

Year and Semester: 2016 Spring
Course Number: CS-383
Team Name: Sovereign
Work Number: HW#3
Work Name: Requirements Document
Work Version: Version 01
Long Date: Monday 08 February 2016
Group: Group 1
Contributors: Dillon Harris, Chris Campbell, Hue Purkett, Cody Helbling, Afnan Alsharif,
Austin Sass, Jesse Frantzich, Tyler Songstad

Emails: harr5275, Camp7325, purk2552, helb4651, alsh5301,
sass8427, fran6139, song6803

Abstract

This is HW#3 for CS-383 – Software Engineering. The purpose of this document is to list our functional and non-functional requirements for the sQuire project. These requirements have been edited and voted in by the group and represent the basic functionality sQuire will be designed to have.

Contents

1	Functional Requirements:	3
1.1	Client Funtionality:	3
1.2	File Management:	3
1.3	Server Functionality	4
1.4	Project Management:	4
1.5	Chat Management:	5
2	Non-Functional Requirements:	6
3	Use Cases:	7
4	Use Case Diagrams:	9
5	Class Diagrams:	10

1 Functional Requirements:

1.1 Client Funtionality:

Requirement: Capable of supporting editing of Java programs.

Description: Users will be able to edit any file save in Java format in the IDE.

Requirement: Capable of compiling Java programs.

Description: Users will be able to compile correctly formed Java files saved in the .java format.

Requirement: Capable of executing Java programs.

Description: User will be able to execute Java files that were successfully compiled.

Requirement: Multiuser Support.

Description: Projects allow the participation of up to 10 users. It was decided that 10 users would be a better requirement instead of 32. 32 users working on one document could lead to a large amount of errors such as too many people trying to work on one section of code. A good quote that come to mind is too many cooks in the kitchen.

Requirement: Shared Sessions.

Description: Users will be able to snap to the view of other contributors.

Requirement: Easily controlled viewports.

Description: User will be able to independently control their own window and snap to the view of other contributors.

Requirement: Chat Widget.

Description: The Chat widget will be located off to the side of the IDE, and the Chat widget will show people who are currently online on the same project level as the current user.

1.2 File Management:

Requirement: Ability to import projects from User's local file system.

Description: Projects can be brought in from common sources such as Github, local computer, and other IDEs.

Requirement: Ability to export projects from User's local file system.

Description: Projects can be able to be exported to common sources such as Github, local computer, and other IDEs.

Requirement: Creating File.

Description: User will be able to create a file in the project.

Requirement: Deleting File.

Description: User will be able to delete a file in the project.

1.3 Server Functionality

Requirement: Lock user's line.

Description: . This feature will prevent multiple users from trying to edit the same line of code.

Requirement: User Profiles.

Description: User will have persistent profiles including email, a profile picture, and project ownership/membership which are viewable by all other users. User profiles will also keep track of how many reviews the user has been involved in via achievements.

Requirement: User Achievements.

Description: User's profile will keep track of achievements such as: most lines of code written, number of reviews user has been in, amount of time the user has spent on the project, etc.

Requirement: Project Membership.

Description: This will be linked to User Profiles. Project membership will show who the owner of the project is, and who else is working on the project.

1.4 Project Management:

Requirement: Create Project.

Description: User will be able to create a project and edit its initial settings.

Requirement: Deleting Project.

Description: Admin will be able to delete an entire project.

Requirement: Invite User to Project.

Description: Admin will be able to send an invitation to a user to join a project.

Requirement: Kick User from Project.

Description: Admin will be able to unassociate a user with the project.

Requirement: Download Project.

Description: User will be able to download the project to a location on their local computer.

Requirement: Import Project.

Description: User will be able to import any prior saved projects into the IDE to work on the project for further improvements.

Requirement: Project File Structure Pane.

Description: The users will be able to see the project file structure. Also the user will be able to move to different files by clicking on icons within the structure.

Requirement: User Awareness.

Description: User will be able to see what other users are working on in a file/project, by showing blinking cursors of different colors like Google Documents.

1.5 Chat Management:

Requirement: User Chat at File level.

Description: User can send and receive chat messages with other individuals working on a particular file.

Requirement: User Chat at Project level.

Description: User can send and receive chat messages with users working on a project.

Requirement: Ability to Rate Comments.

Description: Users reviewing code will be able to up or down-vote comments other users have posted on the code in a project. This will help to ensure code reviews to be as constructive and useful as possible.

Requirement: Project Forum

Description: Public projects will be organized in a forum that facilitates project browsing, joining, and rating.

2 Non-Functional Requirements:

Requirement: Resource Defense System.

Description: System will be resistant to unsafe practices such as resource hogging and tampering with subsystems.

Requirement: Voice Chat.

Description: User will be able to chat with users in same project using built-in voice chat.

Requirement: Contributor History.

Description: Project will include and display an indicator of history of which user contributed/edited which sections of code with timestamps.

Requirement: Syntax Coloring.

Description: Projects will include visual indication of syntax correctness and give clues on how to correct syntax errors.

Requirement: Highlighting of User Edits

Description: Any user will be able to easily highlight another user's edits. In this scenario, edits by other users will be grayed out.

3 Use Cases:

Author Name: Dillon Harris | **Author Email:** harr5275

Task Name: Lock User's Line

Task Category: Server Functionality, Functional System Requirement

Summary: User's will have the option of locking any lines if and only if the user's cursor is the only cursor on the line.

Steps:

1. Click on Settings menu in the File menu bar.
2. Click on enable Line Lock.

Author Name: Dillon Harris | **Author Email:** harr5275

Task Name: Project Membership

Task Category: Server Functionality, Function System Requirement

Summary: User's are able to click a setting and see all the user's associated with the project.

Steps:

1. Click on Project menu in the File menu bar.
2. Click on Show Membership.
3. Menu will pop up showing all user's associated with project.
4. Click on Ok button to close menu.

Author Name: Dillon Harris | **Author Email:** harr5275

Task Name: View User Achievements

Task Category: Server Functionality, Functional System Requirement

Summary: User achievements are view able from the user account's page.

Steps:

1. Click on Account menu in the File menu bar.
2. click on User Achievement.
3. Menu will pop up showing User achievements for the user.
4. Click on Close button to close User Achievement menu.

Author Name: Dillon Harris | **Author Email:** harr5275

Task Name: Login to Account

Task Category: Server Functionality

Summary: User will log into their account after opening application.

Steps:

1. Click on Account menu in File menu bar.
2. click Login.
3. Login Menu will pop up.
4. Enter username and password for account
5. Click Login button.

Author Name: Dillon Harris | **Author Email:** harr5275

Task Name: Show Project Membership

Task Category: Server Functionality, Functional System Requirement

Summary: User will be able to see all members who are working on the project.

Steps:

1. Click on Project menu in File menu bar.
2. Click on View Project Membership.
3. Pop up menu shows all users associated to the current project.
4. Click Ok button to close menu.

4 Use Case Diagrams:

5 Class Diagrams: