MILESTONE 2

Date: 04/09/2020



Team Lead : Cassie Sherman cassiesherman630@gmail.com

Backend Lead: Ryan Hu 213edu@gmail.com

Frontend Lead : Mariam Shahab mariamshahab2@gmail.com

Database Lead : Dylan Luong dluong@mail.sfsu.edu
Github Master : Sawara Bhattarai sbhattarai@mail.sfsu.edu

Documentation

Lead : Arjun Bista arjunbista428@gmail.com

History Table

| Milestones/Version | Date |
|--------------------|------------|
| M2 V2 | 04/09/2020 |
| M2 V1 | 03/36/2020 |
| M1 V2 | 03/09/2020 |
| M1 V1 | 03/05/2020 |
| | |
| | |
| | |
| | |
| | |
| | |

Table of Contents

| Data Definitions V2 | 2 |
|---|----------------------------|
| Functional Requirements V2 Unregistered User Registered-User Priority 1: Priority 2: Priority 3: | 4 4 4 5 7 |
| UI Mockups and Storyboards | 8 |
| High level database architecture and organization Business Rules Description of Entities, Attributes and Relationships Entity Relationship Diagram Database Model | 11 11 11 15 16 |
| High Level APIs and Main Algorithms POST: GET: WS: | 17 17 17 18 |
| High Level UML Diagrams | 19 |
| High Level Application Network and Deployment Diagrams Application Network Diagram Deployment Diagram | 20 20 22 |
| Identify actual key risks for your project at this time Schedule Risk Skills Risk Teamwork Risk Legal/Content Risk | 23 23 23 23 23 |
| Project management | 24 |

Data Definitions V2

- Unregistered User: These are users that have not registered an account yet. They will not have much privilege within our website. They will be able to see the create account page of LingoMingo only.
- 2. Registered User: A client or Premium client who is registered within the database with proper information and necessary documentations and has access to different functionality depending on their privilege within the application.
 - a. Free: A registered user who has free access to all the site's features. They will see ads on their dashboard.
 - b. Premium: A registered user who has access to all the sites features. Ads no longer are displayed on the registered user's UI.
- 3. Match: Users will have matches with other users. They will be matched based on language commonalities. Matches will be displayed in swiping format on a dashboard. Users can swipe right for matching with a matched user and left for unmatching with a user.
- 4. Chat: When a user has matched with another user they have the option to create a chat with that person
 - a. Text: Text chat box where the users can converse to practice written language.
 - b. Voice: Voice chat to practice spoken language without camera.
 - c. Video chat: Both voice and video chat to practice spoken language.
- 5. Rate: Users will be able to rate other users after they chat together via tags. These tags will be precreated and can be chosen by a user to describe other user's performance. These tags will be displayed on the user's profile.
- 6. Profile: This will be where Users talk about their interests and hobbies to let other users know a little bit more to get to know them. Parts of the profile can be private including age, location, and gender.
- 7. Post: People can post status comments or pictures on their account.
- 8. Language: In order to communicate with another user, the users need to know one language in common. We will set English as a common language.
- 9. Message: One user can send a voice or text message to the other user once they are friends.

- 10. Chat rooms: Page where users can interact with other users. Many users can be in one room.
- 11. Notifications: Display or notify the latest activities related to the LingoMingo.
- 12. Host: Registered users can create public or private chat rooms.
- 13. Support: A tab to solve the user's issues and answer their queries.

14. Report:

- a. User: User writing a report to send to an admin about another user if any misconduct or policy violation happened during chat conversations or private messaging.
- b. Post: User writing a report to send to an admin about a post on another user's profile page that is inappropriate or violates policies.

Functional Requirements V2

Unregistered User

Priority 1:

- 1. Unregistered users shall be able to sign up to account
 - 1.1 Unregistered users shall enter a name.
 - 1.2 Unregistered users shall enter date of birth.
 - 1.3 Unregistered users shall enter a username.
 - 1.4 Unregistered users shall enter a password.
 - 1.5 Unregistered users shall confirm a password.
 - 1.6 Unregistered users shall enter an email.
 - 1.7 Unregistered users should enter a city.

