# VOX

###### Realtime Group Chat App

Document Version V2

## Software Development Plan

Names: Gabe Leffew, Hamad Ahmed, Ben Lloyd

| Date Created | Version # |
| --- | --- |
| 12/8/2022 | 3 |
| 10/7/2022 | 2 |
| 9/23/2022 | 1 |

## Product Description

Our goal is to create a centralized location for collaboration in the classroom setting and beyond. Schedule events for your team and find the best team to tackle your toughest goals. Save notes for classes and share with your group to always be on the same page. Manage your groups and create a hierarchy as you see fit, whether you want one to rule them all or total anarchy. Create servers for any group like a club, a group of friends, or your tri-annual family reunion.

## Team Description

| Skills | Hamad | Gabe | Ben |
| --- | --- | --- | --- |
| Git | X | X | X |
| Frontend | X | X |  |
| Backend |  |  | X |
| UI |  |  | X |
| Database (SQL) | X | X | X |
| HTML/CSS | X | X |  |
| Web server | X | X | X |

### Skills required:

* General systems design
* Time management (estimates, finishing work in a timely fashion)

## Software Process Model Description : AGILE

The process model we are using for this project will be Agile. The first reason we chose Agile was that it works well with other systems. An example of this would be that we want to use a kanban board which will integrate with our sprint cycles. This in turn will help us keep up with tasks and know what to prioritize working on.

Another reason for using agile is that it is a flexible model that will enable us to make rapid changes throughout our development process. It allows us to react quickly as we implement the project and we learn more about what we are making. Through its duration there will be design challenges that we will need to adapt to and Agile will help mitigate those.

Agile is the de facto standard in the software industry so we chose this methodology so we can get more comfortable with the process. Learning about the agile methodology in a forgiving environment such as the classroom will help us become accustomed to the process and avoid common mistakes. We hope that by using Agile we can more closely create the professional quality for delivering software.

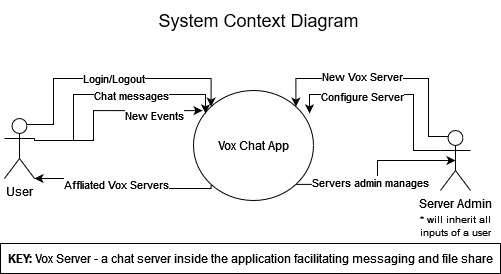
## Product Definition

### Actors

User - Any person using the application that does not own a server inside the application. This is not the same as a user inside a server as that is a per sever basis.

Server Admin - Any person that owns a server inside the application. They have more privileges for a specific server but not the application as a whole.

### Context Diagram



Context Diagram Image

### Personas

I am a college student who always forgets my due dates and can’t communicate effectively with my team members. If only there was one spot to see all this information, I would excel in my academic career.

I am a football coach and can’t keep up with my rowdy team. Some of my dogs forget to come to practice sometimes and other times they don’t hand in their chicken barbecue fundraisers. I need a place to collaborate with my team and help my boys succeed.

### Use Cases



Use case diagram image

## Wireframes

See /Design Files/WireFrame for wire frames

## Project Organization

### Responsibility Matrix

| Skills | Hamad | Gabe | Ben |
| --- | --- | --- | --- |
| Git | X | X | X |
| Frontend | X | X |  |
| Backend |  |  | X |
| UI |  |  | X |
| Database (SQL) | X | X | X |
| HTML/CSS | X | X |  |
| Web server | X | X | X |

## Validation Plan

The definition of done can be described as completing all use case conditions to allow users to communicate effectively while using features within the app. Success from our point of view is to have a functioning app where two users can communicate whether it be by sharing files, chatting, or calling.

## Feasibility Study

### Risk Identification

With being an online application, we need to make sure that we have security for users’ personal data. Whether it be their login credentials or communications. Since chats and files exchanged between users will be saved, there will be a lot of personal data, and attacks on our database without a secure system are a big risk. Another risk is the lack of knowledge for creating our application, this is our first time creating an application of this scale, as well as our first time working with most of these technologies. So, when it comes time to create some of these features, we could find ourselves spending more time learning rather than developing a feature. This in turn may limit some of the features and capabilities of our application. Since we’re planning on having so many features within our application, there is more room for bugs. If we have bugs and problems in our program it could question the reliability of the application. It may also pose difficulty when trying to implement new features into the application.

### Risk Prioritization

Lack of knowledge Reliability Security

### Risk Mitigation

A way we can keep our knowledge up to date is to always keep learning about how to implement features within our application. For security, we can implement hash functions to encrypt the user’s personal data. Reliability can be held to a high standard by implementing code checks while creating the application to review and change code to be most reliable before finish. Security will not be of high priority in this project and instead we plan to focus on the system design.

## Configuration and Version Control

* Our application will utilize semantic versioning MAJOR.MINOR.PATCH
  + MAJOR - Major changes, Incompatible API changes
  + MINOR - Minor features
  + PATCH - Bug fixes
* Our documents will be versioned by the semantic MAJOR
  + MAJOR - changes after group reflection of documents
  + The design document will also be tracked by Git from V2 onwards

## Tools

* Programming Languages: Frontend language, backend language
* VS Code
* Database Management System: SQL
* Web Server

## Architecture

* Hardware:
  + System to host the server
  + Active internet connection
* Software:
  + Browser to use the application
* Research:
  + Deploying to a web server