shows confirmation that feedback was received

*Alternatives:* Actor selects Delete Comment instead of Reply to remove harmful feedback.

*Post-conditions:* None.

---



#### 1.2.4 Comment on Project Idea (dani2918)

---

*Actors:* User

*Goals:* Provide detailed feedback on project ideas

*Pre-conditions:* Actor is signed in, has navigated to a project idea

*Summary:* User provides feedback to or asks questions about a prospective project.

*Related use cases:* Browse Project Ideas, Vote on Project Idea, Manage Project Idea Thread

*Steps:*

1. Actor selects Comment button
2. Actor types feedback into field
3. Actor clicks Submit button
4. System shows confirmation that feedback was received

*Alternatives:* None

*Post-conditions:* None.

---

### 1.2.5 Vote on Project Idea (dani2918)

---

*Actors:* User

*Goals:* Support promising project ideas or offer criticism to unfavorable ones

*Pre-conditions:* Actor is signed in, has navigated to a project idea

*Summary:* User offers support/discourages a project idea so that prospective project administrators get feedback and promising project ideas get publicity

*Related use cases:* Comment on Project Idea

*Steps:*

1. Actor selects and Up Vote or Down Vote button
2. Actor selects Submit button
3. System highlights which button the user has selected

*Alternatives:* None

*Post-conditions:* None.

---

### 1.2.6 Share Project Idea (dani2918)

---

*Actors:* User, Project Administrator

*Goals:* Generate excitement about a project

*Pre-conditions:* Actor is signed in, has navigated to a project idea

*Summary:* Prospective project administrators or users show other users which projects they believe are worthwhile

*Related use cases:* Comment on Project Idea

*Steps:*

1. Actor selects Share button
2. Actor selects an audience and method with which to share selected project
3. Actor selects Submit button
4. System notifies or shows audience the shared project idea

*Alternatives:* None

*Post-conditions:* None.

---

## 1.3 Communication (jank6275)

### 1.3.1 Open project chat (jank6275)

---

*Actors:* User

*Goals:* To open the project chat window.

*Pre-conditions:* User must be registered, signed in, and have a open project.

*Summary:* User opens a project and the project chat automatically opens. The chat window displays chat history and updates when new chat messages are received.

*Related use cases:* Join global chat.

*Steps:*

1. User opens a project.
2. Chat is notified that user has joined.
3. System displays project chat window to the user.

*Alternatives:* None.

*Post-conditions:* None.

---

### 1.3.2 Open global chat (jank6275)

---

*Actors:* User

*Goals:* To open the global chat window.

*Pre-conditions:* User must be registered, signed in, and anywhere on website.

*Summary:* User authenticates with the server and the global chat automatically opens. The chat window displays chat history and updates when new chat messages are received.

*Related use cases:* Join project chat.

*Steps:*

1. User clicks open global chat.
2. Chat is notified that user has joined.
3. System displays global chat window.

*Alternatives:* None.

*Post-conditions:* None.

---

### 1.3.3 Close project chat (jank6275)

---

*Actors:* User

*Goals:* To close the project chat window.

*Pre-conditions:* User must be registered, signed in, and in editor Mode.

*Summary:* User clicks on close project chat and the chat window closes.

*Related use cases:* Close global chat.

*Steps:*

1. User clicks close project chat.
2. Chat is notified that user has left.
3. Client closes project chat window.

*Alternatives:* None.

*Post-conditions:* None.

---

### 1.3.4 Close global chat (jank6275)

---

*Actors:* User

*Goals:* To close the global chat window.

*Pre-conditions:* User must be registered, signed in, and anywhere on website.

*Summary:* User clicks on open global chat and the chat opens, displaying chat history and updating when needed.

*Related use cases:* Close project chat.

*Steps:*

1. User clicks close global chat.
2. Chat is notified that user has left.
3. Client closes global chat window.

*Alternatives:* None.

*Post-conditions:* None.

---

### 1.3.5 Write to project chat (jank6275)

---

*Actors:* User

*Goals:* To send text to project chat.

*Pre-conditions:* User must be registered, signed in, a project opened, with the project chat window open, and the text box selected.

*Summary:* User clicks in the project chat text box and then types a message then either presses enter or clicks the submit button. The text is displayed to all users in the chat, including the user.

*Related use cases:* Write to global chat.

*Steps:*

1. User clicks in the project chat box.
2. User types a message and then presses enter or clicks submit button.
3. Message is relayed to all clients with project chat open.
4. Message is displayed.

*Alternatives:* None.

*Post-conditions:* None.

---



### 1.3.6 Write to global chat (jank6275)

---

*Actors:* User

*Goals:* To send text to global chat.

*Pre-conditions:* User must be registered, signed in, anywhere on website, with the global chat window open, and the text box selected.

*Summary:* User clicks in the global chat text box and then types a message then either presses enter or clicks the submit button. The text is displayed to all users in the chat, including the user.

*Related use cases:* Write to project chat.

*Steps:*

1. User clicks in the global chat box.
2. User types a message and then presses enter or clicks submit button.
3. Message is relayed to all clients with global chat open.
4. Message is displayed.

*Alternatives:* None.

*Post-conditions:* None.

---

### 1.3.7 Modify chat font (jank6275)

---

*Actors:* User

*Goals:* To change a users font style inside the global and project chat.

*Pre-conditions:* User must be registered, signed in, the user settings window opened, and the chat settings tab open.

*Summary:* The user clicks the settings menu and changes their font style for both the project and global chat through a drop down box of available fonts.

*Related use cases:* Modify chat color.

*Steps:*

1. User clicks the settings menu.
2. User clicks chat settings tab.
3. User clicks chat font drop down box.
4. User clicks desired font.
5. User clicks save.
6. The users selection is saved in the database.
7. All further chat messages will use the selected font.

*Alternatives:* None.

*Post-conditions:* None.

---

### 1.3.8 Modify chat color (jank6275)

---

*Actors:* User

*Goals:* To change a users font color inside the global and project chat.

*Pre-conditions:* User must be registered, signed in, the user settings window opened, and the chat settings tab open.

*Summary:* The user clicks the settings menu and changes their font color for both the project and global chat through a drop down box of available colors.

*Related use cases:* Modify chat font.

*Steps:*

1. User clicks the settings menu.
2. User clicks chat settings tab.
3. User clicks chat color drop down box.
4. User clicks desired color.
5. User clicks save.
6. The users selection is saved in the database.
7. All further chat messages from the user will use the selected color.

*Alternatives:* None.

*Post-conditions:* None.

---

### 1.3.9 File Management (snev7821)

#### 1.3.10 Create New Project (snev7821)

**Task Category:** File Management

**Actor:** User

**Summary:** The user performs this task to create a new project.

**Preconditions:**

1. User must be registered.
2. User must be logged in.

**Steps:**

1. User clicks *Projects* in the top menu bar.
2. System opens a drop-down menu.
3. User navigates to *Add -> New Project*.
4. System opens an *Add New Project* dialog window.
5. User provides a title and optionally description.
6. User clicks *Add*.
7. System adds the file(s) to the project.

**Alternatives:**

1. Step 1: The user right clicks in the project panel and the system continues on to step 2 above.
2. Step 5: The user clicks *Cancel* and a new project is not added.

**Postconditions:**

1. A new project is added to the project.
2. The database is updated to reflect the changes.

**Related:** Add Existing File to Project, Add New File to Project

## **Add New File to Project (wern0096) modified for hw4 (snev7821)**

**Task Category:** File Management

**Actor:** User

**Summary:** The user performs this task to add a new file to the project.

### **Preconditions:**

1. User must be registered.
2. User must be logged in.
3. User must have a project open.

### **Steps:**

1. User clicks *File* in the top menu bar.
2. System opens a drop-down menu.
3. User navigates to *Add -> New File*.
4. System opens an *Add New File* dialog window.
5. User selects the file type and names the file.
6. User clicks *Add*.
7. System adds the file to the project.

### **Alternatives:**

1. Step 1: The user right clicks in the project panel and the system continues on to step 2 above.
2. Step 5: The user clicks *Cancel* and a new file is not added to the project.

### **Postconditions:**

1. A new file is added to the project.
2. The database is updated to reflect the changes.

**Related:** Add Existing File to Project

## **Add Existing File to Project (wern0096) modified for hw4 (snev7821)**

**Task Category:** File Management

**Actor:** User

**Summary:** The user performs this task to add an existing file to the project.

### **Preconditions:**

1. User must be registered.
2. User must be logged in.
3. User must have a project open.

### **Steps:**

1. User clicks *File* in the top menu bar.
2. System opens a drop-down menu.
3. User navigates to *Add -> Existing File*.
4. System opens an *Add Existing File* dialog window.
5. User selects *PC* or *SQuire* or *Github*.
6. System updates the dialog to reflect the selected source.
7. User navigates to the file's location and selects it.
8. User clicks *Add*.
9. System adds the file to the project.

### **Alternatives:**

1. Step 1: The user right clicks in the project panel and the system continues on to step 2 above.
2. Step 5-7: The user clicks *Cancel* and a new file is not added to the project.

### **Postconditions:**

1. An existing file is added to the project.
2. The database is updated to reflect the changes.

**Related:** Add New File to Project

## **Delete File (wern0096) modified for hw4 (snev7821)**

**Task Category:** File Management

**Actor:** User

**Summary:** The user performs this task to delete a file from the project.

### **Preconditions:**

1. User must be registered.
2. User must be logged in.
3. User must have a project open.
4. User must be administrator of project.
5. Current project must have at least one file.

### **Steps:**

1. User right clicks a file in the project pane.
2. System opens a drop-down menu.
3. User navigates to *Delete*.
4. System opens an *Delete File* dialog window, asking if the user is sure.
5. User selects *Yes*.
6. System deletes the file from the project.

### **Alternatives:**

1. Step 5: The user clicks *Cancel* instead and the file is not deleted from the project.
2. The user selects multiple files before step 1.

### **Postconditions:**

1. The file is deleted from the project.
2. The database is updated to reflect the changes.

**Related:** Delete Project

**Delete Project (snev7821)**

**Task Category:** File Management

**Actor:** User

**Summary:** The user performs this task to delete a project.

**Preconditions:**

1. User must be registered.
2. User must be logged in.
3. User must have a project open.
4. User must be sole administrator of project.

**Steps:**

1. User clicks *Projects* tab.
2. System opens a drop-down menu.
3. User navigates to *Delete*.
4. System opens an *Delete Project* dialog window, asking if the user is sure.
5. User selects *Yes*.
6. System deletes the project and related files from system.

**Alternatives:**

1. Step 5: The user clicks *Cancel* instead and the project is not deleted.

**Postconditions:**

1. The project (and related files) are deleted.
2. The database is updated to reflect the changes.

**Related:** Delete File



**Export Project (wern0096) modified for hw4 (snev7821)**

**Task Category:** File Management

**Actor:** User

**Summary:** The user performs this task to download a project as a zip file.

**Preconditions:**

1. User must be registered.
2. User must be logged in.
3. User must have a project open.
4. User must have download permissions.

**Steps:**

1. User clicks *File* in the top menu bar.
2. System opens a drop-down menu.
3. User navigates to *Export -> Project*.
4. System opens an *Export* dialog window.
5. User navigates to the export location.
6. User clicks *Export*.
7. System zips the file and downloads it to the specified location.

**Alternatives:**

1. Step 1: The user right clicks in the project panel and the system continues on to step 2 above.
2. Step 5: The user clicks *Cancel* and the project is not exported.

**Related:** Export Files

**Export Files (wern0096) modified for hw4 (snev7821)**

**Task Category:** File Management

**Actor:** User

**Summary:** The user performs this task to download a number of files from a project.

**Preconditions:**

1. User must be registered.
2. User must be logged in.
3. User must have a project open.
4. Must have at least one file in the project.
5. User must have download permissions.

**Steps:**

1. User clicks *File* in the top menu bar.
2. System opens a drop-down menu.
3. User navigates to *Export -> Files*.
4. System opens an *Export* dialog window showing the project files on the left panel and the export location in the right panel.
5. User selects a number of files on the left pane.
6. User navigates to the export location in the right pane.
7. User clicks *Export*.
8. System downloads the selected files to the specified location.

**Alternatives:**

1. Step 1: The user right clicks in the project panel and the system continues on to step 2 above.
2. Step 5: User selects a folder and all files under that folder are selected.
3. Step 5-6: The user clicks *Cancel* and the project is not exported.

**Related:** Export Project

**Open File in New Tab (wern0096) modified for hw4 (snev7821)**

**Category:** File Management

**Actor:** User

**Summary:** Allows users to open a file.

**Purpose:** Opening files is essential in being able to work on a project.

**Preconditions:**

1. User is registered.
2. User is logged in.
3. User has a project open.
4. Current project contains at least one file.
5. User has read permission.

**Steps:**

1. User double clicks a file.
2. The editor opens its contents in a new tab and focuses on it.

**Alternatives:** Step 1: Instead of double clicking a file, the user right clicks it and navigates to *Open*.

## 1.4 File Editing (snev7821)

### 1.4.1 View Line Numbers (wern0096) updated by (snev7821)

**Category:** File Editing

**Actor:** User

**Summary:** Allows the user to hide line numbers to the left of the document.

**Purpose:** In case user wants to hide line numbers so they have more space for text.

**Preconditions:**

1. Must be registered.
2. Must be logged in.
3. User has view permission.
4. A file is open.
5. Line numbers are on

**Steps:**

1. User selects the *View* menu option.
2. System displays a drop-down with various options.
3. User selects the *Hide Line Numbers* option.
4. System hides line numbers to the left of the document.

**Related:**

1. View References
2. View Dates
3. View Authors

### 1.4.2 View References (wern0096) updated for hw4 by (snev7821)

**Category:** File Editing

**Actor:** User

**Summary:** Allows the user to view the number of references to a given function.

**Purpose:** It is useful to know the number of references to a given function for optimization and debugging purposes.

**Preconditions:**

1. Must be registered.
2. Must be logged in.
3. User has view permission.
4. A **code** file is open.

**Steps:**

1. User selects the *View* menu option.
2. System displays a drop-down with various options.
3. User selects the *View References* option.
4. System displays the number of references above each method declaration.

**Related:**

1. Hide Line Numbers
2. View Dates
3. View Authors

### 1.4.3 View Dates (wern0096) updated by (snev7821)

**Category:** File Editing

**Actor:** User

**Summary:** Allows the user to view the last date that each block of a document was edited. Blocks are defined as any number of lines that was written by a single user. Minimum block size is one line.

**Purpose:** This provides a useful metric for how up-to-date parts of the document are.

**Preconditions:**

1. Must be registered.
2. Must be logged in.
3. User has view permission.
4. A file is open.

**Steps:**

1. User selects the *View* menu option.
2. System displays a drop-down with various options.
3. User selects the *View Dates* option.
4. System displays the last date that each block of a document was edited.

**Related:**

1. Hide Line Numbers
2. View References
3. View Authors

#### 1.4.4 View Authors (wern0096) updated by (snev7821)

**Category:** File Editing

**Actor:** User

**Summary:** Allows the user to view the last author that edited each block of a document. Blocks are defined as any number of lines that was written by a single user. Minimum block size is one line.

**Purpose:** This is an accountability tool allowing other users to identify who is responsible for a change to a document.

**Preconditions:**

1. Must be registered.
2. Must be logged in.
3. User has read permission.
4. A file is open.

**Steps:**

1. User selects the *View* menu option.
2. System displays a drop-down with various options.
3. User selects the *View Authors* option.
4. System displays the name of the last editor of each line of the document.

**Related:**

1. Hide Line Numbers
2. View References
3. View Dates

### 1.4.5 Format Document (wern0096) updated by (snev7821)

**Category:** File Editing

**Actor:** User

**Summary:** Allows the user to format the document to a specified format

**Purpose:** An easy tool for making sweeping changes to a large part of a document.

**Preconditions:**

1. Must be registered.
2. Must be logged in.
3. User has read/write permission.
4. A file is open.
5. The document has formatting options set.

**Steps:**

1. User selects the *Edit* menu option.
2. System displays a drop-down with various options.
3. User selects the *Format Document* option.
4. System formats the current document to the formatting settings currently set.

**Alternatives:**

1. If no formatting settings are currently set, display a dialog box after step 3 and give the option for the user to do so now.

**Related:**

1. Find/Replace
2. Comment Section



### 1.4.6 Find/Replace (wern0096) updated by (snev7821)

**Category:** File Editing

**Actor:** User

**Summary:** Allows the user to find and/or replace phrases.

**Purpose:** This is a powerful tool that allows a user to make safer, quicker, and more efficient changes to a document.

**Preconditions:**

1. Must be registered.
2. Must be logged in.
3. User has read/write permission.
4. A file is open.

**Steps:**

1. User selects the *Edit* menu option.
2. System displays a drop-down with various options.
3. User selects the *Find/Replace* option.
4. System displays a small form in an unobtrusive location.
5. User enter the phrase to find and selects find.
6. System highlights and focuses on the first occurrence of the phrase and all highlights all other occurrences.

**Alternatives:**

1. User selects option to replace in step 5 and enters a phrase with which to replace the found occurrences of the searched phrase. The system replaces each occurrence.

**Related:**

1. Format Document
2. Find/Replace

### 1.4.7 Comment Section (wern0096) updated by (snev7821)

**Category:** File Editing

**Actor:** User

**Summary:** Allows the user to comment out a part of a document.

**Purpose:** A useful and quick way to disable a large part of a document.

**Preconditions:**

1. Must be registered.
2. Must be logged in.
3. A file is open.
4. User has read/write permission.
5. Current open document supports commenting.

**Steps:**

1. User selects the *Edit* menu option.
2. System displays a drop-down with various options.
3. User selects the *Comment Section* option.
4. System comments the selected area.

**Alternatives:**

1. If document does not support commenting, display a dialog box telling the user.

**Related:**

1. Format Document
2. Find/Replace

#### 1.4.8 Display Typing User (wern0096) updated by (snev7821)

**Category:** File Editing

**Actors:**

1. User
2. Other Users

**Summary:** As the user types, the system displays their name, their typing, and their caret, in a different color, to other users.

**Purpose:** Differentiate who is typing what.

**Preconditions:**

1. Must be registered.
2. Must be logged in.
3. User has read/write permission.
4. A file is open.
5. Other users have the same document open.

**Steps:**

1. User begins typing.
2. System displays the user's typing, the user's name, and the user's caret, in a different color, to Other Users.
3. Other Users see User typing, his username, and his caret, in a different color.

#### 1.4.9 Display Syntax Highlighting (wern0096) updated by (snev7821)

**Category:** File Editing

**Actor:** User

**Summary:** As the user types code, the editor will change font color for different code structures and keywords.

**Purpose:** Aids the user is writing code and identifying key code parts.

**Preconditions:**

1. Must be registered.
2. Must be logged in.
3. User has read/write permission.
4. A supported code file is open.

**Steps:**

1. User begins typing.
2. System automatically colors special code structures and keywords.

**Related:** Display Syntax Errors

## **1.5 User Preferences (snev7821)**

### **1.5.1 View User Preferences (snev7821)**

**Name:** View User Preferences

**Category:** User Preferences - Editor

**Actor:** User

**Summary:** User views their preferences and from here can change them

**Purpose:** Allows user to view their preferences and change them

**Preconditions:**

1. Must be registered
2. Must be logged in
3. User is on user homepage

**Steps:**

1. User clicks "Manage Editor Preferences"
2. System presents user with preferences page

**Related:** Modify chat font, Modify chat color, Edit user color

### **1.5.2 Modify Chat Font (snev7821)**

**Name:** Modify Chat Font

**Category:** User Preferences - Editor

**Actor:** User

**Summary:** User changes the chat font

**Purpose:** Allows user to change what chat font they see for themselves and others

**Preconditions:**

1. Must be registered
2. Must be logged in
3. User is on user editor preferences page

**Steps:**

1. User clicks "Modify Chat Fonts" button
2. System brings up list of fonts, for the user and others
3. User selects a font for self
4. User sets a font for others
5. System saves user choices after each user action