Registered-User

Priority 1:

- 2. Registered users shall be able to log in to account.
 - 2.1 Registered users shall enter an email or an username.
 - 2.2 Registered users shall enter a password.
- 3. Registered users shall be able to log out to account.
 - 3.1 Registered users have to be logged in before being able to log out.
- 5. Registered users shall be able to submit data.
 - 5.1 Registered users shall be able to submit images in the chat.
 - 5.2 Registered users shall be able to submit text in the chat.
 - 5.3 Registered users shall be able to make new posts.
 - 5.4 Registered users shall be able to edit posts.
 - 5.5 Registered users shall be able to delete posts.
 - 5.6 Registered users shall be able to post unlimited.
- 7. Registered users shall be able to delete the account.
 - 7.1 Register user's profile shall also be deleted.
 - 7.2 Register users shall be able to temporarily deactivate accounts.
 - 7.3 Register user shall be able to reactive account.

- 10. Registered users shall be able to make changes in setting.
 - 10.1 Registered users shall be able to delete their account from the settings.
 - 10.2 Registered users shall be able to set their notifications.
 - 10.3 Registered users shall be able to change their password.
- 14. Registered users shall be able to choose a primary language.
 - 14.1 Registered users shall choose from a list of available languages on LingoMingo.
 - 14.2 Registered users shall be able to change their primary language.
- 16. Registered users shall be able to choose a language to learn.
 - 16.1 Registered users shall be able to change the language they want to learn.
 - 16.2 Registered users shall choose from a list of available languages on LingoMingo.
- 18. Registered users shall be able to choose the skill level they're at.
 - 18.1 Registered users shall be able to change the skill level
 - 18.2 Skill level shall be scaled numerically. Example: (1-10 rating, 1=worse 10=best)
- 37. Registered users shall be able to rate.
 - 37.1 Registered users shall be able to rate other users.
 - 37.2 Registered users shall be able to rate the apps.
- 20. Registered users shall be able to choose a language they want to teach.
 - 20.1 Registered users shall be able to update the language they want to teach
 - 20.2 Registered users must be fluent in the language to be able to teach.
- 26. Registered users shall be able to match with another registered user.
 - 26.1 Registered users shall match based on:primary language, language to learn, language they want to teach, user rating
 - 26.2 Registered users shall be able to unmatch with another registered user.
 - 26.3 Registered users shall be able to match with multiple registered users at once.
- 4. Registered users shall be able to access and edit a profile.
 - 4.1 Registered users shall enter a new name while editing.
 - 4.2 Registered users shall be able to upload profile pictures.
 - 4.3 Registered users shall be able to delete profile pictures.
 - 4.4 Registered users shall be able to update profile pictures.
 - 4.5 Registered users shall be able to update their city.
 - 4.6 Registered users shall be able to add a phone number.
 - 4.7 Registered users shall be able to update a phone number.

- 4.8 Registered users shall be able to update profile statuses.
- 68. Registered users shall be able to ask for the help.
 - 68.1 Registered users shall be able to ask help with the support team.
 - 68.2 Registered users shall be able to email support teams.
- 28. Registered users shall be able to add another user as a friend.
 - 28.1 Registered users shall be able to unfriend.
 - 28.2 Registered users shall be able to block a friend.
 - 28.3 Registered users shall be able to create a chat with a friend user.
 - 28.4 Registered users shall be able to add additional specifications to matching, such as dialect.
- 24. Registered users shall be able to write a bio
 - 24.1 Bio shall be displayed in user's profile
 - 24.2 Registered users shall be able to update their bio
 - 24.3 Registered users default bio shall be empty.

Priority 2:

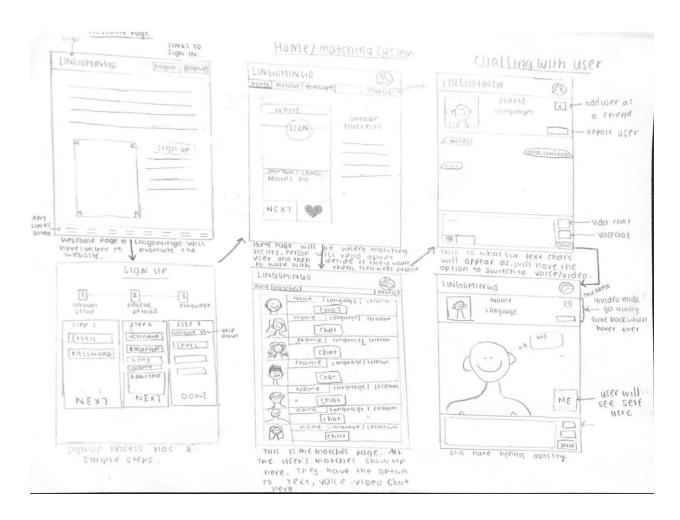
- 30. Registered users shall be able to choose icons for display
 - 30.1 Registered users' icon shall appear on their profile.
 - 30.2 Registered users shall be able change to a different icon.
- 34. Registered users shall be able to choose the methods for communication
 - 34.1 Registered users shall be able to communicate via text.
 - 34.2 Registered users shall be able to communicate via voice.
 - 34.3 Registered users shall be able to communicate via video call.
- 56. Registered users shall be able to make the group chat.
 - 56.1 Registered users shall be able to add people in group chat.
 - 56.2 Registered users shall be able to make video chat in a group.
 - 56.3 Registered users shall be able to make voice chat in a group.
 - 56.4 Registered users shall be able to see all the activity in a group chat.
 - 56.5 Registered users shall be able to mute the chat.
 - 56.6 Registered users shall be able to remove people from group chat.
 - 56.7 Registered users shall be able to choose whom to chat with.
 - 56.8 Registered users shall be able to decline the group chat request.

- 56.9 Registered users shall be able to turn on the notification in group chat.
- 66. Registered users shall be able to upgrade their account.
 - 66.1 Registered users shall be able to upgrade their account to premium to skip the ads.
 - 66.2 Registered users shall be able to stay in basic membership which comes with the ads.

Priority 3:

- 18. Registered users shall be able to change their skill levels.
 - 18.1 Registered users shall be able to change their skill levels.
 - 18.2 Registered users shall be able to set their skills level.
- 32. Registered users shall be able to learn multiple languages.
 - 32.1 Registered users shall be able to learn more than one language.
 - 32.2 Registered users shall be able to teach language to others.
- 40. Registered users shall be able to report an account.
- 42. Registered users shall be able to manage the match users.
 - 42.1 Registered users shall be able to skip through the match.
 - 42.2 Registered users shall be able to unmatch with the matched users.
 - 42.3 Registered users shall be able to check if the matched user is a real or bot.
- 51. Registered users shall be able to access other user's profiles.
 - 51.1 Registered users shall be able to like the other user's post.
 - 51.2 Registered users shall be able to make comments on other user's posts.

