sQuire: A Web Based Collaborative Editor

bolt1003, wern0096, alsh5301, sass8427, dani2918, boss2849, snev7821, jank6275

February 5, 2016

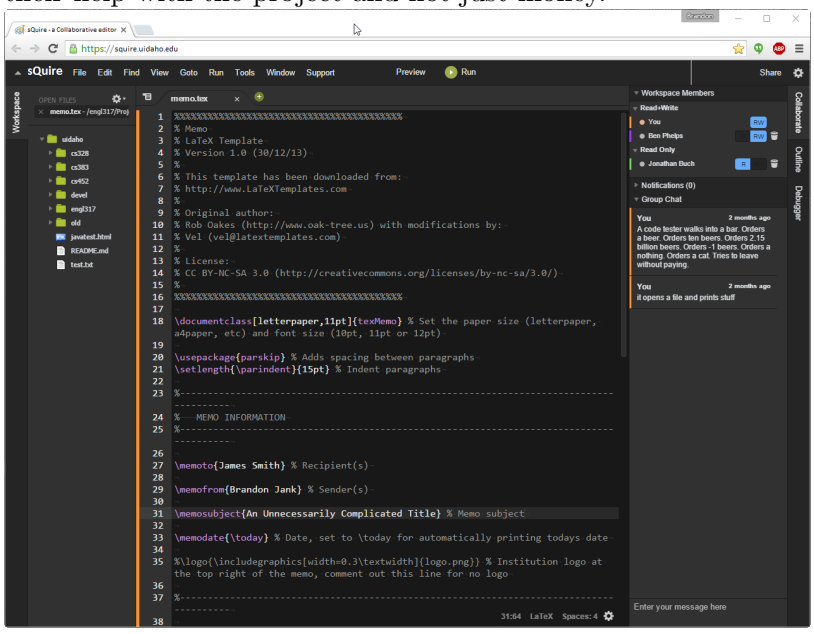
Contents

Chapter 1

Application Domain Specification

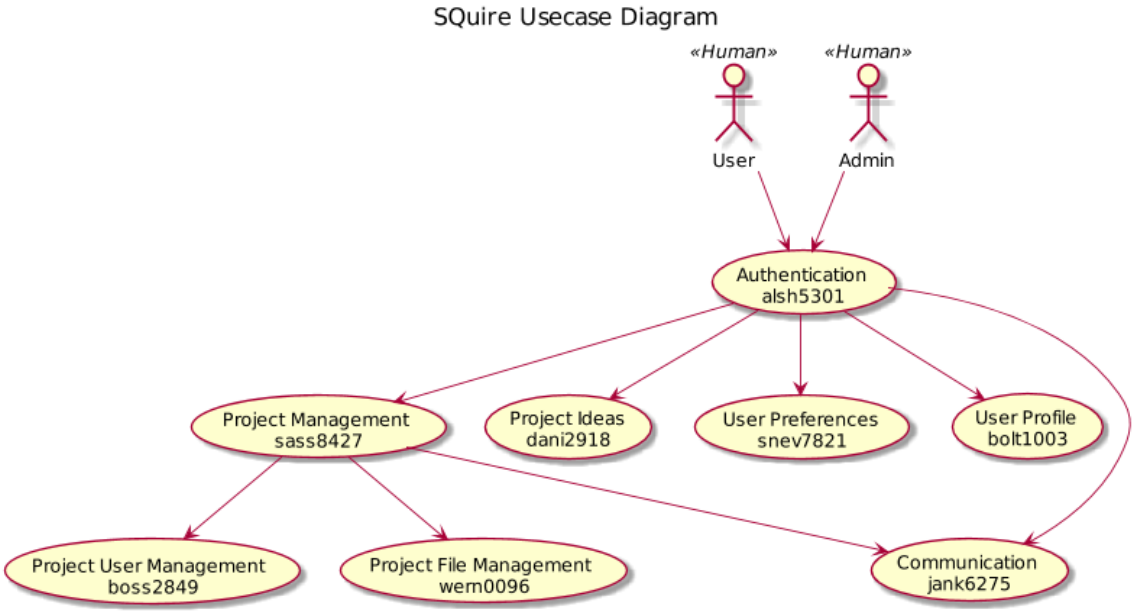
1.1 Program Premise

Figure 1.1: Squire will be a web-based collaborative integrated development environment with a project development center. Squire will allow multiple users to edit files and communicate in real time. First, projects are stubbed out by a user and then other users can join and/or vote to support for their favorite projects. After a certain amount of support, planning, and documentation is reached for a project, the project becomes a fully fleged project and then community development can start. Think kickstarter for code where people pledge their help with the project and not just money.



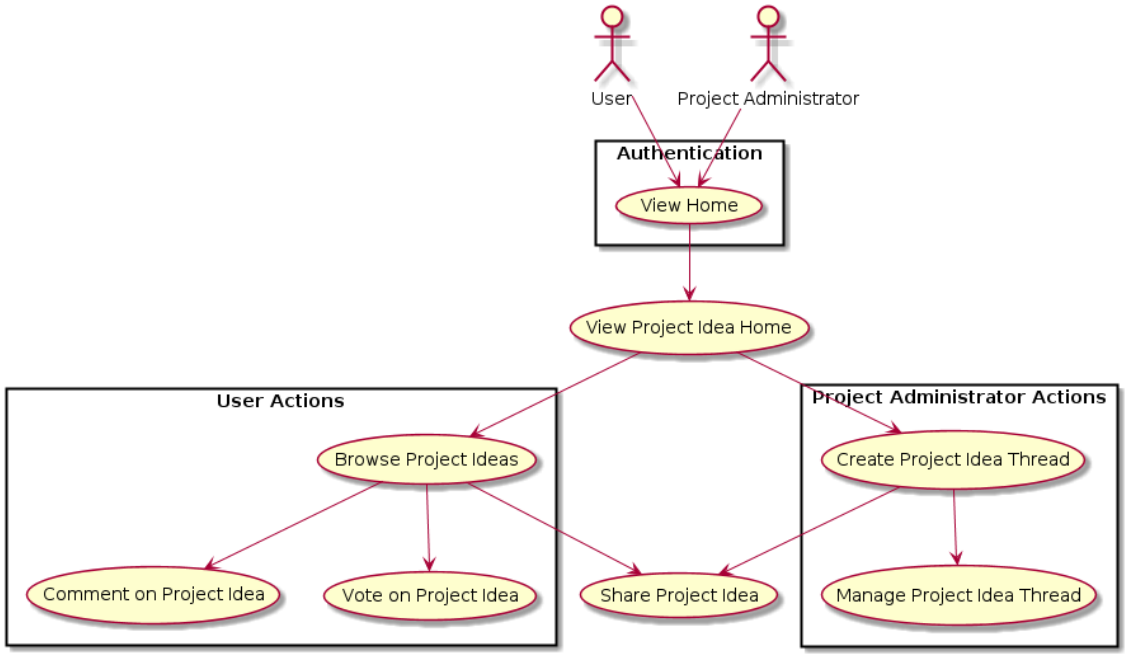
1.2 Use Case Diagrams

1.2.1 Overview (jank6275)

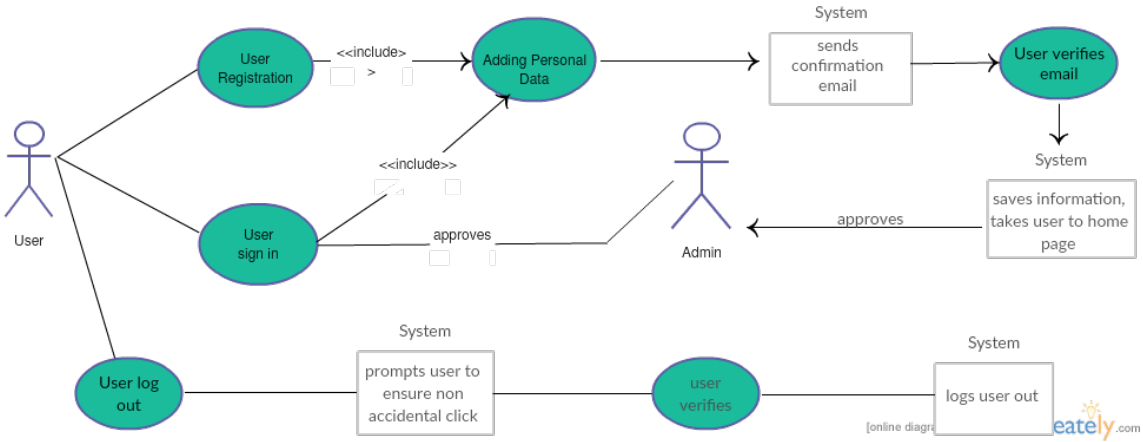


A usecase diagram that shows all of sQuireś features.

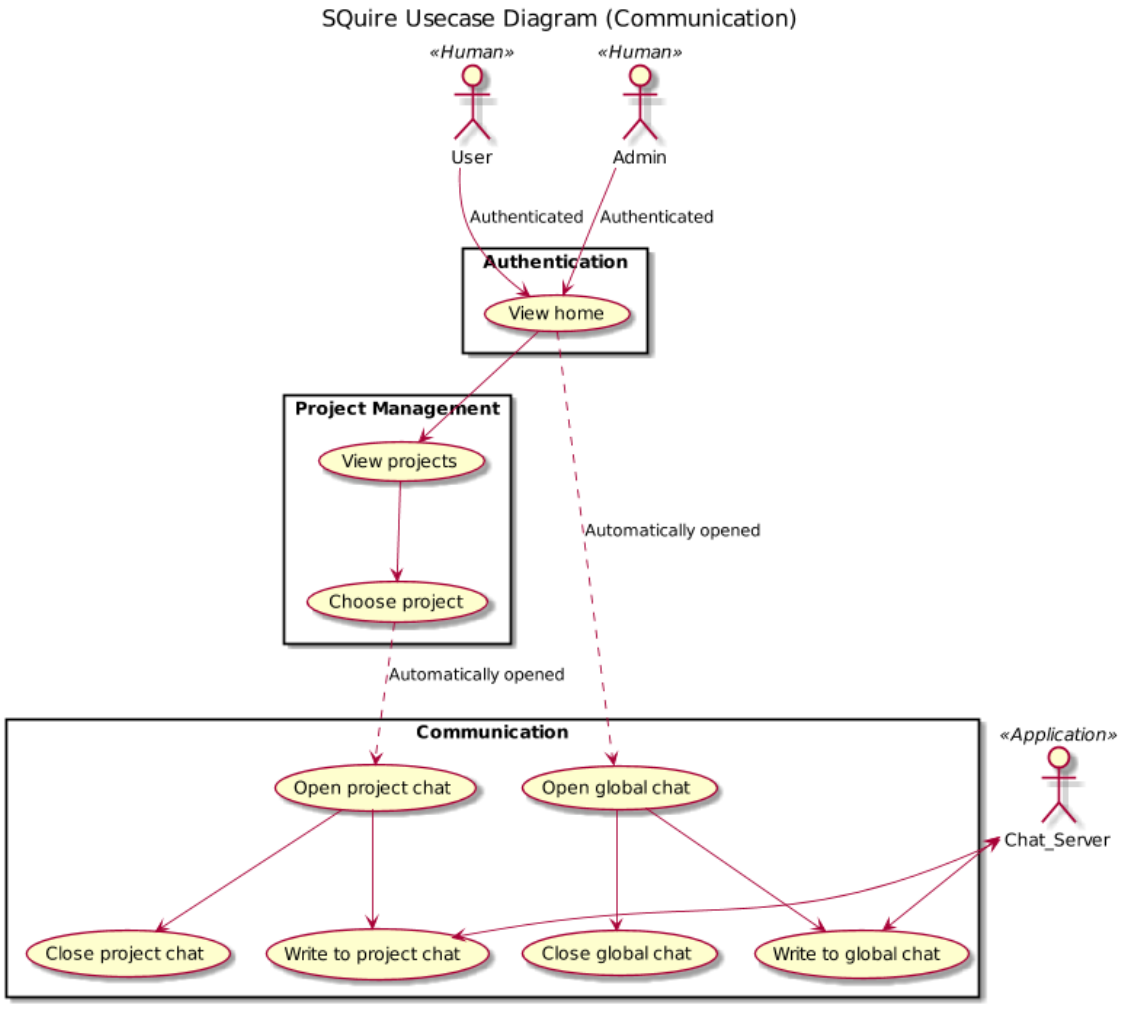
1.2.2 Project Ideas (dani2918)



1.2.3 Authentication (alsh5301)

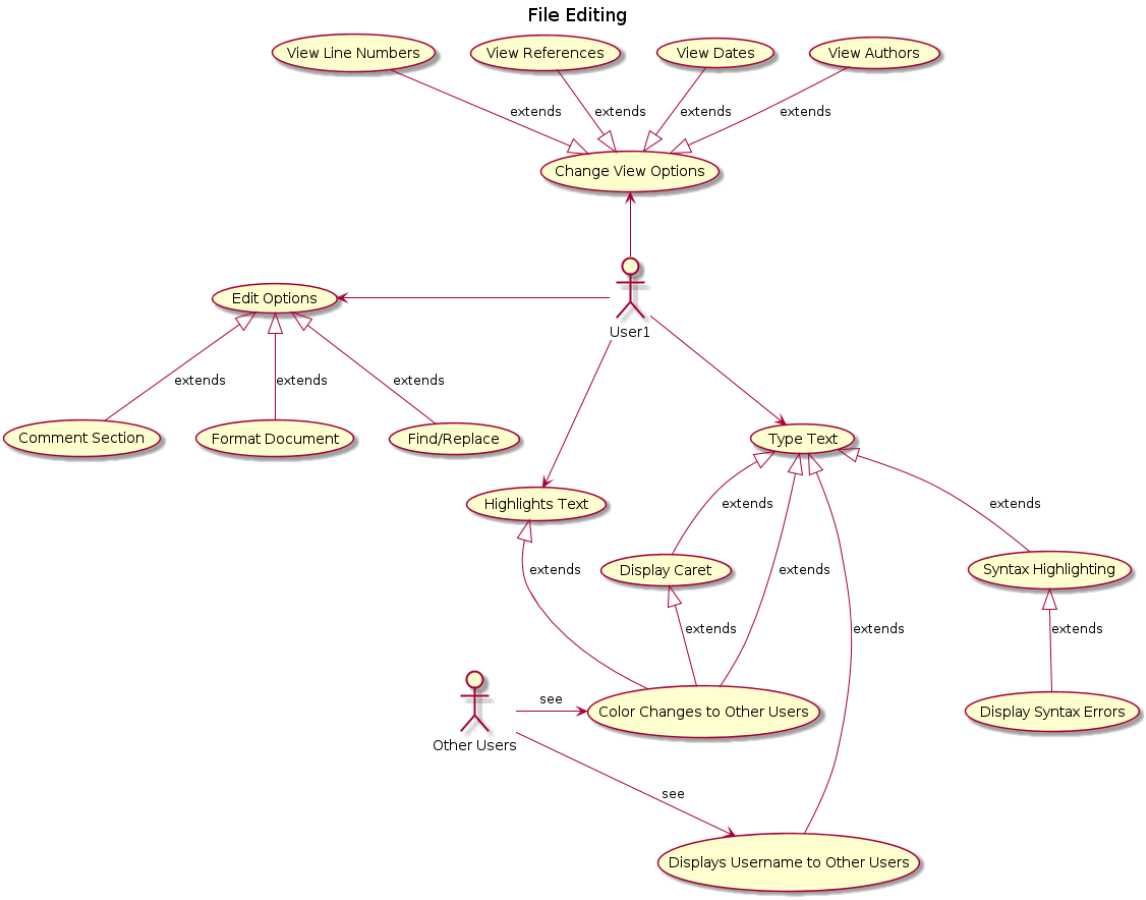


1.2.4 Communication (jank6275)

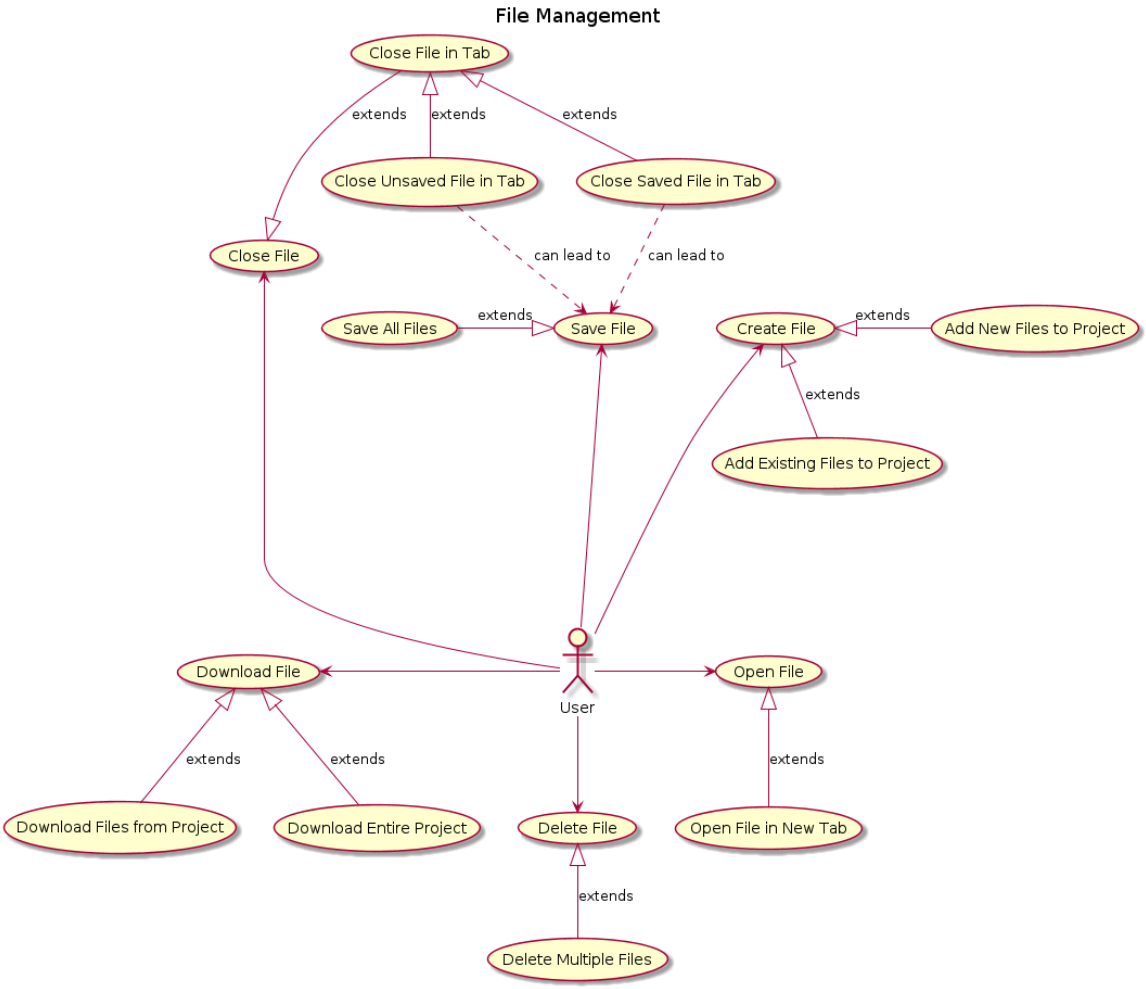


A usecase diagram for sQuireś communication features.

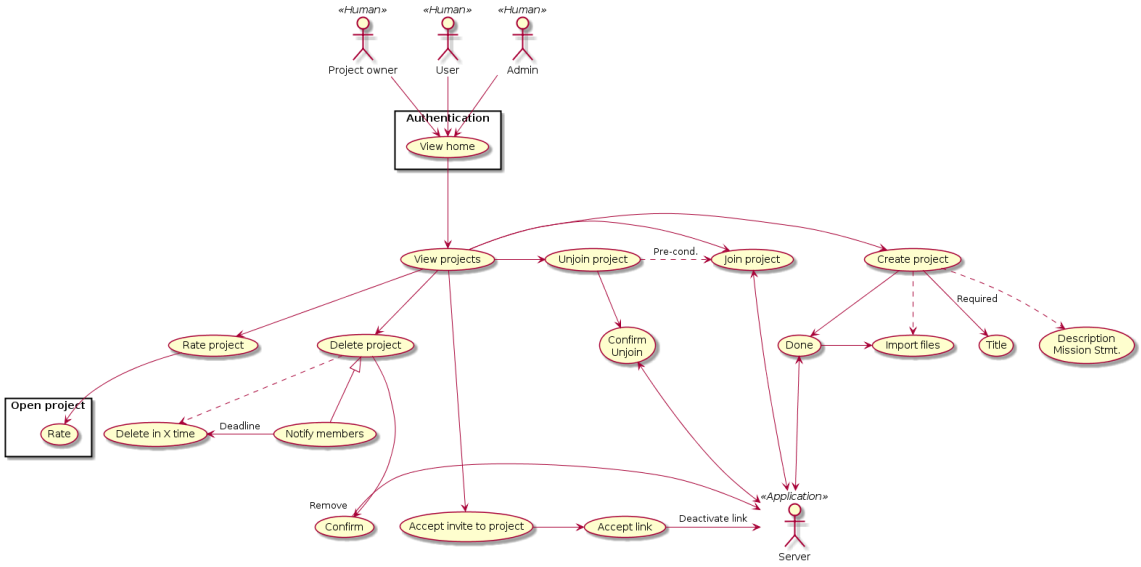
1.2.5 File Editing (wern0096)



