

Team No. 5

Team Members: Junpeng Cao, Blake Rieschick, Bohui Lu, Josh High, Yizhou Wu

Project Name

Let'sTalk

Project Synopsis

A social platform that utilizes machine learning identifying individual user's interest, and matches users with voice chatting in real time.

Project Description

The project is being overtaken because all of our team members are interested in different kinds of social software and we are passionate to implement some creative features to help people to make more friends in their pastime. In nowadays' society, people have less time to make friends, and people want to keep everything in a short period of time. Popular application likes Tik' Tok is so successful because it focuses on the shortness of videos, and people can watch it anytime in any occasions. Our project's goal is to take the concept of 'shortness' from nowadays' application, and provides a platform for people to make friends around the world in a short period of time by voice chatting.

Machine learning model would be implemented into the application for identifying user's browsing history on the application and individual interests and habits, and those data would be saved and updated on a regular basis for matching people that share the same interests with voice chatting.

The application allows users to post "moment" like any other social media application, and their moments will be constantly pushed to those who shares the same interests. The voice chat is only available during matching period, and message chatting is only available for those that like their match. During registration of an account, user needs to complete their profile by answering short questions to help the application better identify who they are and filling up information such as he/she's tag, and nickname.

The main problem that we might have is the backend part because we don't have any experience in this field. We have to spend time to learn and practice machine learning and integrating different servers. Hopefully all the features described above could be implemented and easy to maintain.

Project Milestones

2019 Fall :

- (4 weeks)
- Project requirements defined. Ideas' discussion and exchanged.
- Github set up. Designs layout decided.
- Front-end design. Online server research.
- Prototype design, database scheme
- Virtual Environment created for Project
- Database connected with application

- (4 weeks)
- Initial front-end coding/researching voice input online server
- Initial back-end coding
- back-end coding
- back-end coding

2020 Spring:

- (4 weeks)
- back-end coding/database installed
- front-ends styled,
- UI design

- (4 weeks)
- Front-end/Back-end connected
- Bugs checking
- Project test
- Project release

Project Budget

- Need to rent a cloud server for data storage, it will be free since the size of information needed to be stored will not be big at the beginning.
- Training on using database and coding for the backend is required since most of the teammates are not familiar with it.

Work Plan

Junpeng Cao:

2019 Fall:

- Finish the prototype of this project.
- Front-end design for the User page.
- UI implementation for User page.

2020 Spring:

- Learn how to connect the front-end part to the server.
- User page function implementation.

Blake Rieschick:

2019 Fall:

- Download "Let's Talk" repo and configured environment to support app.
- Work on back-end implementations of the project including implementing mysql database and necessary framework to connect with front-end.
- Implement user accounts on database.
- Implement retention of user data

2020 Spring:

1. Add user configurations to save video, text, talk preferences
2. Connect call logs, text logs, video call logs within user accounts across app services

Bohui Lu:

2019 Fall:

- Front-end design for Log-In page
- Code that can verify correct user's input or correct reaction to user's inputs.
- Help to make all the front-end pages as dynamic as possible
- Back-end research such as online server and database integration with web application
- Testing for the web application in different scales

2020 Spring:

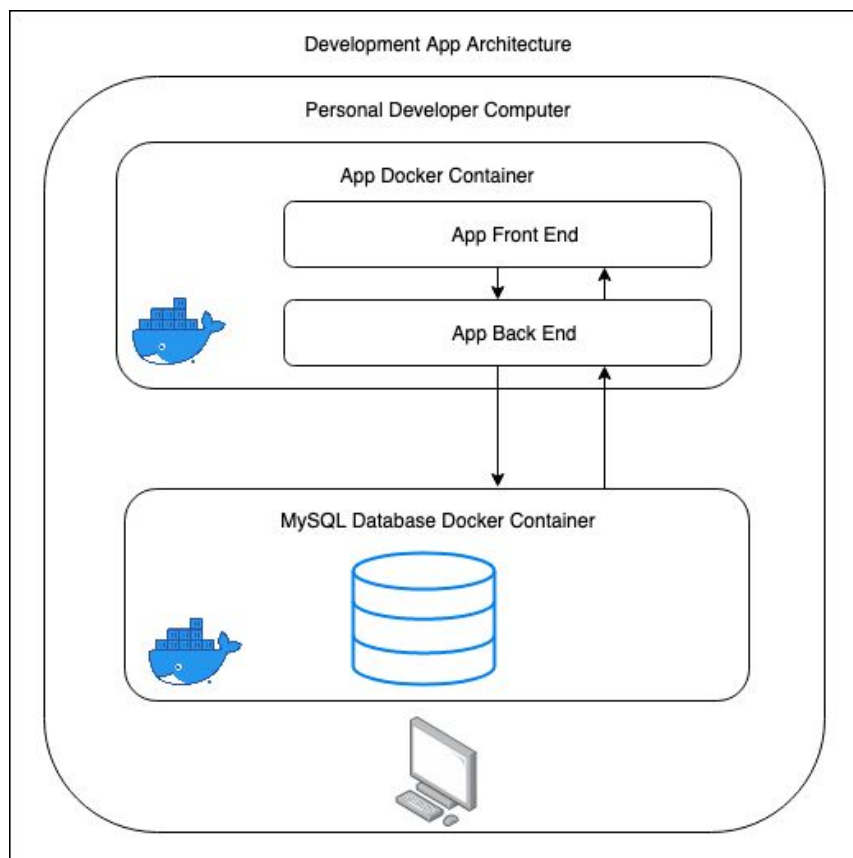
- Multi-users implementation of the web application
- Security of database