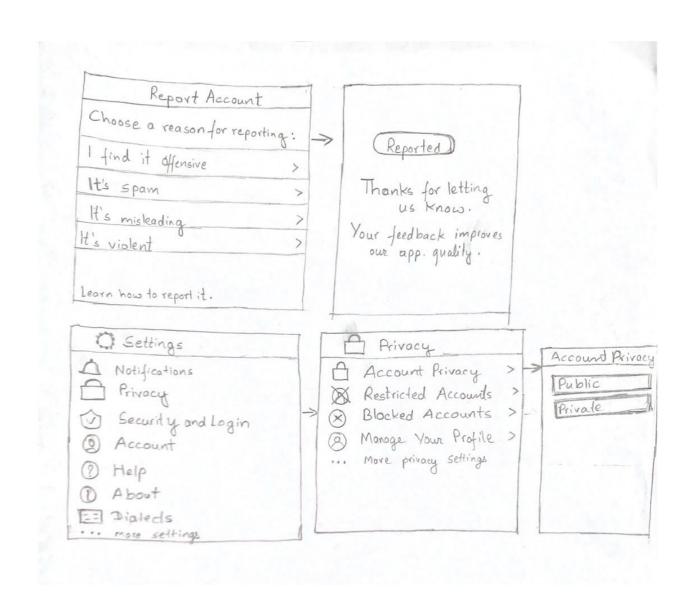
UI Mockups and Storyboards



Unregistered users start off on the welcome page. There will be a login option and a sign up option. The sign up page will be split into three separate, but easy steps : account setup, profile details, and language options.

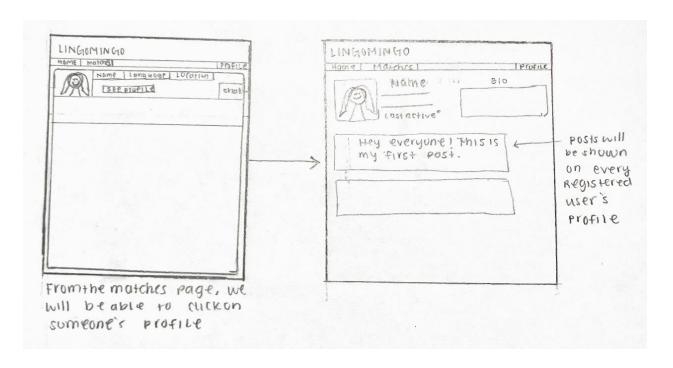
Once a user becomes registered, the home page will be where the matching system is. We will directly allow users to choose which user's they like and want to work with by clicking on the heart button or next to see the next person's profile.

Users will be able to see their matches on the matches page where they will have options to chat, add as a friend, see the person's profile and report any user.



When a registered user chooses to report another registered user, the following options will come up in the browser. It will then be investigated through the LingoMingo team.

Registered users can access their settings through their profile. Users can also change their account to premium through the account button. They also have the option of looking for specific dialects through the settings.



Any registered user will be able to look at other user's profiles if they have matched with them. They will be able to see when they were last active to make sure the user is not a robot.

High level database architecture and organization

Business Rules

- 1. An unregistered user can create multiple account
- 2. A registered user must be linked to only one account
- 3. Registered users can upgrade one account to premium
- 4. A registered user will match with multiple registered users based on similar languages
- 5. Registered users can create many chats with any of their matches
- 6. Profiles must choose at least one primary language
- 7. Profiles must choose at least one learning language
- 8. Regular accounts will display at least one advertisement
- 9. Accounts must have only one profile
- 10. Profiles can create multiple posts
- 11. Accounts can friend with multiple accounts

Description of Entities, Attributes and Relationships

Unregistered Users are users that haven't made an account on LingoMingo yet. Non-registered users wouldn't be able to utilize any of LingoMingo's features, besides the ability to sign up an account on LingoMingo and browse the homepages.

Entity: Unregistered User **Attributes:** User id

Relationship: Create (Account)

Non-registered user and account: 1 to M

Non-registered Users can make one account. Meanwhile an account is linked to one non-registered user. After an account is created, the non-registered user becomes a registered user.

Registered Users can use all the features LingoMingo has to offer: matching, messaging, voice calling, and video calling. Registered users can customize their profiles to express their individuality publicly or privately.

Entity: Registered User

Attributes: User_id: key, Account_id: key, Name: composite, Dob: multi-value, Age: derived, Primary Language, Learning language

Relationships: LinkedTo (Account), Make (Post), Choose (Primary Language)(Learning Language), Send (Chats)

Registered users and Account: 1 to 1

Registered users are linked to an account. If the register user deletes their account, both the user and account ids won't be reused.

