

SENG 513 - Group 17 - Project Proposal

ExVault

March 13th, 2022

Group Members:

Youssef Abdel Maksoud

Ezra Guia

Irtaza Ansari

Nathan Bhandari

Table of Content

Introduction:	3
User requirements:	3
Should Do	3
Could Do	3
Primary User Group	3
Secondary User Group	3
Functional and implementation requirements	4
Proposed technologies	4
Backend	4
Frontend	4
Appendix	4

Introduction:

The project, **ExVault**, is a file sharing web application. The file(s) can be stored online to be accessed by another intended user. There have been issues in the past with coming up with a system-agnostic way of transferring files quickly to other users. For example, transferring a file from an Android phone to a linux computer is difficult without the use of something like Google Drive. However, for small files it can be a little excessive to set up a folder, upload the file, and then share a link. **ExVault** would allow this kind of transaction to be completed quickly by having users invite each other to public or private “chat rooms”. Once in the room users will be represented by an avatar. By dragging and dropping files over each user one can share with a specific person or the whole room. Rooms are not remembered but under their account each user can see files they have uploaded and received. The file will be stored online for a certain period of time due to pricing constraints and will have a limitation as to how large the file is.

User requirements:

Should Do

- *Access across multiples devices*
- *Share files*
- *Delete files (NO TRANSCRIPTS WHATSOEVER)*
- *Create new files, resume files (pause syncing) and download files.*

Could Do

- *Utilize web-browser extension to render different types of files*
- *Direct file share with a specific person OR with the entire group.*
- *Set up different permissions for strangers versus known contacts*

Primary User Group

- Users who will be added into the system to share their files in a chat room making it accessible to anyone to share their files.

Secondary User Group

- For the privacy concerned users, passwords and usernames can be created to access specific files. The chat rooms provide authentication since a login is required.

Functional and implementation requirements

- Authentication (username, password)
- Multiple chat rooms (maybe regarding the topic of discussion of the room)
- List of room users
- Authentication for rooms (servers) with people you know
- No authentication required for rooms with people you don't know

Proposed technologies

Backend

Node.js, fs, formidable, MongoDB

Frontend

HTML, CSS, React.js

Node.js is required to use the fs(file system) module which allows the parsing of files from a user and the formidable module which allows for things like file uploading to the browser. HTML, CSS and a front end library will be used to quickly enhance the user experience by guaranteeing a professional interface that is responsive and works with all browsers.

Appendix

- <https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/JavaScript-No-dejs-File-Upload-Example-Ajax>
- <https://codepen.io/aybukeceylan/pen/yLOxRyG>