Preliminary Project Design

The primary goal of the application is to provide a platform for people that just want to have someone to talk to from around the world to make friends. The application will be compatible with both Android and ios. The application mainly features and promotes identifying a new friend with his/her voice, and voice is a unique way of identifying different individuals.

Back End Architecture

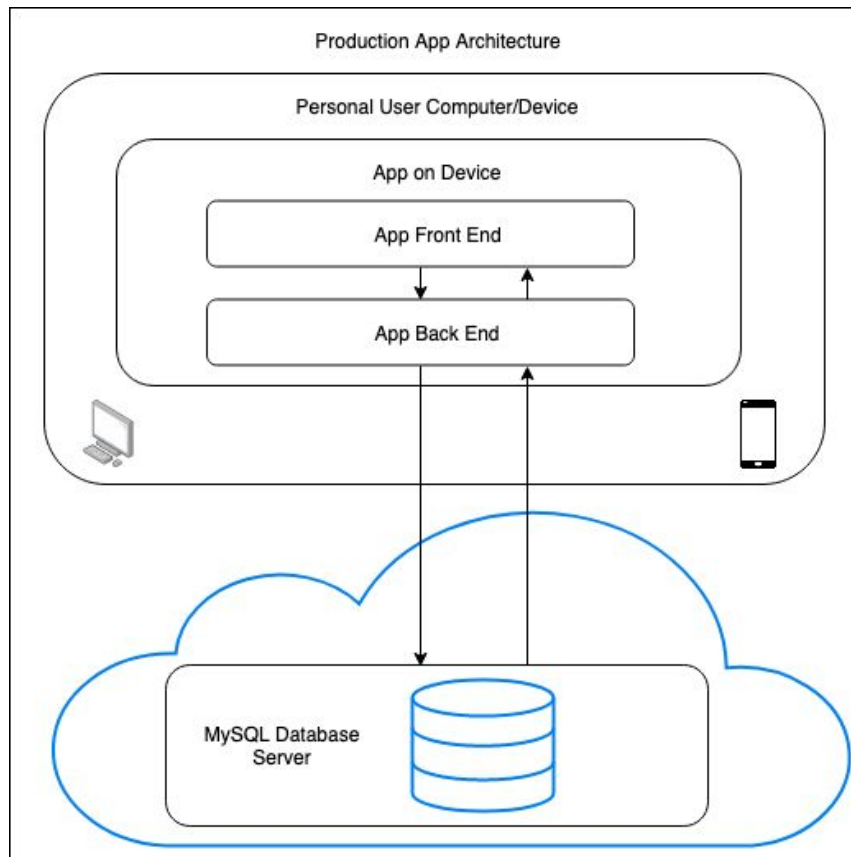
Development Architecture



For development, which will consist of the primary environment for which the development team will program within, will consist of two docker containers working on the developers personal machine that work together to achieve the full app experience. The first container will run the app in a docker container as if it was on a server with viewJS. Then the second container will hold the MySQL database and be used to store the app information. As the instance of the MySQL database will be unique on each users computer the data contained therein will also be unique but once the app shifts to production the MySQL database would

be migrated to a server where it can take requests across either an intranet or the internet given we decide to expose it publicly on the web.