Registered users and Primary Language: M to N

Register users can have one or multiple primary languages. By allowing registered users to select multiple primary languages, they would be able to teach multiple languages to other users and not be limited to having one primary language.

Registered users and Learning Language: M to N

Registered users can have one or multiple learning languages. Some registered users may want to learn multiple languages and would be able to select multiple languages.

Registered users and chats: M to N

Registered users can chat to multiple users simultaneously. Users aren't restricted to messaging one person at a time as it would be limiting for the user's experience.

Accounts are created by unregistered users which then become linked to a registered user. Every registered user would be assigned to an account that is unique to them. In other words, if the registered user decides to delete the account, the account id won't be reused. Also, there are two types of accounts: regular and premium. Regular accounts are free and provide almost all the features on LingoMingo has to offer besides the no ads feature.

Entity: Account

Attributes: Account_id: key, User_id: key, Email: key, Profile_id: key, Average rating, Type (Regular or

Premium)

Relationship: IsA(Free/Premium), Has(Profile), Gain(Rating)

Account and Regular: 1 to 1, ISA relationship

An account can be a regular or premium account. Regular accounts are the default account type that is free and offers almost all of the LingoMingo's features besides the no-ad feature.

Account and Premium: 1 to 1, ISA relationship

An account can be a regular or premium account. Premium accounts are the upgraded account type that isn't free and offers all the features LingoMingo has to offer. Premium accounts won't have to watch advertisements every so often unlike regular accounts.

Account and profile:1 to 1

An account can only have one profile, as having multiple profiles wouldn't serve any purpose.

Account and rating: 1 to 1

An account has an averaged rating based on their past conversation. If the registered user was helpful, they would have a higher rating. Meanwhile users that are inappropriate would either have very low ratings or be reported. Each account has one average rating, which is why it's one to one relationship.

Profiles are created upon finishing the sign up process. In a profile, registered users are able to express themselves through posts, bios, languages that they know and languages that they want to learn.

Entity: Profile

Attributes: Profile_id: key, Account_id: key, Posts

Relationships: Have(Account), Have(Post)

Profile to Account: 1 to 1

Each profile is unique to an account. No profile id would be reused for another account.

Profile to Posts: 1 to M

Each profile can have none to many posts depending on the user's preference. Posts are created by one registered user and would remain at the registered user's profile.

Chats are created from the matching system, where two registered users match and start talking to one another. Chats, for now, remains as only messaging but we may add more entities for the video and voice feature. Registered users can chat with multiple people simultaneously.

Entity: Chat

Attributes: Chat_id:key, User_id:key, Learning Language **Relationships:** Send(Registered User), MadeBy(Match)

Chat to Registered user: M to N

Chats are created from the matching system between two registered users. Or the chat can be created from a friends list, if we manage to implement the friend system.

Posts are made by users and appear on the user's profile. Registered users can post profile statuses to express themselves publicly or to their friends. Posts are posted in the registered user's profile, which can be reported if the post is inappropriate.

Entity: Post

Attributes: Profile_id:key, Post_id:key

Relationships: Have(Profile), Made(Account)

Profile to Post: 1 to M

Registered users can have multiple posts on their profile if they choose to post. Posting is optional and can be public or private. Posts are created by one profile, as each profile is responsible for their own posts.

Advertisements appear only for regular accounts. LingoMingo allows other companies to advertise their products on LingoMingo. Advertisements appear only for regular accounts. Premium accounts won't have any advertisements when they chat with another registered user. Advertisements will appear periodically throughout the use of LingoMingo

Entity: Advertisement

Attributes: Ad id: key, Company name: composite

Relationships: Have(Account)

Advertisement to regular account: M to N

Advertisements can be seen by many regular accounts, while each regular account can see many advertisements.