**Related:** Modify chat color, Edit user editor theme

### 1.5.3 Modify Chat Color (snev7821)

**Name:** Modify Chat Color

**Category:** User Preferences - Editor

**Actor:** User

**Summary:** User changes the chat color

**Purpose:** Allows user to change what chat color they see for themselves and others

**Preconditions:**

1. Must be registered
2. Must be logged in
3. User is on user editor preferences page

**Steps:**

1. User clicks "Modify Chat Colors" button
2. System brings up list of colors, for the user and others
3. User selects a color for self
4. User sets a color for others
5. System saves user choices after each user action

**Related:** Modify chat font, Edit user editor theme

#### **1.5.4 Edit User Editor Theme (snev7821)**

**Name:** Edit User Editor Theme

**Category:** User Preferences - Editor

**Actor:** User

**Summary:** User changes the Editor theme

**Purpose:** Allows user to change theme of the collaborative editor

**Preconditions:**

1. Must be registered
2. Must be logged in
3. User is on user editor preferences page

**Steps:**

1. User clicks "Modify Editor Theme" button
2. System brings up list of themes for editor
3. User selects a theme
4. System saves user selection

**Related:** Modify chat font, Edit user color



### **1.5.5 Turn Off Global Chat (snev7821)**

**Name:** Turn Off Global Chat

**Category:** User Preferences - Editor

**Actor:** User

**Summary:** User turns off global chat

**Purpose:** Allows user to choose whether or not to engage in global chat

**Preconditions:**

1. Must be registered
2. Must be logged in
3. User is on user editor preferences page

**Steps:**

1. User checks "turn global chat off" box
2. System brings up warning, explaining what this does
3. User selects accept
4. System saves user selection
5. System disconnects user from global chat

**Related:** Global chat

### 1.5.6 Psuedo-offline Mode (snev7821)

**Name:** Psuedo-offline Mode

**Category:** User Preferences - Editor

**Actor:** User

**Summary:** User changes to offline mode

**Purpose:** Allows user to turn off online features, including chat, public profiles, etc. Site then serves as basic editing environment

**Preconditions:**

1. Must be registered
2. Must be logged in
3. User is on user editor preferences page

**Steps:**

1. User clicks "offline mode" button
2. System brings up warning
3. User selects accept
4. System saves user selection
5. System closes chat
6. System loads offline user page
7. Upon disconnect with site, online mode restarts upon next connection

**Related:** None

## 1.6 Project Management

### 1.6.1 Create Project (Created by bolt1003)

---

*Actors:* Users of sQuire.

*Goals:* Create a Project.

*Pre-conditions:* The user is logged in and at the dashboard.

*Summary:* The user creates a project.

*Related use cases:* None.

*Steps:*

1. User selects the "+" icon and a wizard appears.
2. A name is chosen for the project.
3. Language is selected from a drop down menu.
4. User clicks finish.

*Alternatives:* Create project from the editor.

*Post-conditions:* The user assigns permissions to access the project.

---

### 1.6.2 Open a project (Created by bolt1003)

---

*Actors:* Users of sQuire.

*Goals:* Choose the desired project and open it.

*Pre-conditions:* One or more projects are available, the user is logged in and at the dashboard.

*Summary:* User looks through a list of projects and selects the desired project.

*Related use cases:* None.

*Steps:*

1. User clicks on projects in the menu bar.
2. A list of projects appears and the user clicks on the desired project.

*Alternatives:* Open a project from recent projects.

*Post-conditions:* User closes sQuire.

---

### 1.6.3 Join Project (Created by sass8427, Revised by bolt1003)

---

*Actors:* Users of sQuire.

*Goals:* Join an existing project.

*Pre-conditions:* Must be registered, logged in and have permission to join a project.

*Summary:* The user logs in, chooses a project, and joins the project.

*Related use cases:* Invite user to project, Accept user invite.

*Steps:*

1. The user selects a project.
2. The user chooses the "Join".
3. The project is added to the users projects bar.
4. The user selects the project and selects "open".

*Alternatives:* User may decline an invitation to join a project.

*Post-conditions:* None

---

#### 1.6.4 Leave project (Created by sass8427, Revised by bolt1003)

---

*Actors:* User

*Goals:* Remove member status from project.

*Pre-conditions:* Logged in, member of the respective project, not project owner.

*Summary:* A member of a project can unjoin that project at any time as long as they are not the project owner. To prevent mistakenly unjoining a project, the user is asked to confirm their decision.

*Related use cases:*

*Steps:*

1. User selects a project.
2. User clicks "Unjoin".
3. User is prompted to confirm their decision
4. User clicks "Confirm".
5. User is removed from project member list.

*Alternatives:* User clicks "Cancel" at step 4, in which case the task is ends at that point.

*Post-conditions:* None.

---

### 1.6.5 Delete Project(Created by mora5651, Revised by bolt1003)

---

*Actors:* Users of sQuire.

*Goals:* Delete an existing project.

*Pre-conditions:* The user has the appropriate permissions to delete project.

*Summary:* A user deletes a project from the project workspace.

*Related use cases:* Create a project.

*Steps:*

1. The user selects a project.
2. The user clicks on the "Delete project" button.
3. A dialog is displayed.
4. User select "delete" to delete the project.

*Alternatives:* User may choose not to delete the project in the confirmation display.

*Post-conditions:* None.

---

### 1.6.6 Export Project(Created by knic1468, Revised by bolt1003)

---

*Actors:* User of sQuire.

*Goals:* Export a workspace to a local file.

*Pre-conditions:* The user needs permission to export the project.

*Summary:* User saves a file containing the project settings and files to a local machine.

*Related use cases:* Importing a project, Creating a new project.

*Steps:*

1. The user clicks on the "Export File" button.
2. System prompts the user to select a location and filename.
3. User selects a file location.
4. The user enters a file name.
5. The user selects "export".

*Alternatives:* The user cancels the export, The system prompts that a file already exists with the same name.

*Post-conditions:* None.

---



### 1.6.7 Accept Invite to Project (Created by carl7595, Revised by bolt1003)

---

*Actors:* User who received the invite.

*Goals:* Gain access to a Project.

*Pre-conditions:* User has a valid email address.

*Summary:* Access is granted to a project using an invitation email.

*Related use cases:* Create an account.

*Steps:*

1. Invitee clicks on the link received by email.
2. The link opens in a browser.
3. Dialog appear welcoming them to the project.
4. The project is added to their Projects list.

*Alternatives:* The user ignores the invite.

*Post-conditions:* Email link is deactivated.

---

### 1.6.8 Remove User from Project (Created by carl7595, Revised by bolt1003)

---

*Actors:* User of sQuire

*Goals:* Revoke access to the Project for a single or multiple users.

*Pre-conditions:* The user has permission to edit the Project access list.

*Summary:* One or more user accounts are removed from the access list for a project.

*Related use cases:* Add users to a project.

*Steps:*

1. The user selects the access list for the project.
2. The user selects an account.
3. The user selects "Remove from Project".
4. The user is prompted for confirmation
5. The user selects 'Yes'.

*Alternatives:* The user selects 'No' and the access list is not modified.

*Post-conditions:*

- The user that was removed is notified of the change.
- The user is prevented from accessing files.

### 1.6.9 Edit Project Permissions (Created by benz5834, Revised by bolt1003)

---

*Actors:* User of sQuire

*Goals:* Edit the permissions for a project

*Pre-conditions:* The user is logged in.

*Summary:* User opens up the settings menu and navigates to permissions, adds (or removes) users individual access rights to the project.

*Related use cases:* Add user to project, Remove user from project.

*Steps:*

1. The user selects a project.
2. The user selects settings.
3. The user selects permissions.
4. The user selects user from list of users.
5. The user adds read or write permissions to user.
6. The user selects save to save changes.
7. The user exits settings.

*Alternatives:* User can remove read or write permission instead in step 6. User can discard changes instead in step 7.

*Post-conditions:* None.

---

## 1.7 Settings - Preferences and Profile (brec9824)

### 1.7.1 View A User's Profile (brec9824)

---

*Actors:* User of sQuire.

*Goals:* User views a profile page.

*Pre-conditions:*

1. The user is logged in.

*Summary:* User clicks on their username or another user's name and a goes to a new page with the selected user's profile page.

*Related use cases:* Modify Profile Info.

*Steps:*

1. The user clicks on their username or another user's name.
2. The user's system sends a request to the main sQuire system for the selected users profile information.
3. sQuire system approves the request and sends the selected user's full profile info.
4. The user's system loads a new page displaying the selected user's full profile info.

*Alternatives:* In step 3 sQuire approves the request but because of the selected user's privacy settings only partial profile info is sent to the user.

*Post-conditions:* None.

---

### 1.7.2 Modify Profile Info (brec9824)

---

*Actors:* User of sQuire.

*Goals:* User updates their profile info including project preferences.

*Pre-conditions:*

1. The user is logged in.
2. The user is at their profile page.

*Summary:* User clicks on the edit button, modifies their info, clicks save and their info gets saved.

*Related use cases:* Modify Profile Privacy.

*Steps:*

1. The user clicks the edit button.
2. The user's system sends a request to the sQuire system.
3. sQuire system receives the request and verifies the user's credentials.
4. The user's system loads a new page displaying the user's profile but with editable text boxes.
5. The user edits their desired info.
6. The user clicks the save button and the user's system sends the updated info to the sQuire system.
7. sQuire system receives the new data, verifies the data meets pre-defined requirements and approves the update.
8. User is returned to their profile page as before with their updated info.

