

# Google Workspaces Alternative Application

## *Requirements and Specifications*

Wahkiakum School district, Naselle School district, and Wahkiakum 4-H



**Team Toto**

Albert Lucas

Ben Kaufmann

Daniel Semenko

<b>Introduction</b>	<b>3</b>
<b>System Requirements Specification</b>	<b>3</b>
Use Cases	4
Functional Requirements	24
Non-Functional Requirements	27
<b>System Evolution</b>	<b>28</b>
<b>Glossary</b>	<b>29</b>
<b>References</b>	<b>30</b>

## **I. Introduction**

Wahkiakum School district, Naselle School district, and Wahkiakum 4-H are in need of an application that allows student users to communicate with each other across classrooms, clubs, groups, etc. To satisfy these district needs, our team is planning on analyzing, designing, and building a stand-alone application from scratch to improve student learning and connection.

This document is a Software Requirements Specification, which contains plans for Team Toto's communication application. It will go into detail about who is involved with the project, the different users and stakeholders, use cases, and requirements for the application.

This document is written to lay out the functional and non-functional requirements for our communication application. This will also serve as a basis for the agreements between Team Toto, Ananth Jillepalli, our mentor, and Ron Wright, our client.

Our hope with this document is to inform readers of all the ways that users will interact with our application, and how the interaction differs among user types. In addition, we will list the requirements that any stakeholders may need for the application.

## **II. System Requirements Specification**

This section specifies the software product's requirements. This section specifies all the software requirements to a level of detail sufficient to enable designers to design a software system to satisfy those requirements, and to enable testers to test that the software system satisfies those requirements.

In project management, the clients are the ones who define the requirements of the project and set the parameters such as budget and deadlines. Therefore, for Team Toto's communication application, the clients are the school district who are instating the use of this project. They need this online platform to allow for various methods of communication between students and faculty. These faculty members are the ones setting the requirements and specifications that the program must follow. They also help oversee the deadline for completion.

The users, on the other hand, would be anyone who actively uses the application. In the case of this project, the potential users could be students or faculty. Students can use the application to join class spaces and communicate with faculty or other users through direct or voice communication. The faculty, on the other hand, are the ones creating, using, and moderating these class spaces, while also being able to monitor student communications. Faculty can get an overview of the students who are in their spaces.

Finally, the stakeholders are anyone who is affected by the implementation of the project. The point of this project from the beginning is to connect faculty and students. Therefore, the potential stakeholders would include the faculty, who are able to create, use, and monitor class spaces and personal chats. Students are also stakeholders, as they are given the opportunity to join classes and participate in class space activities, and communicate with faculty or other students through private or voice communications. In addition, there could be administrators who aren't connected to any one class space, but through the implementation of the program, are able to moderate class spaces as needed.

## II.1. Use Cases

### Use Case #1

Name	Create Student Account
Users	Faculty
Rationale	People should not be allowed into the application if they are not verified students. Thus, faculty should go through and create accounts for students, with their school ID as part of their account. Once a student account is created, students can change their password to whatever they want.
Triggers	The faculty clicks the “Create Student Account” button on the sidebar.
Preconditions	Faculty is on any page with a view of the sidebar.
Actions	<ol style="list-style-type: none"><li>1. The faculty is taken to “Student Creation” page</li><li>2. The faculty fills out the minimum required information for student account creation (ID, username, temporary password)</li><li>3. Student account information is stored in the database for later updating and retrieval.</li><li>4. The Faculty is returned to the main home page</li></ol>
Alternative Paths	If the user’s registration information already exists, then don’t add the user to the database and display an appropriate error message on redirect.
Postconditions	The student account exists in the database.
Acceptance Tests	The student account is found in student listings and can join spaces or private communications.