Production Architecture



In terms of using the production app architecture this is when the app begins migration from development to production. In the case of production level architecture the user would download the app to their device and then use the apps front end user interface to communicate with the MySQL database that is stored on a remote server in the could either privately or publicly.

MySQL Database Interaction

The MySQL database is what the app will use as a means of retaining user data and associate users with posts, and relate those posts among users. Currently the design of the entity relationship diagram is still being produced but the following actions have been considered a part of the application that relies on various functionality of the MySQL database to interact with the users data.

- Post Creation
 - InsertPost

- InsertAnonPost
- Post Interaction
 - IsFav
 - IsHeart
 - IsLike
 - CommentPost
 - Find Post FilterByRegion
 - Find Post FilterByAnon
 - DeletePost
- Profile Settings
 - ChangeName
 - AddLikes
 - ChangePassword/ResetPassword

Posting

The user will interact with the front end of the app to create a post. When a post is created there are two options that exist for that post, as it can either be identified by a user, or posted anonymously. If the user chooses to create a post with it associated against their account then a new post entity will be created within the MySQL database and associated with a user that posted it. If the post created by the user was chosen to be anonymous then the post entity will still be created but not associated with any user.

Post Interaction

Interacting with a given post can include many simple features that are automatically presented with the post on the front end. This can include favoriting, liking, or hearting a given post by a given user. If any one of these actions occur, the post entity's attribute containing a counter of the number of interactions with the post within the MySQL database is incremented. Thus the database will show how much interaction exists between users and the post. Moreover, the like entity will bind the user that liked it to the post they like such that they can never interact a certain way with a post more than once.

Another feature of post interactions, includes the ability to search for a given post based on some criteria to help filter the search down. Thus far, we have discussed using both region, and if anonymous posting is implemented then filtering by that would work well too. Searching is a crucial part of the app as it lets a given user go out and search for content they wish to obtain.

Additional features of post interaction include commenting. Like most chat applications that involve users communicating with one another, it is important to be able to interact in ways

outside of like and hearts and such, but to include comments that allow communication across a variety of languages and users to one another.

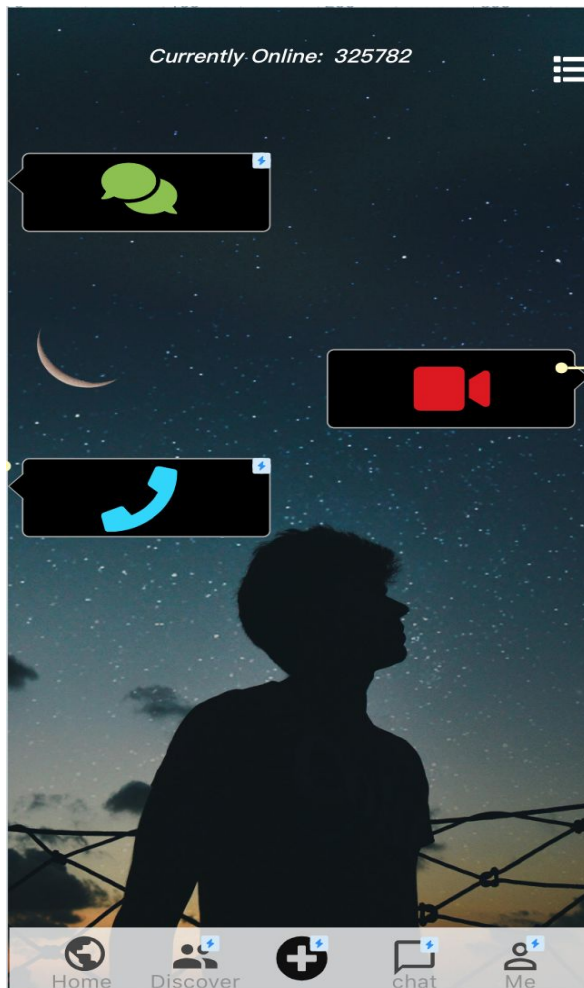
Then lastly, with regard to post interaction, users that create a given post will have the ability to delete their posts. Only the user that creates a post can have the ability to take it down such that other users don't have the power to alter and interact with posts not belonging to them.

Profile Settings

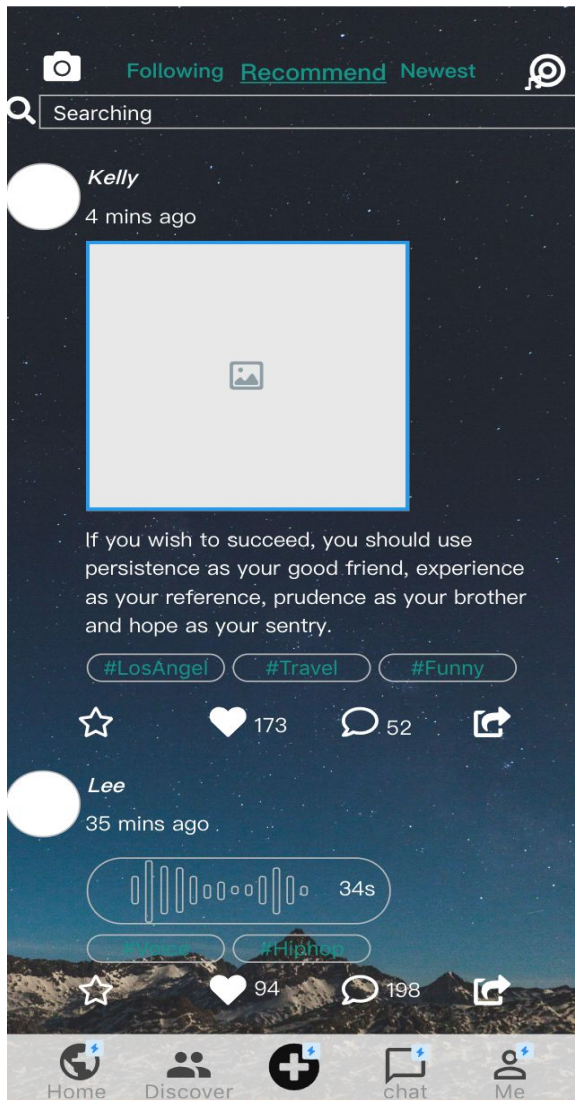
Given that a user of our application is signed up and registered within the MySQL database, they will have the option to change the details associated with their account. Such actions include the user being able to change their first and last name. Add content about themselves such as likes and dislikes. Then finally they can change or reset their password if the situation arises where they might need to.

Frontend and UI:

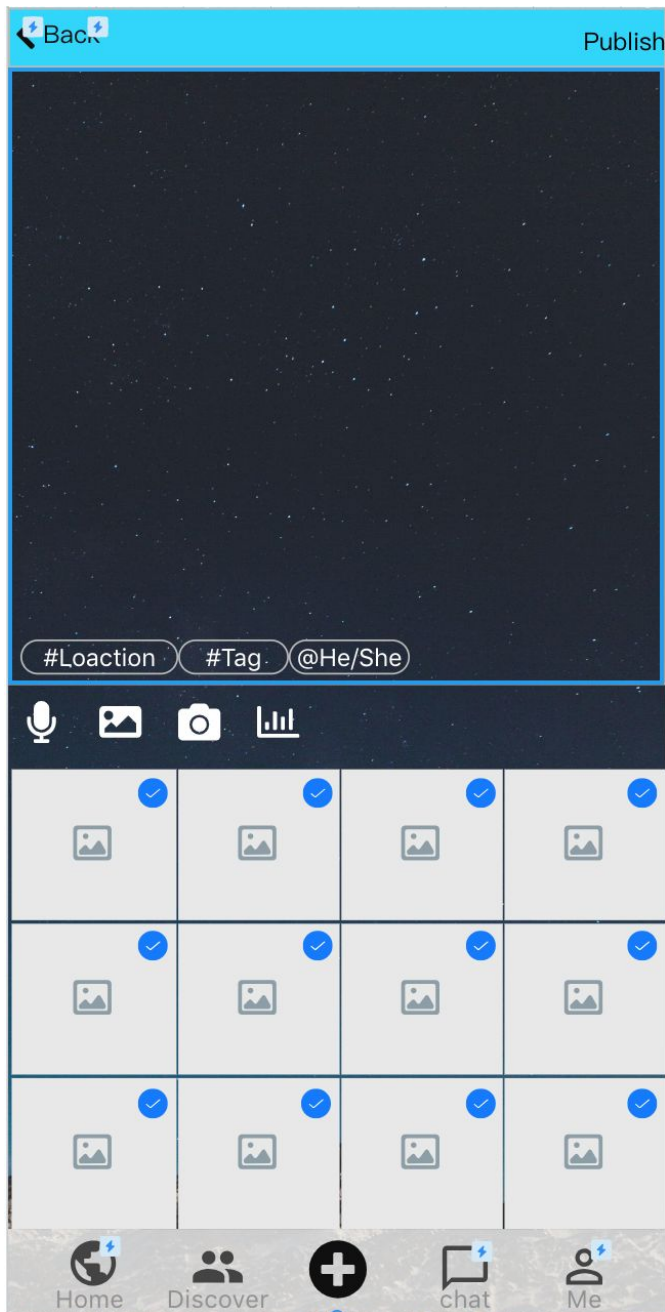
This app will be written using Javascript framework "Vue.js" and it has five main features (Star(Homepage),Square(Moment from other users),Chat,Users page and Post).