*Alternatives:*

1. In step 3 the sQuire system denies the request because the user was idle too long and is not logged in anymore.
2. In step 7 the sQuire system denies the user's request to update their profile: 1. the user's email was invalid 2. the user's password didn't meet the security requirements 3. the user used ineligible words or phrases. User<sup>53</sup> is notified of the denial and is returned to step 5.

### 1.7.3 Modify Profile Privacy (brec9824)

---

*Actors:* User of sQuire.

*Goals:* User updates their profile info.

*Pre-conditions:*

1. The user is logged in.
2. The user is at their profile page.

*Summary:* User clicks on the privacy level checkbox, clicks save and their new privacy level is saved.

*Related use cases:* Modify Profile Info.

*Steps:*

1. The user clicks the appropriate privacy checkbox next to the data they want to change the privacy of.
2. The user clicks the save button.
3. sQuire system receives the request and verifies the user's credentials.
4. sQuire system receives the updated privacy settings and approves the update.
5. User is returned to their profile page as before with their updated info.

*Alternatives:*

1. In step 3 the sQuire system denies the request because the user was idle too long and is not logged in anymore.

*Post-conditions:* None.

---

#### 1.7.4 Delete Account (brec9824)

---

*Actors:* User of sQuire.

*Goals:* Delete the user's account.

*Pre-conditions:*

1. The user is logged in.
2. The user is at their profile page.

*Summary:* User clicks on the delete account button, then confirms their choice and their account is deleted from sQuire servers after a set amount of time.

*Related use cases:* Modify Profile Info.

*Steps:*

1. The user clicks the delete account button.
2. System covers the room window with a new window that is dark and nearly transparent.(Gives the appearance that the page is dimmed)
3. System opens a pop-up window that contains a confirm button, a cancel button, and text that asks the user if they are sure and notifies them that this action is permanent.
4. The user clicks the submit button.
5. System closes the pop-up windows and the dim window in the background.
6. System kicks the user from their account and adds the account to a list for future deletions.
7. User is returned to sQuire's home page.

*Alternatives:*

1. If the user in step 4 clicks cancel or clicks out of the pop-up window and onto the dim window in the background. The dim window created in step 2 and the pop-up window in step 3 closes and action is taken.

*Post-conditions:* None.

### 1.7.5 Quick Jump To Projects (brec9824)

---

*Actors:* User of sQuire.

*Goals:* Go to a projects page that is listed in a user's profile.

*Pre-conditions:*

1. The user is logged in.
2. The user is at a user's profile page.

*Summary:* User clicks on the appropriate project name and then is redirected to the projects home page.

*Related use cases:* None.

*Steps:*

1. The user searches through the projects listed in the user's profile page and clicks the project name they would like to go to.
2. System receives the request and searches for the specified project in the project data base.
3. System finds the project and sends the redirect info.
4. The user is then redirected to the given projects home page screen.

*Alternatives:*

1. None.

*Post-conditions:* None.

---



## 1.8 Project User Management (boss2849)

### 1.8.1 Modify read/write access (boss2849)

---

*Actors:* User

*Goals:* Modify a user's permissions.

*Pre-conditions:* User is signed in and holds Admin rights for the currently selected Project

*Summary:* User modifies another User's read/write permissions to portions of the project.

*Related use cases:* None.

*Steps:*

1. User clicks Permissions Management
2. System displays permissions management window
3. User selects a file, multiple files, directory or entirety of project and grants/revokes read/write access
4. System modifies the target User's permissions and notifies them.

*Alternatives:* 3. User selects cancel, System discards changes.

*Post-conditions:* None.

---

## 1.8.2 Remove User (boss2849)

---

*Actors:* User

*Goals:* Remove a user from project.

*Pre-conditions:* User is signed in, in project with Admin rights, and is on User Management page

*Summary:* User removes a selected user from the Project

*Related use cases:* Invite User, Modify Read/Write Access

*Steps:*

1. User clicks Remove User button.
2. System displays list of active users for project.
3. User selects one or more other users from the list and presses Remove.
4. System prompts User for verification.
5. User presses Confirm.
6. System removes the selected users from the project.
7. System revokes read and write access from the selected users.
8. System notifies selected users that they have been removed from the project.

*Alternatives:* User presses Cancel in steps 3 or 5, System returns user to User Management page

*Post-conditions:* None.

---

### 1.8.3 Invite User to Project (boss2849)

---

*Actors:* User

*Goals:* Invite user(s) to project

*Pre-conditions:* User is signed in, in project with Admin rights, and is on User Management page

*Summary:* User invites user(s) to the current project.

*Related use cases:* Remove User, Join Project

*Steps:*

1. User clicks Invite Users button
2. System prompts user to enter username(s)/email(s)
3. User enters username(s)/email(s) of the user(s) to invite and presses Ok.
4. System looks up the specified user(s) and notifies them of invitation to the Project

*Alternatives:*

1. User presses cancel in step 3, System returns User to User Management page
2. In step 4, username(s)/email(s) don't match any users, System notifies User of failed invitations.