## Use Case #2

Name	Create Faculty Account
Users	Faculty
Rationale	Regular account creation, but it must have added verification to confirm that faculty accounts are actually linked to a faculty member. Faculty, with proper verification, can then create and fill the rest of their account.
Triggers	The faculty clicks “Create Faculty Account” button on the login screen.
Preconditions	Faculty is on the “Login” page.
Actions	<ol style="list-style-type: none"><li>1. User fills out all fields with their contact information</li><li>2. The faculty information is stored for later retrieval.</li><li>3. The user is redirected to the login page, with a message indicating that they registered successfully.</li></ol>
Alternative Paths	If the user’s registration information already exists, then don’t add the user to the database and display an appropriate error message on redirect.
Postconditions	The faculty exists in the database.
Acceptance Tests	The faculty is found in faculty listings and can create class spaces, utilize class space functions, and monitor and edit class posts/comments/communications.

### Use Case #3

Name	User Login
Users	Students, Faculty
Rationale	As long as the user's account exists in the database, users should be able to log on with a user and password. Users need exclusive access to the application.
Triggers	The user clicks the "Login" button after entering their username and password.
Preconditions	User is not logged in, user account already exists
Actions	<ol style="list-style-type: none"><li>1. User enters username and password and presses the "log in" button.</li><li>2. User is authenticated, then taken to the main page.</li></ol>
Alternative Paths	User account does not exist yet, can press "Create Faculty Account" to go to the Faculty Account Creation page.  User may provide incorrect username or password, returned to empty log in page with corresponding message.
Postconditions	User is logged into the correct user-type page. That is, a student will be taken to the student view page, while the faculty will be taken to the faculty view page.
Acceptance Tests	A student member can log in and be taken to their home page, but can not create class spaces. A faculty member can log in and be taken to their home page and is able to see their class spaces or create new ones.

#### Use Case #4

Name	Logout
Users	Students, Faculty
Rationale	Users should be able to log out.
Triggers	User clicks “Log Out” button.
Preconditions	User is logged in, on any page.
Actions	<ol style="list-style-type: none"><li>1. User is logged out.</li><li>2. User is taken to login screen</li><li>3. “Successful logout” message is displayed on the screen.</li></ol>
Alternative Paths	None
Postconditions	The user is successfully logged out.
Acceptance Tests	After logging out, no user is currently logged in.

## Use Case #5

Name	View Account
Users	Student, Faculty
Rationale	Any user should be able to view their own account, or any other user's account (student or faculty). They should be able to find a user through a class space, or through the user search on the sidebar.
Triggers	User clicks another user's profile block. User clicks their own profile block.
Preconditions	User is logged in, on any page with the sidebar.
Actions	<ol style="list-style-type: none"><li>1. User clicks another user's profile block, or their own "My Profile" button.</li><li>2. User is taken to the page, where their/the users information is displayed.</li></ol>
Alternative Paths	None
Postconditions	The queried user profile page is opened up, where profile information is visible.  The user's own profile page is opened up, where their personal profile information is visible.
Acceptance Tests	The correct account information is displayed.



## Use Case #6

Name	Edit Account
Users	Students, Faculty
Rationale	<p>Users shall be able to edit their account details, such as profile description, user image, contact information, etc.</p> <p>Faculty can edit student accounts (in case moderation is needed), but only modify profile description and profile picture.</p>
Triggers	<p>User clicks “Edit Profile” button on own profile page.</p> <p>Faculty clicks “Edit Profile” button on student profile page.</p>
Preconditions	<p>User is logged in, user on own profile page.</p> <p>Faculty is logged in, viewing student profile page.</p>
Actions	<ol style="list-style-type: none"> <li>1. Taken to Edit Profile page. Student users can only have their profile picture and description modified. Faculty can change everything else, except for log in information.</li> <li>2. After clicking “save”, updated information is stored in database.</li> <li>3. User is returned to previous profile page.</li> </ol>
Alternative Paths	If user presses “revert changes,” or leaves the page without saving the changes, return to the profile page without updating the database.
Postconditions	The updated information exists in the database.
Acceptance Tests	<p>Users should open their profile, make and save changes, and see those changes updated.</p> <p>Faculty should open a student profile, make and save changes, and see those changes updated.</p>

## Use Case #7

Name	Delete Account
Users	Faculty
Rationale	Only Faculty can delete either a student's account, or their own account. A student should not be able to randomly delete their account.
Triggers	Faculty clicks "Delete Profile" button on profile page.
Preconditions	Faculty should be on their own profile page, or viewing a student's profile page.
Actions	<ol style="list-style-type: none"><li>1. After clicking "Delete Profile" button, faculty should have to confirm selection (no accidental deletions).</li><li>2. All posts, comments, and/or related class spaces belonging to the deleted user should be deleted.</li><li>3. Students that are part of a class of a now deleted faculty are redirected to their home page, with a message stating that the class has been deleted.</li></ol>
Alternative Paths	If user presses "no" on the confirmation, user returns to previous page.
Postconditions	Account is no longer visible in user listings, not in the database.
Acceptance Tests	Logging in as the faculty, and deleting a students account should make their account and all their posts and comments not visible anymore, nor should it be able to be logged into.

## Use Case #8

Name	Create Post
Users	Faculty and Students
Rationale	Faculty and Students must be able to create a post on Post Space page
Triggers	The user clicks "Create Post"
Preconditions	The user is in logged in
Actions	<ol style="list-style-type: none"><li>1. The use fills out post field, detailing what they would like to post publicly in space</li><li>2. Click submit to create a publicly visible post</li></ol>
Alternative Paths	None
Postconditions	There is a Space in the data for the user to post in. Users are redirected to the new page
Acceptance Tests	Post is found in "Post Space" page and can comment/react on post

## Use Case #9

Name	View Post
Users	Faculty and Students
Rationale	Faculty and Students
Triggers	The user clicks on an existing post on a “space” page
Preconditions	The user is logged in
Actions	<ol style="list-style-type: none"><li>1. The user is on a “space” page</li><li>2. The user clicks a post, and the post reveals the content comments within that Post page</li></ol>
Alternative Paths	None
Postconditions	The post is now viewable, showing the post content, reactions and comments
Acceptance Tests	Post contents are able to be viewed

## Use Case #10

Name	Edit Student's Post
Users	Faculty. Only Faculty can Edit a Student's Post
Rationale	Faculty Users must be able to Edit any Student user's post from Post Spaces.
Triggers	Faculty users click the "Edit Post" button to modify the content within their post. Once the Post is edited, the users can click the "Submit Edit" button.
Preconditions	The Faculty user is logged in, is in the Post Space and there exists a post created by a Student user
Action	<ol style="list-style-type: none"><li>1. The Faculty user clicks on a post they would like to edit in a Post Space</li><li>2. The user clicks "Edit Post" button to edit the contents of a post</li><li>3. Once the post content is modified, the user clicks the "Submit Edit" button to repost a new post.</li></ol>
Alternative Paths	None
Postconditions	The new modified Post content is posted the Post Space
Acceptance Tests	Viewing an edited Post reflects the new changes made in the edit.

## Use Case #11

Name	Delete Student's Post
Users	Faculty. Only Faculty can Edit a Student's Post
Rationale	Faculty Users must be able to Delete any Student user's post from Post Spaces.
Triggers	Faculty users click the "Delete Post" button after clicking a post to remove any unwanted Student user posts from the Post Space.
Preconditions	The Faculty user is logged in, is in the Post Space and there exists a post created by a Student user.
Action	<ol style="list-style-type: none"><li>1. The Faculty user clicks on a post they would like to remove from a Post Space</li><li>2. The user clicks "Delete Post" button</li><li>3. Once the button is pressed, the post is removed from the database and from the Post Space page</li></ol>
Alternative Paths	None
Postconditions	The post is no longer available on the Post Space page
Acceptance Tests	The Post Space no longer contains the deleted Post.

## Use Case #12

Name	Comment on Post
Users	Faculty and Students
Rationale	Faculty and Students must be able to comment on a post on a Post Space page
Triggers	The user clicks on "Comment" button
Preconditions	The user is logged in and there exists a post on a Post Space page
Action	<ol style="list-style-type: none"><li>1. The user clicks on a post or "View Post" button</li><li>2. The user clicks on the "Comment" button and fills out the comment area box for the comment content to be posted</li><li>3. Users presses Enter button or "Submit Comment" to submit content to database and be posted on the Post page</li></ol>
Alternative Paths	None
Postconditions	The comment is visible on the post after clicking the "View Post" button
Acceptance Tests	Viewing a Post displays the comment made on it.