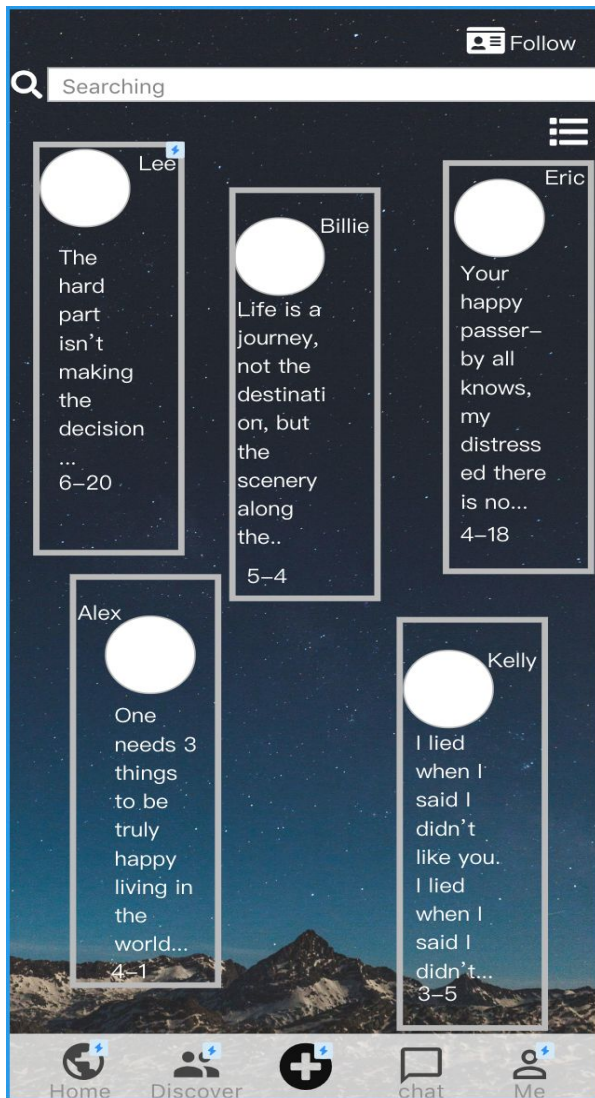
Star(Homepage): It has three buttons includes (message,voice chat and video chat). The chat matching is based on the tags that users created. For example, X has tag "student" will match with Y who has tag "student". Actually, we want to let the users to build their own background picture and background music in home page. However, there are too many features that we need to implement and we will add those two feature if we have extra time after finishing the whole project.