*Post-conditions:* None.

---

#### 1.8.4 Promote User to Admin (boss2849)

---

*Actors:* User

*Goals:* Promote a specified User to Admin

*Pre-conditions:* User is signed in, in project with Admin rights, and is on User Management page

*Summary:* User selects another User to be given Admin rights for the project.

*Related use cases:* Demote Admin

*Steps:*

1. User selects Promote to Admin.
2. System displays a list of non-Admin active users.
3. User selects user(s) and presses Submit.
4. System prompts user for confirmation.
5. User selects Confirm.
6. System grants Admin permissions to the selected user(s).

*Alternatives:* User presses cancel in steps 3 or 5, no action taken.

*Post-conditions:* None.

---

### 1.8.5 Demote Admin (boss2849)

---

*Actors:* User

*Goals:* Demote Admin to user

*Pre-conditions:* User is signed in, in project with Admin rights, and is on User Management page

*Summary:* User demotes selected Admins to normal Users for the project.

*Related use cases:* Promote User to Admin

*Steps:*

1. User selects Demote Admin
2. System displays list of Admins
3. User selects Admin(s) to demote and presses Submit.
4. System prompts User for confirmation.
5. User presses Confirm.
6. System revokes Admin rights from selected User(s)

*Alternatives:*

1. User presses cancel in steps 3 or 5, no action taken
2. User attempts to demote Admin that is the Owner of the project, System rejects request and notifies User.

*Post-conditions:* None.

---

### 1.8.6 Block User (boss2849)

---

*Actors:* User

*Goals:* Block a user from the project

*Pre-conditions:* User is signed in, in project with Admin rights, and is on User Management page

*Summary:* User blocks a user from the project, making them unable to view the project.

*Related use cases:* Demote Admin

*Steps:*

1. User clicks Block User.
2. System displays a list of active users.
3. User selects other user(s) to block and presses Submit.
4. System prompts User for confirmation.
5. User presses Confirm.
6. System blocks selected user(s) from the project, revoking read/write access, and revoking Admin status as necessary.

*Alternatives:* User presses cancel in steps 3 or 5.

*Post-conditions:* None.

---

## 1.9 Compile (boss2849)

---

*Actors:* User

*Goals:* Compile source

*Pre-conditions:* User is logged in and viewing project.

*Summary:* User requests that the code be compiled, the server compiles the code.

*Related use cases:* Run, Compile To Jar

*Steps:*

1. User selects “Compile” from “Build” dropdown menu for the current module.
2. The Server receives the request to compile.
3. The Server caches the current state of the project using the SnapshotManager and compiles it using the active CompilerPlugin.
4. The Server returns the results of compilation to the User.

*Alternatives:* In step 1, user chooses to compile entire project, including all sub modules.

*Post-conditions:* None.

---

## 1.10 Run (boss2849)

---

*Actors:* User.

*Goals:* Run the program.

*Pre-conditions:* User is logged in and viewing a project.

*Summary:* User chooses to run the program and the server compiles it or executes the last compilation result if no changes.

*Related use cases:* Compile

*Steps:*

1. User selects “Run” from the “Build” menu drop down.
2. The Server receives the request to execute.
3. The Server retrieves the most recent compilation from the SnapshotManager.
4. The Server spawns a new window for the client that is the interface to the program.

*Alternatives:* In step 3 the SnapshotManager has either an out of date compilation or no last compilation, the Server invokes the compiler to compile the project.

*Post-conditions:* None.

---



## 1.11 Package to Jar (boss2849)

---

*Actors:* User

*Goals:* Compile and package source to a jar

*Pre-conditions:* User is signed in and viewing a project.

*Summary:* User selects to build the project to a jar, the server outputs a jar on the project path.

*Related use cases:* Compile

*Steps:*

1. The user selects “Compile To Jar” from “Build” dropdown menu.
2. The Server receives the request to build a jar.
3. The Server fetches the most recent compilation from the Snapshot-Manager.
4. The Server packages the result of the last compilation to a jar and outputs it on the project path.
5. The Server notifies the user of success.

*Alternatives:* In step 3 the SnapshotManager has either an out of date compilation or no last compilation, the Server invokes the compiler to compile the project. In step 4 or 5, the compilation process fails and the Server notifies the user with the reason of failure.

*Post-conditions:* None.

---

## 1.12 Enable code freeze (boss2849)

---

*Actors:* User

*Goals:* Impose a code freeze on the project.

*Pre-conditions:* User is signed in, viewing a project, and has admin rights.

*Summary:* User places a code freeze on the project, preventing editing until undone.

*Related use cases:* None.

*Steps:*

1. The User selects “Code Freeze” from the dropdown menu.
2. The Server receives the request for code freeze.
3. The Server restricts all editing of project files.

*Alternatives:* None.

*Post-conditions:* None.

---