### Use Case #13

Name	Edit Student's Comment
Users	Faculty. Only Faculty can Edit a Student's Post
Rationale	Faculty Users must be able to Edit any Student user's comment from Post Spaces.
Triggers	Faculty users click the "Edit Comment" button to modify the content of their comment. Once the comment is edited, the users can click the "Submit Edit" button.
Preconditions	The Faculty users is logged in, is in a Post Space and there exists a comment on a post
Action	<ol style="list-style-type: none"><li>1. The user clicks on a post or "View Post" button</li><li>2. The users find a comment they would like to edit and click the "Edit Comment" button to modify the content within the comment area box</li><li>3. Once the edits are finalized, the users clicks on the "Submit Edit" button to submit the modified content to the database and to post the new modifications on the comment</li></ol>
Alternative Paths	None
Postconditions	The new modified Comment content is posted on the post
Acceptance Tests	Viewing an edited Comment reflects the new changes made in the edit.



#### Use Case #14

Name	Delete Student's Comments
Users	Faculty. Only Faculty can delete another Student's Post
Rationale	Faculty users must be able to delete and Student user's comment from Post Spaces
Triggers	Faculty users click the "Delete Comment" button to remove a
Preconditions	The Faculty user is logged in, is in the Post Space and there exists a comment on a post created by a Student user.
Action	<ol style="list-style-type: none"> <li>1. The user clicks on a post or "View Post" button</li> <li>2. The users find a comment they would like to remove and click the "Delete Comment" button</li> <li>3. Once the button is pressed, the comment is removed from the database and from the Post page</li> </ol>
Alternative Paths	None
Postconditions	The comment is removed
Acceptance Tests	The comment is no longer visible on the Post it was made under.

## Use Case #15

Name	Create Post Space
Users	Faculty.
Rationale	Faculty need to be able to create Post Spaces to organize the Posts together. Students should not be able to do this since they are unable to moderate the Post Space.
Triggers	Faculty user clicks the "Create Post Space" button from the left sidebar.
Preconditions	Faculty user is logged in.
Actions	<ol style="list-style-type: none"> <li>1. Faculty user fills out Post Space fields.</li> <li>2. The Post Space information is created and stored for later retrieval.</li> </ol>
Alternative Paths	If the new Post Space already exists, then don't add it to the database and display an appropriate error message on redirect back to the original page.
Postconditions	The new Post Space is now created. The user is redirected to the new Post Space page.
Acceptance Tests	Post Space is found in the database.

## Use Case #16

Name	Delete Post Space
Users	Faculty.
Rationale	Faculty users need to be able to delete a Post Space so the database does not get cluttered/disorganized.
Triggers	Faculty user clicks "Delete Post Space" button from either the "Individual Post Space" page or the "View all Post Spaces" page.
Preconditions	Faculty user is logged in and has joined the Post Space they intend to delete
Actions	1. Faculty user is prompted with "Are you sure you want to do this?" prompt.
Alternative Paths	If a Faculty user changes their mind when asked for reverification, they are redirected to the previous page they were on.
Postconditions	Post Space is deleted from the database. Faculty user is redirected to the previous page. If the previous page no longer exists, the Faculty user is redirected to home.
Acceptance Tests	Post space is no longer able to be found in the database.

## Use Case #17

Name	Add User to Post Space
Users	Faculty.
Rationale	Faculty users need to invite other users to a Post Space so groups can be formed.
Triggers	Faculty user clicks the “Invite” button from the “Individual Post Space” page.
Preconditions	Faculty user must be logged in and a part of the Post Space they are trying to invite users to.
Actions	<ol style="list-style-type: none"><li>1. Faculty user is prompted with a search bar meant for ID or username to enter.</li><li>2. “Add” button displays next to each search result that is not a part of the Post Space already.</li><li>3. Faculty user closes the prompt when they are done inviting users.</li></ol>
Alternative Paths	None.
Postconditions	Users are added to the Post Space. Faculty user is redirected to the previous page.
Acceptance Tests	User is able to see the Post Space and appears on the Post Space user list.

## Use Case #18

Name	Remove Student from Post Space
Users	Faculty.
Rationale	Faculty users must be able to remove unwanted Student users from Post Spaces.
Triggers	Faculty user clicks the “Remove Student From Post Space” option under the “Post Space Settings” button.
Preconditions	Faculty user is logged in, is in the Post Space and the Student user they intend to remove is also in the Post Space.
Actions	<ol style="list-style-type: none"><li>1. Faculty user is displayed every Student user in the Post Space with a “Remove” button next to each of their names.</li><li>2. Faculty user closes the prompt when they are done removing users.</li></ol>
Alternative Paths	None.
Postconditions	Student Users are removed from Post Space. Faculty user is redirected to the previous page.
Acceptance Tests	User is no longer able to see the Post Space and does not appear on the Post Space user list.

## Use Case #19

Name	Join Discord Voice Channel
Users	Faculty, Students.
Rationale	Having Voice Channels allows for online group meetings. Users must be able to join these Voice Channels from the application.
Triggers	User clicks the “Join Voice Channel” button from the “Individual Post Space” page.
Preconditions	User must be logged in and in the Post Space of the Voice Channel they are trying to join.
Actions	<ol style="list-style-type: none"><li>1. User is redirected to Discord.</li><li>2. User joins the dedicated Discord Voice Channel for that Post Space.</li><li>3. User disconnects from Voice Channel.</li></ol>
Alternative Paths	Something is wrong with opening Discord (logging in, opening the app, etc.) and the user is redirected to the previous page.
Postconditions	User is redirected to the previous page.
Acceptance Tests	User is successfully redirected to the Discord Voice Channel.

## Use Case #20

Name	Request to join a Post Space
Users	Faculty, Students. Faculty do not need authorization to join a Post Space
Rationale	It would be convenient to the users if they could search through all the visible Post Spaces to join instead of relying on invites only.
Triggers	User clicks the "Request to Join" button next to the desired Post Space in the "View All Post Spaces" page.
Preconditions	User is logged in and not already a part of the Post Space they are attempting to request to join.
Actions	<ol style="list-style-type: none"><li>1. Request is sent to the Faculty users within a Post Space.</li><li>2. Faculty user is prompted with a notification to accept or deny the request.</li></ol>
Alternative Paths	Request is denied and now the user can only be added by a Faculty user to a Post Space.
Postconditions	Request is accepted and the user is added to the Post Space.
Acceptance Tests	The request is received by the Faculty users of the Post Space. Once the request is approved, the User is able to see the Post Space and appears on the Post Space user list.

## **II.2. Functional Requirements**

### **II.2.1. User Authentication and Permissions**

**Authentication:** Student and Faculty identities need to be verified before the creation of both faculty and student accounts. Users should have student or faculty ID numbers to which their accounts will correspond. Faculty accounts with verified identities will be able to create student accounts and have moderative abilities.

**Source:** Team Toto development team originated this requirement. The requirement is necessary for the security of the application.

**Priority:** Priority Level 0: Essential and required functionality.

### **II.2.2. User Records**

**Title:** The databases created during development will be responsible for storing all user, space, post, comment, and other data. Users need to be able to view a list of user records. Faculty need to be able to create these functions, have access to them, and allow access for students to access them.

**Source:** Team Toto development team originated this requirement. The requirement is necessary for the development of the application, and for application usage.

**Priority:** Priority Level 0: Essential and required functionality.

### **II.2.3. Administrative Moderation**

**Comment and Post Review:** Faculty accounts with verified identities will be able to moderate student activity through the editing and deleting of student activity. Additionally, Faculty users will be able to change the permissions of Student users within each Post Space to allow/disallow certain actions. This includes student posts, comments, profile information, and other communications.

**Source:** The client originated this requirement. The requirement is necessary for faculty users and CIPA Compliance.

**Priority:** Priority Level 0: Essential and required functionality.



#### II.2.4. Posts

**Post Creation and Usage:** Posts can be created by all users given this permission in a Post Space. Each Post also has an individual settings page that can allow/disallow comments and pin the Post within the Post Space. All users are able to edit, delete and change the settings of their own posts, while Faculty users can edit, delete and change the settings of the Posts made by Student Users.

**Source:** The client originated this requirement. The requirement is necessary for communication within the application.

**Priority:** Priority Level 0: Essential and required functionality.

#### II.2.5. Post Spaces

**Post Space Creation and Usage:** Post Spaces are the text channels in which Posts are created and displayed to the rest of the users within that Post Space. Faculty users are the only ones who can create Post Spaces. After creation, Faculty users can add other users to the Post Space and manage their permissions to Post and Comment within the Post Space. Other managerial duties include the ability to remove users, delete the Post Space, hide the Post Space from the search.

**Source:** The client originated this requirement. The requirement is necessary for communication within the application.

**Priority:** Priority Level 0: Essential and required functionality.

#### II.2.6. Training Materials

**Training for Students and Teachers:** The application must have training materials to train students, faculty, and administrators on how to use the application. Since this application will be unfamiliar to all users, provided training materials must clearly demonstrate how to use the application. The application is focused on usage by students in high school and younger, and by Faculty. The student training materials must be easy to read and learn how to do.

**Source:** The client originated this requirement. The requirement is necessary for both student and faculty users.

**Priority:** Priority Level 0: Essential and required functionality

### II.2.7. Commenting

**Comment Creation and Usage:** Comments can be created in a thread underneath Posts by all users given this permission in a Post Space. Users can edit and delete their own comments, while Faculty users can edit and delete the comments of other Students.

**Source:** Team Toto development team originated this requirement. The requirement is necessary for communication within the application.

**Priority:** Priority Level 0: Essential and required functionality.

### II.2.8. Discord Voice Communication

**Discord Voice Communication External Function:** Whenever a Post Space is created, a Discord Voice Channel is automatically created as well. Users can join this Voice Channel, externally directing them to the Discord Application where they are able to vocally communicate with other users.

**Source:** Team Toto development team originated this requirement. The requirement is desirable for the expanded usage of the application from a communication standpoint.

**Priority:** Priority Level 1: Desirable functionality.

### II.2.9. Private Messaging

**Private Messaging between users:** Users are able to privately message other users, setting up a Text message-like conversation.

**Source:** Team Toto development team originated this requirement. The requirement is desirable for communication between Users.

**Priority:** Priority Level 1: Desirable functionality.

### II.2.10. Searching Post Spaces

**Searching:** Users are able to search (by keyword) for Post Spaces. They are then able to request to join a Post Space, where a Faculty user can either accept or deny them.

**Source:** Team Toto development team originated this requirement. The requirement is desirable for ease of use.

**Priority:** Priority Level 1: Desirable functionality.

### **II.2.11. Home Post Space**

**Personalized Home Post Space:** Users are able to select a Home Post Space that they are greeted with upon application startup.

**Source:** Team Toto development team originated this requirement. The requirement is an extra feature for the development and usage of the application.

**Priority:** Priority Level 2: Extra features or stretch goals.

### **II.2.12. Post Space Organization**

**Organization:** Users can organize the Post Spaces they are a part of into folders.

**Source:** Team Toto development team originated this requirement. The requirement is an extra feature for the development and usage of the application.

**Priority:** Priority Level 2: Extra features or stretch goals.

## **II.3. Non-Functional Requirements**

### **II.3.1. Easy to Use**

The application will be used by both students and faculty, who will be new to the software. It must be easy to use and accessible to all in order for users to actually want to use the application. If teachers struggle to use the application's features, its uses are useless.

### **II.3.2. Maintainable**

When the project is completed and delivered to the client, the school districts will become the owners and maintainers of the product. Thus, this application should be easy for them to maintain.

### **II.3.3. Visually Appealing**

As the application is focused on usage by late elementary-age to high school-age students, the application must be visually appealing enough to keep the attention of students and get them to want to use it.

### **II.3.4. Quick Response Time**

The server should respond to page accesses in a reasonable time. On a fast connection, 3s response time is reasonable.

### **II.3.5. Cross-Platform Availability**

The application would work from platform to platform. Students utilize Chromebooks through school checkouts, but will also utilize applications on mobile devices. The application should run and look consistent regardless of platform.

### **II.3.6. Security**

The application must be secure in regard to application security and user integrity. The accounts of users shall stay secure, and a password change shall be required every 90 days. User accounts shall be verified before being imputed into the database.

### **II.3.7. CIPA Compliant**

The application shall follow all required guidelines and stay compliant with the required CIPA guidelines.

## **III. System Evolution**

The main fundamental assumption we are making is the fact that these requirements are all CIPA-compliant. Functionality such as Posts and Commenting should be safe since Faculty users will be able to edit and delete inappropriate content. However, functionality such as Private Messaging and especially Discord Voice Communication might need to change over time to adjust to CIPA.

When it comes to Discord Voice Communication, several ideas have been discussed to comply with CIPA. The Voice Channels will likely need to be private, meaning the only way to join them is through the Post Space. This way, users outside of the Post Space will be unable to join. If all voice calls need to be monitored, bots can be created that record the audio conversations. This could potentially lead to more issues however, and needs future discussion. Discord allows for private messaging between accounts that is unmonitored. Disabling this functionality is an obvious option, but may not be achievable. If our team finds out that using Discord is not a realistic option, then an alternative plan will be put in place. Using a different platform (Google Voice, Telegram) and creating our own voice channels are all options up for consideration. Overall, our team believes the benefits of using Discord outweigh the risks it holds and we will move forward with it until further discussions and research has been made.

## IV. Glossary

**Administrative Moderation** - Faculty users with administrative accounts have extra permissions that student users do not, such as editing and deleting other user's posts. These accounts also have permissions to moderate the behavior of student users.

**Authentication** - A form of validating a user's ID and password are within the database and correct. Once validated, users can log in.

**Back-End** - The server side; Everything necessary for an application that users do not see, including code/scripting, databases, software architecture, security, etc.

**Cross-platform** - Software that is designed to work on several computing platforms. In this application, software must work on both Chromebooks and mobile devices.

**Chromebooks** - Runs using Chrome OS. The main platform students use and the main platform for this project.

**CIPA Compliance** - The Children's Internet Protection Act our application must comply with.

**Database** - An organized collection of structured information, or data, typically stored electronically in a computer system.

**Discord Voice Communication** - A very popular VoIP (Voice over Internet Protocol) and instant messaging social platform that the school previously used.

**Front-End** - The client-side; everything that users actually see on an application, including the web pages, performance, UI, etc.

**Full-Stack Application** - The entire depth of an application, both front-end and back-end.

**Software Architecture** - The fundamental structures of a software system and the discipline of creating such structures and systems

**Stakeholders** - Individuals, groups, or organizations directly involved with, or indirectly affected by, a project, product, service, or enterprise.

**UI (User Interface)** - The space where interactions between humans and applications occur; all the physical buttons, text boxes, etc. that users interact with within an application.

**User** - Any entity that has an account on the application. Users will have specific permissions depending if they are a faculty or student.

**User Records** - Data of user's posts, comments, space and profile. These are all stored on a database used for functions throughout the application.

## V. References

"Children's internet protection act (CIPA)," *Federal Communications Commission*, 28-Apr-2020. [Online]. Available: <https://www.fcc.gov/consumers/guides/childrens-internet-protection-act>. [Accessed: 04-Oct-2022].

"ChromiumOS Developer Guide," *ChromiumOS Docs - ChromiumOS Developer Guide*. [Online]. Available: [https://chromium.googlesource.com/chromiumos/docs/+HEAD/developer\\_guide.md](https://chromium.googlesource.com/chromiumos/docs/+HEAD/developer_guide.md). [Accessed: 04-Oct-2022].

"IEEE standards," *IEEE*. [Online]. Available: <https://www.ieee.org/content/ieee-org/en/standards/index.html/>. [Accessed: 03-Oct-2022].

S. Berkun, *Making things happen: Mastering project management*. O'Reilly Media, 2015.