Requirements Specification

Uninitialized Local Team: Kaveh Buenaventura, Angela Ferro Capera, Jason Dekema, Ian Dudder, Kez May, Hailey Schauman.

1. Introduction

The purpose of this document is to specify the requirements for the collaboration tool's functionality. The document requirements are separated into Basic Requirements and Delighter Requirements. The Basic Requirements entails everything that the system must include to be functional. The Delighter Requirements are not necessary for the functionality of the system, but are necessary to set the application apart and draw in users.

2. Scope

This application seeks to create an online space for university and college students to collaborate on a class group project. The application seeks to connect students and provide them with the tools they need to collaborate. The basic functionality consists of a shared cloud file system, a collaborative file editor demonstrating real-time changes to all members working in the document, task management functionality, and a discussion board to allow team communication. The delighter functionality consists of an extensions page for users to download additional functionality to the application and an in-app voice-call service.

3. Definitions

3.1 Shared Space - the group project page containing the project requirements, shared storage, and all other collaboration tools. It is accessible to all members of the team and no one more.

3.2 Live collaboration - real-time editing of a file reflecting changes to all team members present in the document.

3.3 Kanban board - a project management tool allowing a project to be divided into different tasks and assigned to specific team members. The tasks are organized into columns on the board to specify which state the task is in.

3.4 CRUD - Create, Read, Update, Delete Action.

3.5 Creator - the person who creates and sets up the shared space.

3.6 Team manager - the person responsible for task management.

3.7 Team member - any person in the shared space.

4. Basic Requirements

4.1 Functional Requirements

4.1.1. The creator shall be able to manually assign team members into groups.
4.1.2. The creator shall be able to randomly assign team members into groups.
4.1.3. The creator shall be able to message team members and groups directly and list the specific requirements for each assignment in their course/s.
4.1.4. Creators shall be able to distribute class resources to each group.
4.1.5. The real-time collaboration tool shall be able to show version control history.
4.1.6. The shared storage shall allow named directories to organize the files.
4.1.7. The team member shall be able to perform CRUD actions on the real-time collaboration files.
4.1.8. The owner of a file shall get to choose which team members get to edit their files.
4.1.9. Team members shall be able to live collaborate to edit files including:
- Documents
- Sheets
- Drawings
- Etc.

4.1.10. The application shall allow an annotation feature for stored images and a comment feature for stored documents.
4.1.11. The application shall show where the cursors are for every user active in a live-collaboration document.
4.1.12. The application shall highlight syntax for any program stored in the shared cloud.
4.1.13. Team managers and team members should be able to visualize task progress in the task management section.
4.1.14. A task on the kanban board shall be linked to the assignment description or requirements so that it is easier to understand the task requirements.
4.1.15. Team managers and team members must be able to
- Update their progress on the task.
- Perform CRUD operations on task cards.
- Assign tasks to team members
- Perform CRUD operations to columns of a task board

4.1.16. Team members shall view task backlog in a configurable way as
- To-do list
- Kanban
- MoSCoW
- Etc.

4.1.17. The application shall be allow to filter task by
- Filter by assignee
- Filter by milestone

- ○ Filter by project
- ○ Filter by label

4.1.18. The application shall make use of graphs to represent the progress of

- ○ Individuals on assigned tasks
- ○ Milestones
- ○ Overall project

4.1.19. The application shall allow team members to submit their assignments to the application.

4.1.20. The application shall have a discussion board for all-team communications.

4.2 Non-Functional Requirements

4.2.1. The application shall be available as a desktop application and a mobile application.

4.2.2. The application shall be compatible with

- ○ Windows,
- ○ Linux,
- ○ MacOS,
- ○ Android, and
- ○ iOS.

4.2.3. Team members shall be able to upload and download documents of any file type to a shared cloud storage space of 15 gigabytes per shared space.

4.2.4. The shared storage shall not limit the size of an individual file upload unless it exceeds the total storage space

4.2.5. The live collaboration must be highly responsive, and reflect any changes a user makes to all other users within 5 seconds.

4.2.6. Files must not lag given more objects in a file

4.2.7. The live collaboration feature must display where active users' cursors are within 3 seconds of them moving their cursor.

4.2.8. The application shall allow the creator of the shared space to upload project requirements and resources of up to 2 gigabytes.

4.2.9. The application shall scale out as needed (depending on traffic)

4.2.10. The application shall guarantee 99.8% of availability

4.2.11. The application shall provide users with the following detailed error messages:

- ○ Incorrect login info.
- ○ Course sections that are not available.
- ○ Incorrect format of files trying to be submitted.

4.2.12. The UI and clickable elements on the application should be incredibly responsive with 1 second or less of delay

4.2.13. The application must provide adequate security features including log-in authentication, editing authorization, password recoverability/changing, data confidentiality, and extension moderation/screening

4.2.14. The shared cloud storage must have reliable data preservation.

5. Delighter Requirements

5.1 Functional Requirements

5.1.1 The application shall allow new functionality to be added on the extensions page.

5.1.2 The application shall allow a voice-call server for members of the group to join.

5.1.3 The application shall allow assignments to be linked to group folders containing the necessary contents of the assignment.

5.1.4 The application shall allow assignment requirements to be linked to files or to highlighted text-blocks.

5.1.5 When a team member is on the submission page for an assignment

a. The application shall suggest the linked files/folder for submission.

b. The application shall display the assignment requirements on the same page.

c. When a team member clicks/taps a requirement, the application shall display a preview of the file containing the section that fulfills the requirement

5.2 Non-Functional Requirements

5.2.1 The application must be open source to allow third parties to create and add new functionality to the extensions page.

5.2.2 The voice-call service must preserve high quality sound transmission with minimal static and no more than a quarter-second of delay.

5.2.3 The application should be easily scalable to account for new extensions and features.