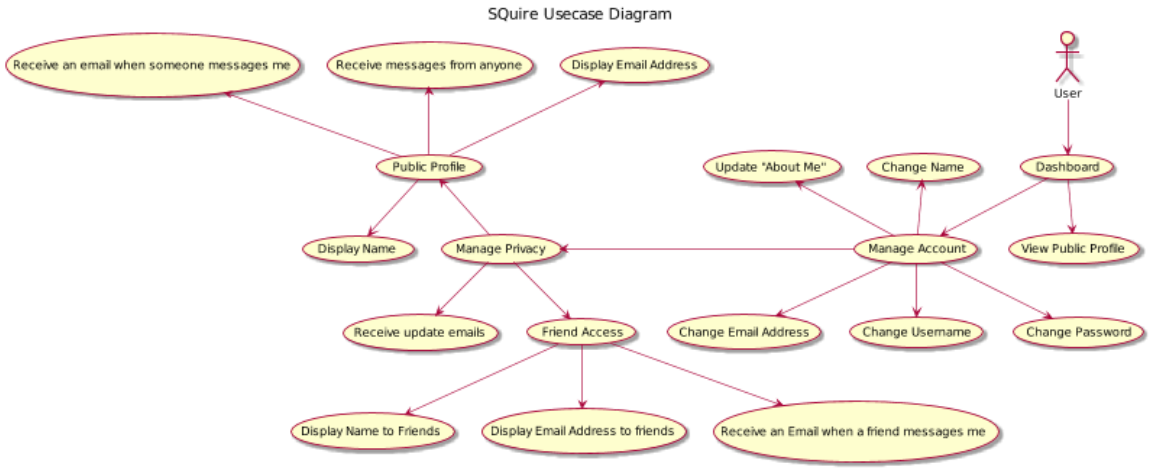
1.2.6 File Management (wern0096)



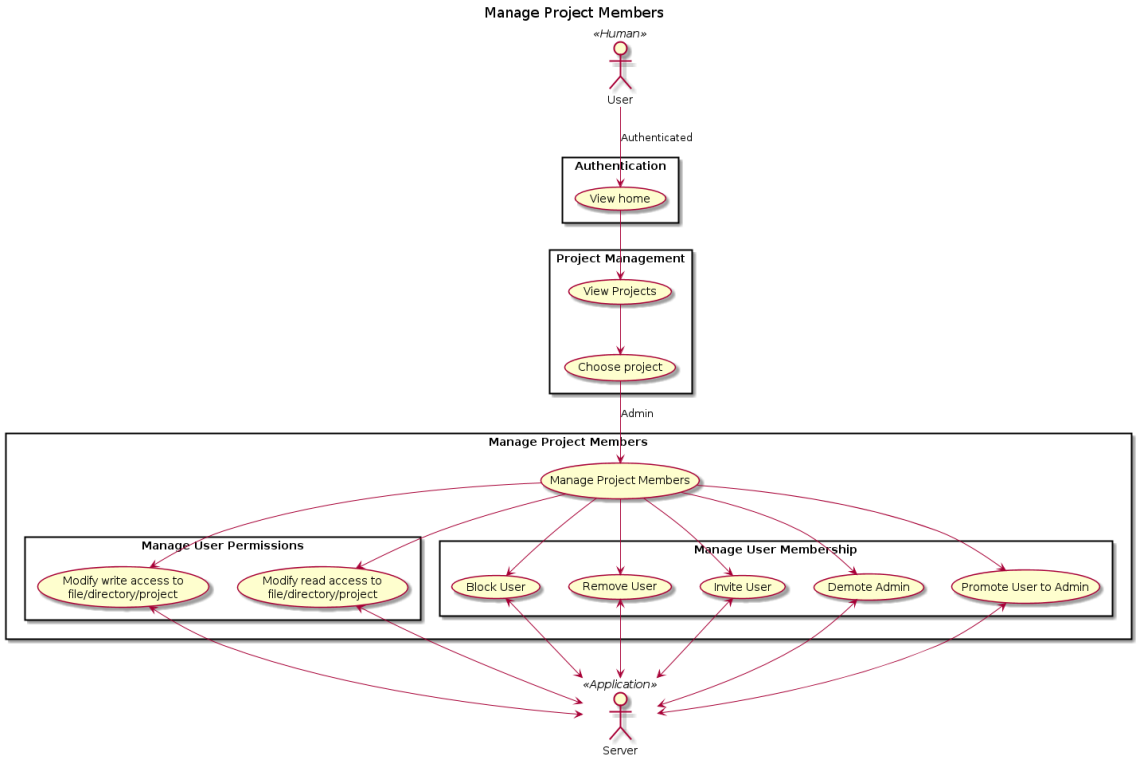
1.2.7 Project Management (sass8427)



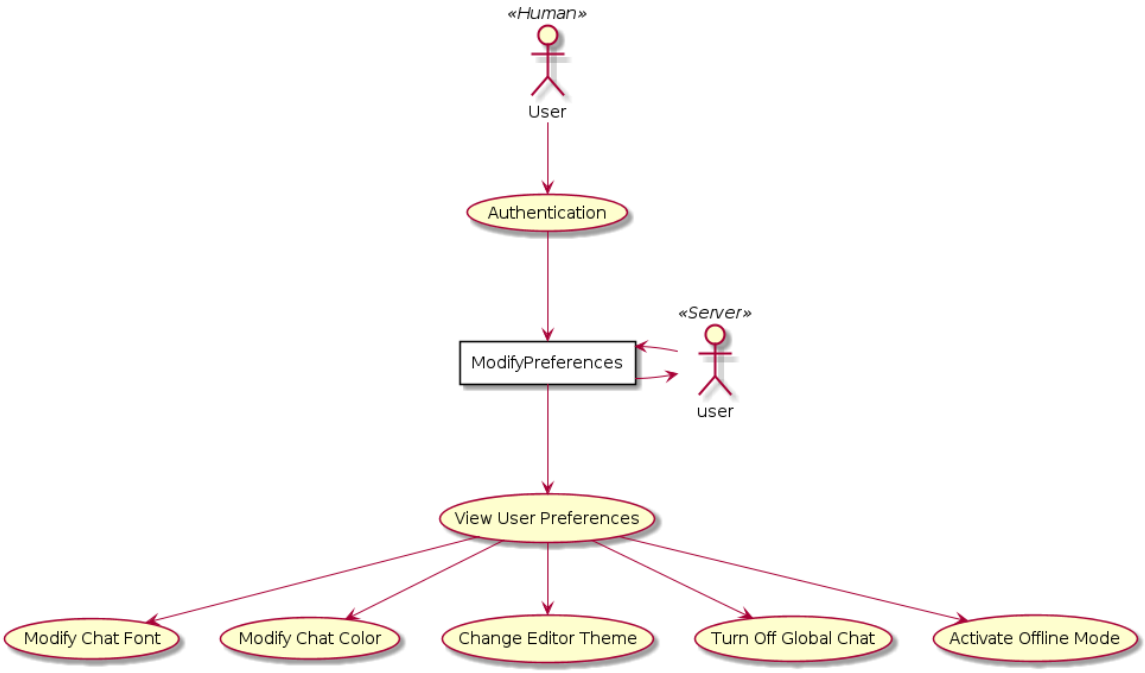
1.2.8 User Profile Management (bolt1003)



1.2.9 Project User Management (boss2849)



1.2.10 User Editor Preferences (snev7821)



1.3 Use Case Descriptions

1.3.1 Authentication (alsh5301)

Sign up (alsh5301)

Actors:	user.
Goals:	To register and create an account in the program.
Pre-conditions:	None.
Summary:	The user signs up and creates an account using their email address and creates username and password. Once they can access to the program.
Related use cases:	None.
Steps:	1. User is prompted to enter email and password. 2. System sends confirmation email. 3. User verifies email. 4. User clicks finish System presents user with additional info page. 5. User enters password, may choose to upload photo, works, etc. 6. System saves information, takes user to home page.
Alternative 1:	User already has an account.
Alternative 2:	User doesn't confirm email. Delete request after timeout period.

Sign in (alsh5301)

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 email and password
2. System verifies information
3. If info is correct user is presented with 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

---

Logout (alsh5301)

Actors:	Users.
Goal:	Existing user logs out
Precondition:	User must be logged in
Summary:	The user can log-out of the program after finishing process the project
Steps:	1. User clicks the "log-out" button. 2. System prompts user to ensure non-accidental click. 3. User verifies. 4. System logs user out. 5. User enters password, may choose to upload photo, works, etc. 6. System saves information, takes user to home page.
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.

1.3.2 Project Ideas (dani2918)

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.

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.

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.

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.

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.

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.

1.3.3 Communication (jank6275)

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.

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.

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.

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.

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.

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.

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 userś selection is saved in the database. 7. All further chat messages will use the selected font.
Alternatives:	None.
Post-conditions:	None.

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 userś selection is saved in the database. 7. All further chat messages from the user will use the selected color.
Alternatives:	None.
Post-conditions:	None.

1.3.4 File Management (wern0096)

Add New File to Project (wern0096)

Task Name: Add New File to Project

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 pane 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)

Task Name: Add Existing File to Project

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 pane 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)

Task Name: Delete File

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. Current project must have at least one file.

Steps:

1. User clicks 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.

Export Project (wern0096)

Task Name: Export Project

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 pane 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)

Task Name: Export Files

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 pane and the export location in the right pane.
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 pane 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)

Name: Open File in New Tab

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*.

Save File (wern0096)

Name: Save File

Category: File Management

Actor: User

Summary: Allows users save edited files.

Purpose: Saving files is essential in maintaining the integrity of a project.

Preconditions:

1. User is registered.
2. User is logged in.
3. User has a project open.
4. Project contains at least one file.
5. User has read permission.
6. User has write permission.
7. At least one file has been edited since the last save.

Steps:

1. User edits one or more files.
2. System enables a flag signaling that edits have occurred.
3. User clicks a save button or keybind.
4. System saves current open file.

Alternatives:

1. A *Save All* button that saves all open, edited files.

Postconditions:

1. The database is update to reflect the save operation.
2. The save button is grayed out.
3. The save all button is grayed out if it was clicked.

Close Saved File (wern0096)

Name: Close Saved File

Category: File Management

Actor: User

Summary: Allows users to close opened, saved files.

Purpose: Closing files is essential in managing the workspace and increasing productivity.

Preconditions:

1. User is registered.
2. User is logged in.
3. User has a project open.
4. User has at least one file open.
5. User has read permission
6. User has write permission.
7. Current opened file has no unsaved changes.

Steps:

1. User clicks the 'x' on the edge of an open tab.
2. System closes current open file.

Postconditions:

1. The opened tab and file are closed.

Related: Close Unsaved File

Close Unsaved File (wern0096)

Name: Close Unsaved File

Category: File Management

Actor: User

Summary: Allows users to close opened, unsaved files.

Purpose: Closing unsaved files is a possibly risky procedure. It is important that the use case reduces the chance of the user losing their work.

Preconditions:

1. User is registered.
2. User is logged in.
3. User has a project open.
4. User has at least one file open.
5. User has read permission
6. User has write permission.
7. Current opened file has unsaved changes.

Steps:

1. User clicks the 'x' on the edge of an open tab.
2. System presents a dialog telling the user that there are unsaved changes to the file.
3. System asks the use if he wants to *Save Changes and Don't Close*, *Save and Close*, *Close and Don't Save*, or *Cancel*.

Alternatives:

1. If the user selects *Save Changes*, the file is saved as per the Save File use-case.
2. If the user selects *Save and Close*, the file is saved as per the Save File use-case and closed as per the Close Saved File use-case.
3. If the user selects *Close and Don't Save*, the file is closed.
4. If the user selects *Cancel*, nothing happens.

Postconditions:

1. The database is updated if the user chose to save the file.

Related:

1. Close Saved File
2. Save File

1.3.5 File Editing (wern0096)

View Line Numbers (wern0096)

Name: View Line Numbers

Category: File Editing

Actor: User

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

Purpose: Makes it easier to communicate position in code. It is also a useful metric to have.

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 Line Numbers* option.
4. System displays line numbers to the left of the document.

Related:

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

View References (wern0096)

Name: View References

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

View Dates (wern0096)

Name: View Dates

Category: File Editing

Actor: User

Summary: Allows the user to view the last date that each line of a document was edited.

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 line of a document was edited.

Related:

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

View Authors (wern0096)

Name: View Authors

Category: File Editing

Actor: User

Summary: Allows the user to view the last author that edited each of line of the document.

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

Format Document (wern0096)

Name: Format Document

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

Find/Replace (wern0096)

Name: Find/Replace

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

Comment Section (wern0096)

Name: Comment Section

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

Display Typing User (wern0096)

Name: Display Typing User

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.

Display Syntax Errors (wern0096)

Name: Display Syntax Errors

Category: File Editing

Actor: User

Summary: As the user types code, the editor will underline syntax errors with a red line.

Purpose: Aids the user is writing correct code.

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 displays any syntax errors as a red underline under the incorrect section.

Related: Display Syntax Highlighting

Display Syntax Highlighting (wern0096)

Name: Display Syntax Highlighting

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

1.3.6 User Prefrences (snev7821)

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

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

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

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

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

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

1.3.7 Project Management (sass8427)

Create project (sass8427)

Actors:	User
Goals:	Create a new project with user specs.
Pre-conditions:	Project creator is logged in, viewing projects.
Summary:	In order for a user to create a new project, they must be logged in to sQuire and provide at a minimum a project title. Although optional, also adding a mission statement, description, and initial project files is highly encouraged on the basis that they will create more interest for the new project.
Related use cases:	Import file

Steps:

1. "New Project" is clicked by the user.
2. Project creation windows appears.
3. User adds a title in appropriate entry.
4. User adds any of the following (Optional): mission statement, description.
5. User decides whether or not to perform the task: "Import file to the project".
6. User clicks "Create".
7. System imports files and list project.

Alternatives:	None.
Post-conditions:	None.

Delete project (sass8427)

Actors: Admin or project owner.

Goals: Remove project from sQuire sever.

Pre-conditions: logged in, viewing projects.

Summary: Deleting a project is a task the owner may want to perform for the purpose of cleaning up no longer relevant projects. Also, a sQuire admin may find the need to delete projects, either to free up wasted resources, or to fulfill everyday managerial duties. A time to deletion can be added to give project members time to retrieve data if they desire before the project is actually removed.

Related use cases:

Steps:

1. Actor clicks delete icon on respective project.
2. Delete dialog opens.
3. Actor chooses whether or not to add a deletion time deadline.
4. Actor chooses whether or not to notify user of project deletion.
5. Actor presses Delete.
6. Confirmation window is displayed.
7. User confirms or disregards deletion.

Alternatives: Actor may click Cancel at any time.

Post-conditions: None.

Join project (sass8427)

Actors:	User
Goals:	Join a project created by some other user.
Pre-conditions:	logged in, viewing projects, have not already joined project.
Summary:	A simple task that allows user to join projects for the purpose of working collaboratively.
Related use cases:	
Steps:	1. User clicks join project. 2. Notification is sent to project owner for review.
Alternatives:	None.
Post-conditions:	None.

Unjoin project (sass8427)

Actors:	User
Goals:	Remove member status from project.
Pre-conditions:	logged in, viewing projects, member of 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 performs the task: “Choose project”. 2. User clicks ”Unjoin”. 3. User is promted 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.

Accept invite to project (sass8427)

Actors: User who received the invite (Invitee), sQuire user who sent the invite (Referrer), and Project Owner (May be the same as the Referrer).

Goals: Gain access to a project

Pre-conditions: Authorized user with access to the project sends an email invite to another user.

Summary: A project owner or member of can send email s to other sQuire user's. The invited user can accept the invite by clicking the link in the email at which point the link is deavtivated.

Related use cases:

Steps:

1. Invitee clicks on the link received by email.
2. New browser window opens with sQuire site
3. Message appears that invitee's account has been added to the access list for the Project.
4. The project is added to their Projects list.
5. Owner and Referrer (who may be the same) receive an email informing them a new collaborator has been added.
6. Project is opened in sQuire after a short delay.

Alternatives: 2B. If invitee is not logged in to sQuire, they are prompted for their credentials.
2C. If invitee does not have a sQuire account, they are prompted to first create an account.
3B. Project has been deleted. Invitee is informed the project does not exist and asks if they would like to create a new project with that name.
3C. Join link has been deactivated because of previous use/time limit. Invitee is informed that the invite has expired and is no longer valid, instructing them to contact the Project Owner for access.

Post-conditions: Link is deactivated.

Rate project (sass8427)

Actors:	User
Goals:	Provide basic project evaluation
Pre-conditions:	logged in, viewing projects
Summary:	Projects can be rated based on a five star rating scale. Although the rating system itself is largely superficial, it can help sQuire users find projects that interest them greater.
Related use cases:	Choose project.
Steps:	1. User performs the task: "Choose Project". 2. In viewing mode, user chooses to rate project (1 — 5 stars).
Alternatives:	None.
Post-conditions:	None.

1.3.8 User Profile Management (bolt1003)

View User Profile (bolt1003)

Actors:	Users of sQuire.
Goals:	Users views their public profile page.
Pre-conditions:	1. The user has registered for an account. 2. The user is logged in. 3. The user is at the dashboard.
Summary:	User opens their profile to see the public view of the profile.
Related use cases:	User modifies their profile.
Steps:	1. The user selects their avatar. 2. A drop down appears and the user selects "View public profile".
Alternatives:	None
Post-conditions:	None.

Modify User Profile (bolt1003)

Actors: Users of sQuire.

Goals: Users updates their profile.

Pre-conditions:

1. The user has registered for an account.
2. The user is logged in.
3. The user is at the dashboard.

Summary: User navigates to their profile to change their username.

Related use cases: The user modifies their email address or biography.

Steps:

1. The user select their avatar.
2. A drop down appears and the users selects "My profile".
3. The user selects modify.
4. The user changes their username.
5. The user clicks update to save the changes.
6. The user clicks finish to disable modify.

Alternatives:

1. Step 4: The user changes their password.
2. Step 4: The user changes their email.
3. Step 4: The user changes their "About Me" information.
4. Step 4: The user changes their name.

Post-conditions: The user returns to the dashboard or exits sQuire.

Modify notifications from friends (bolt1003)

Actors: Users of sQuire.

Goals: Users modifies email updates from friends.

Pre-conditions:

1. The user has registered for an account.
2. The user is logged in.
3. The user is at the dashboard.

Summary: User navigates to the privacy manager and changes notifications options for people in their friends list.

Related use cases: The user modifies global freind permissions.

Steps:

1. The user select their avatar.
2. A drop down appears and the users selects "My profile".
3. The users selects manage privacy.
4. The user selects friend communication management.
5. The user selects a friend.
6. The user click the toggle switch next to "Receive email updates when this friends messages me".

Alternatives:

1. Step 5: The user clicks the toggle switch next to "Display email address to this friend".
2. Step 5: The user clicks the toggle switch next to "Display name to this friend".

Post-conditions: The user returns to the dashboard or exits sQuire.

Modify public messaging preferences (bolt1003)

Actors: Users of sQuire.

Goals: Users modifies settings to allow messages from any user on squire.

Pre-conditions:

1. The user has registered for an account.
2. The user is logged in.
3. The user is at the dashboard.

Summary: User navigates to the privacy manager and changes messaging public messaging options.

Related use cases: The user changes public notifications preferences.

Steps:

1. The user select their avatar.
2. A drop down appears and the users selects "My profile".
3. The users selects manage privacy.
4. The user selects manage public privacy settings.
5. The user click the toggle switch next to "Allow messages from any sQuire user".

Alternatives:

1. Step 5: The user clicks the toggle switch next to "Display email address to the public".
2. Step 5: The user clicks the toggle switch next to "Display name to the public".

Post-conditions: The user returns to the dashboard or exits sQuire.

Modify global email settings (bolt1003)

Actors:	Users of sQuire.
Goals:	Users modifies settings to allow email notifications.
Pre-conditions:	1. The user has registered for an account. 2. The user is logged in. 3. The user is at the dashboard.
Summary:	User navigates to the privacy manager and changes email preferences.
Related use cases:	Change public or freind notifications.
Steps:	1. The user select their avatar. 2. A drop down appears and the users selects "My profile". 3. The users selects manage privacy. 4. The user clicks the toggle switch next to "Allow email notifications".
Alternatives:	None.
Post-conditions:	The user returns to the dashboard or exits sQuire.

1.3.9 Project User Management (boss2849)

Modify read/write access (boss2849)

Actors:	User
Goals:	Modify a userś permissions.
Pre-conditions:	User is signed in and holds Admin rights for the currently selected Project
Summary:	User modifies another Userś 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 modifes the target Userś permissions and notifies them.
Alternatives:	3. User selects cancel, System discards changes.
Post-conditions:	None.

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.

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) dont́ match any users, System notifies User of failed invitiations.

Post-conditions:	None.

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.

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.

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.