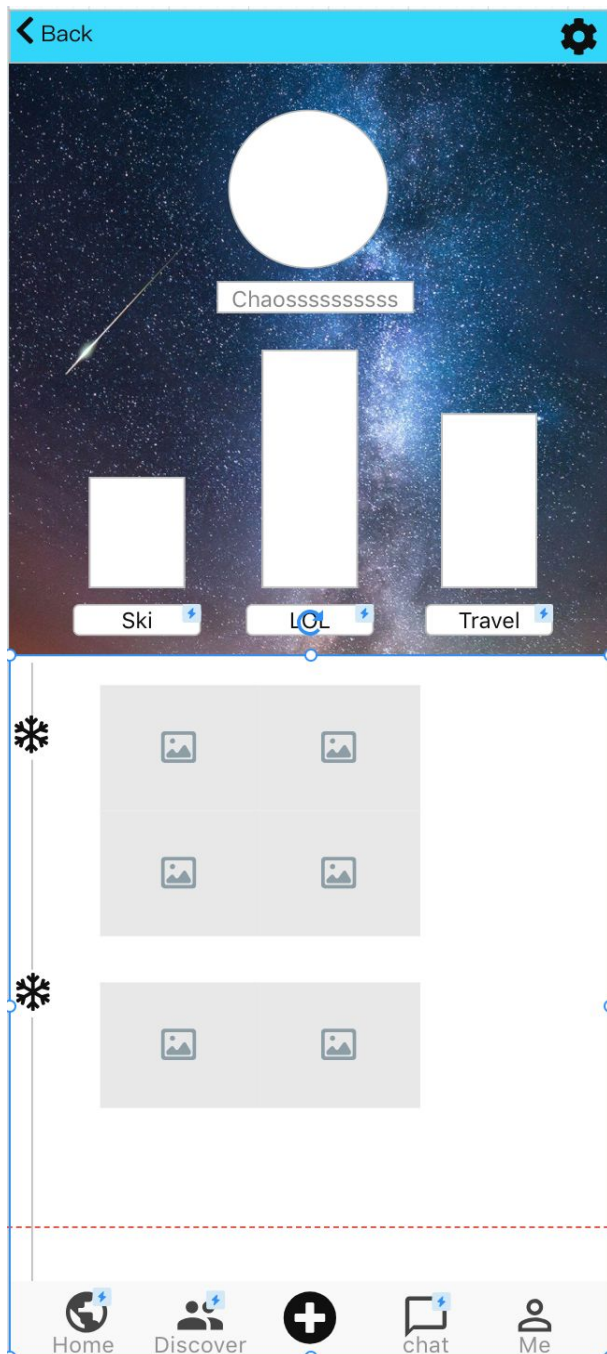
Square(Moment):Users go to the “Square” to browse his/her friends’ post.Each post is a separate blocks with user’s icon and posting time on the top left corner, main body at the middle, tags below the body and interaction functions at the bottom. Every user can share their daily life in the square by clicking the camera icon on the top left corner of the screen to post voice messages, videos , photos or just plain text. Customized tag can be added to these posts for future topic searches conducted by other users. Every user can click the star to add the post into their “favourite” for future use. They can also click the heart to like the post or make comments under the post. The post can also be shared with other users.



Post: Pictures, recorded voice segment, and text are supported in the individual's post. Since the application mainly features voice chat, unlike other social media platform, it allows users to post their voices. It encourages people to speak out and share their stories through voices. Those users being identified shared the same interests would receive pushing contents of each other(stories), and the pushing contents' feature could be done by using machine learning model.



Chat: For the chat page, users can find the lists of their friends by clicking the “Follow” icon on the top right corner. All the chats users had will be listed on this page in individual blocks, with the friend’s icon on the top followed by the last message the conversation had. Clicking on the block will jump into the specific window that allows user to chat with friends. We tried to develop a novel way to list the chats rather than use the formal way by listing the chats in a vertical line. If the users have many chats in this window and find it difficult to find out one specific chat, they can use the search bar to find the chat directly.



Userpage: The user has to create some tags when they create a new account. For this feature, it would show the top 3 tags with it's score, the score is based on the feedback from other users that every time the matching complete. Also, it would show the name, picture or voice that the user creates. Users can also rename their nickname by clicking "setting" button on the top right corner.

Ethical Issues

Security:

There is no doubt that we should take good care of the users' data including their usernames, passwords, email address or phone number. Definitely, a protective mechanism should be applied to prevent hackers to steal the user information. Additionally, it is not common that scams can happen in social media applications. We will notice the users when the conversation began to involve money issues. It is also necessary to provide the users an effective method to report the users managed to do scams. Once the report received, we will freeze or cancel the account of the suspect depending on the situation.

Content/IP:

The application exists potential of being misused by some of the users. Sensitive and illegal contents can possibly be posted through pictures, voice, and text. Users could also use the application as a platform for false advertisement. One of the biggest issues the application has to encounter is that it's hard to prevent some of the users to make improper remark during a match since two users are communicating through voice. Users could be offended and harassed by those who try to misuse the application. The application's major goal it to match and encourage people that share the same interests to make friends, but sometimes it is hard to identify someone actually shares the same interest with others by just tags and having them to answer some designated question. The system might not be able to match people precisely.

Privacy:

For this part, we promise that we will not "monitor" the user includes theirs voice and chat message data. Also, users information includes phone numbers or payment method(for the future payment feature) is very important for us and we will not sell them to the third party. Password is the key to protect user privacy. Therefore, we will remind user to update their password once a week. In addition, we will give some privacy tips to our users and let them know how to protect their privacy.

Change Log

Task	Changed
Project Description	No
Project Milestones/Gantt Chart	No
Project Budget	No

