HW4: Use Cases

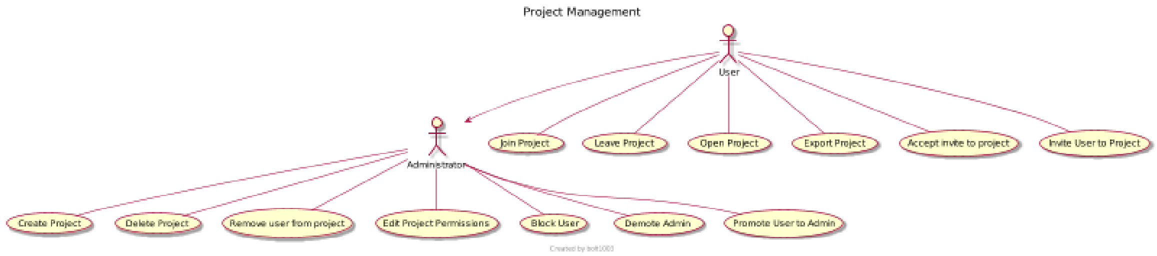
bolt1003

March 5, 2016

Chapter 1

Use Case Diagrams

1.1 Project Management (bolt1003)



Chapter 2

Sequence Diagrams

2.1 Compiler (bolt1003)

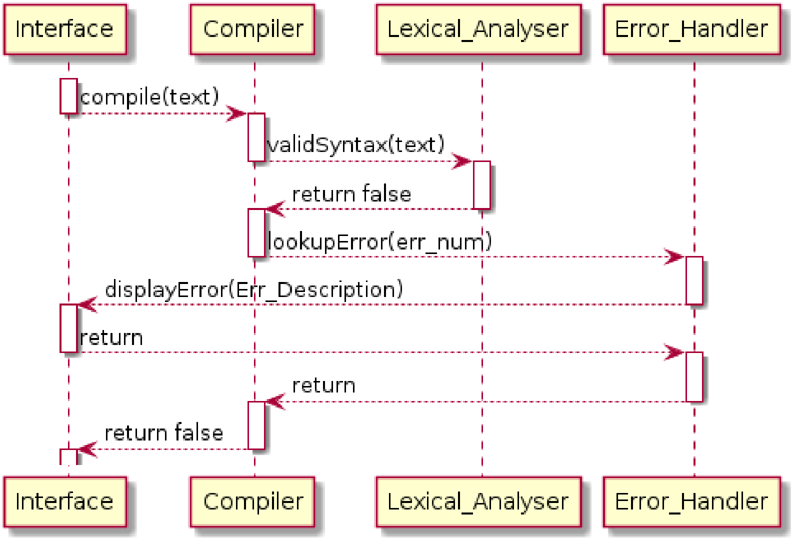


Figure 2.1: The sequence of a user attempting to compile a program with a syntax error.

Chapter 3

Class Diagrams

3.1 Project Management (bolt1003)

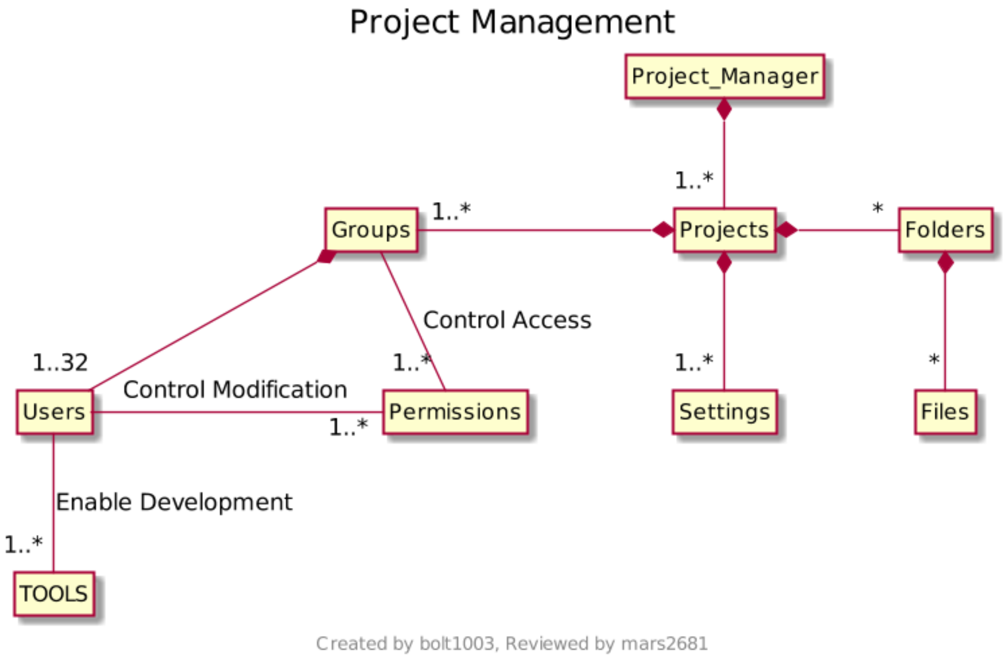


Figure 3.1: Project management allows for the group to delicate permissions and create projects

Chapter 4

Requirements Documentation

4.1 Non-Functional Requirements

Non-functional requirements cover all the remaining requirements which are not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviours. Squire's non-functional requirements are:

- sQuire will leverage technology developed for the web to ensure reliablity. Technologies such as redunant hardware, redunant power providers, redunant internet services, load balancing and virtualization will enable sQuire to be reliable.(bolt1003)
- sQuire will run in a virtual machine on top of redunant hardware. Using a virtual machine allows for mutliple instance to be running and tested. The backend will run on redunant hardware which will prevent hardware failure from affecting sQuire usage. In turn, allowing infrastructure to be serviced without affecting sQuire. (bolt1003)

4.2 Functional Requirements

Functional requirements will specify a behaviour or function. Squire's functional requirements are:

4.2.1 Communication (bolt1003)

- Global text chat.
- Per project local chat.
- Private chat between users.
- A friends list with friend status icons and avatars.
- A global list of members in the project and status.
- Use of standard protocols such as XMPP or IRC.
- Third-party chat client access.

Chapter 5

Application Domain Specification

5.1 Project Management (bolt1003)

5.1.1 Create Project

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 choosen 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.

5.1.2 Open a project

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.

5.1.3 Join Project

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 inventation to join a project.

Post-conditions: None

5.1.4 Leave project

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 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.

5.1.5 Delete Project

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.

5.1.6 Export Project

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.

5.1.7 Accept Invite to Project

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.

5.1.8 Remove User from Project

Actors: User of sQuire

Goals: Revoke access to the Project for a single or multuple 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.

5.1.9 Edit Project Permissions

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.

5.1.10 Modify read/write access

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.

5.1.11 Remove User

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.

5.1.12 Invite User to Project

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.

5.1.13 Promote User to Admin

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.

5.1.14 Demote Admin

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.

5.1.15 Block User

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.

5.2 File Management (Created by snev7821, revised by bolt1003)

5.2.1 Add New File to Project (snev7821)

Actors: User of sQuire

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

Pre-conditions:

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.

Post-conditions:

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

Related: Add Existing File to Project

5.2.2 Add Existing File to Project (snev7821)

Actors: User of sQuire

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

Pre-conditions:

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.

Post-conditions:

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

Related: Add New File to Project

5.2.3 Delete File (snev7821)

Actors: User of sQuire

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

Pre-conditions:

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.

Post-conditions:

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

Related: Delete Project

5.2.4 Export Files (snev7821)

Actors: User of sQuire

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

Pre-conditions:

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

5.2.5 Open File in New Tab (snev7821)

Actors: User of sQuire

Summary: Allows users to open a file.

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

Pre-conditions:

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.

5.3 File Editing (snev7821)

5.3.1 View Line Numbers (snev7821)

Actors: User of sQuire

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

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

Pre-conditions:

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

5.3.2 View References (snev7821)

Actors: User of sQuire

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

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

Pre-conditions:

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

5.3.3 View Dates (snev7821)

Actors: User of sQuire

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.

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

Pre-conditions:

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

5.3.4 View Authors (snev7821)

Actors: User of sQuire

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.

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

Pre-conditions:

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

5.3.5 Format Document (snev7821)

Actors: User of sQuire

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

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

Pre-conditions:

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

5.3.6 Find/Replace (snev7821)

Actors: User of sQuire

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

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

Pre-conditions:

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

5.3.7 Comment Section (snev7821)

Actors: User of sQuire

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

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

Pre-conditions:

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

5.3.8 Display Typing User (snev7821)

Actors: User of sQuire

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

Goals: Differentiate who is typing what.

Pre-conditions:

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.

5.3.9 Display Syntax Highlighting (snev7821)

Actors: User of sQuire

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

Goals: 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

5.4 User Prefrences (snev7821)

5.4.1 View User Preferences (snev7821)

Actors: User of sQuire

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

Goals: Allows user to view their preferences and change them

Pre-conditions:

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

5.4.2 Modify Chat Font (snev7821)

Actors: User of sQuire

Summary: User changes the chat font

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

Pre-conditions:

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

5.4.3 Modify Chat Color (snev7821)

Actors: User of sQuire

Summary: User changes the chat color

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

Pre-conditions:

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

5.4.4 Edit User Editor Theme (snev7821)

Actors: User of sQuire

Summary: User changes the Editor theme

Goals: Allows user to change theme of the collaborative editor

Pre-conditions:

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

5.4.5 Turn Off Global Chat (snev7821)

Actors: User of sQuire

Summary: User turns off global chat

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

Pre-conditions:

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

5.4.6 Psuedo-offline Mode (snev7821)

Actors: User of sQuire

Summary: User changes to offline mode

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

Pre-conditions:

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