Year and Semester:	2016 Spring
Course Number:	CS-383
Team Name:	Sovereign
Work Number:	HW#3
Work Name:	Requirements Document
Work Vesion:	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:

Requirement: Capable of supporting editing, compilation, and execution of Java programs.
Description: Programs will be viewed as projects and include multiple directories and files within an overall directory.

Requirement: Ability to import/export 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: Shared Sessions with easily controlled viewports.
Description: User will be able to independently control their own window and snap to the view of other contributors.

Requirement: User Chat to several levels of other users.
Description: User can chat with and view messages from file group, project group, and individuals.

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 Awareness.
Description: User will be able to see what other users are working on in a file/project, or who are online in the case of friends.

Requirement: Ability to Rate Comments.
Description: Users reviewing code will be able to up or down-vote comments, which help make the code better. This will help to ensure code reviews to be as constructive and usefull as possible.

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

Requirement: File Management.
Description: Users will be able to have complete file management through a “File” menu. File management operations will include “Saving a file”, “Loading a file”, etc.

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 as well as those working on a different file for the project.

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. Allowing only 10 users mitigates this problem.

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: 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: 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.