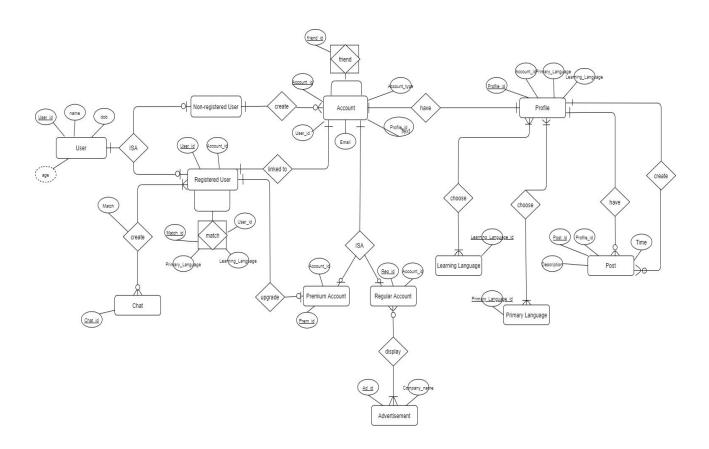
Ratings are something an account receives once they are finished chatting. Each account will average out the ratings it receives to give one definitive rating.

Entity: Rating

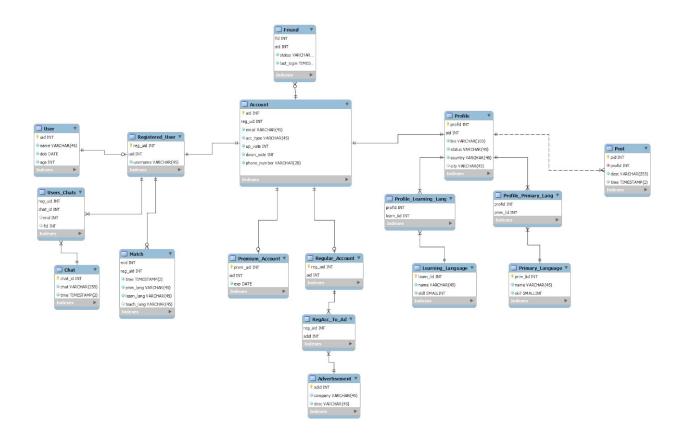
Attributes: Account_id, chat_id, rating_id

Relationships: Gain(Account)

Entity Relationship Diagram



Database Model



 Media storage: Decide if images and video/audio will be kept in file systems or in DB BLOBs (decision on file vs. BLOBs must be made by the end of M2). Describe any other special data format requirements like for video/audio/GPS etc.

All media files will be stored on the server. The database will store the file path, where the media file is located. For now, we don't have any special data format requirements yet. If we do, we'll include it.

For example, when a user visits a person's profile. The person's profile picture will be obtained through the file path that's provided by the database.

Search/filter architecture and implementation: what will be the alg/SW for search; how will you organize search items from the user; what DB terms will be searched, how it will be coded and organized in the DB (check instructors' suggestions in the class. OK to use SQL and %like).

Our most used search function would likely be the matching process. User are getting match suggested to them on a daily basis. We plan to implement the search function inside Django since the framework has very powerful database management built in. For retrieving users, we can use User.objects.all() and make them filterable by doing Users.objects.filter(field=query). We can encapsulate these functions into one of the API for convenience and security. For example if we want to match a user who knows english and wants to learn german, the match api will filter the user query with primary_lang=german and combine that with learning_lang=english. So that the user gets matches that share at least one commonly known language.

High Level APIs and Main Algorithms

APIs

Login User:

When a user types in their username and password and clicks on submit, the data will be sent to the database and checked if the user exists by username. If the username exists, then the password will be checked if valid or not. The server then returns a response with the user's session and redirects users to their page of intended visit or back to the index page.

Register User:

This is used for users to register and make an account. Users will add their name, email, password, birthdate, location, language they want to learn and teach, skill level for each language to make their account. The data then will be sent to django as a POST request. We validate the form and if valid, save the form data into the User object and store the user object into our database. We then authenticate() function to authenticate the user using their username and password from the form, and use login() function to return a session to the user. Both authenticate and login are part of django.contrib.auth API/ user management system.

Match user:

In this API, users are matched based on the language they want to learn/teach and their skill level. There will be three skill levels: advanced, intermediate and beginners. Users will also be matched by their login timestamp.

Chat, Audio and Video Calling:

Once the user gets matched with other users, they will have the options to text, audio call or video call. Text based chats will be implemented with websocket. The frontend will send chat text to the server with websocket. The server will then forward the text to all other users in that chat session, and store the chat in the database.

Audio and Video Calling will be implemented using webrtc's API.

(https://webrtc.org/getting-started/media-devices)

We first capture the user's media device. Then initiate the peer connection with a stun server. Upon connection with a remote peer, webrtc's api will start streaming media over likely over a TURN server. At the current stage we are still familiarizing ourselves with webrtc and technology behind it. But we can possibly set up a TURN/STUN server on our aws instance as well.

Rate User:

This API will be used to rate the app and the users. Once the user matches with another user it will be allowed to rate users. To rate users are asked if they would like to match with the other user again. If yes, they are prompted with a list of pre created tags. Tag examples: would you match with this user again?, caring, inspirational, accessible, clear, tough, hilarious, respectful. These tags will display on the user's profile. If no, the user will be removed from the possible match list.

Rate Website: Users can also rate the website and users will be using the thumbs up and down to show if they like the website or not.

Report User: This API will be used to report a user. One user fills out a report form the reported user's account name and reason for the report will be sent to the API. The server will then check the user for other reports and send a suspension to the users account.

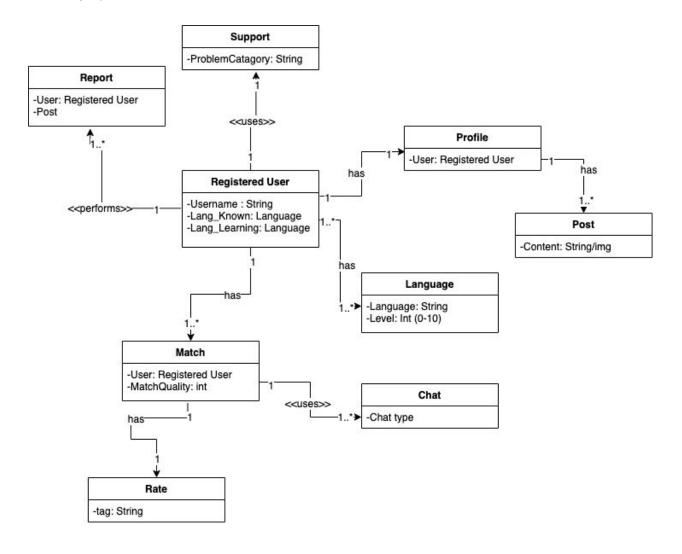
Algorithms

Matching:

First filter all users by primary language against Matcher's learning language. Combine the list filtering all users by learning language against the matcher's primary language. Then sort the user list by the rating from highest to lowest, lastly sort them by last login timestamp from most recent to least recent. So that matcher's matches will guarantee at least one common known language, and have recent and high quality users presented to them.

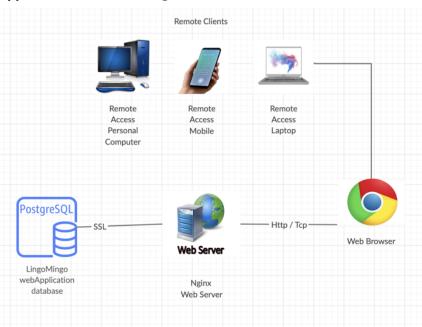
High Level UML Diagrams

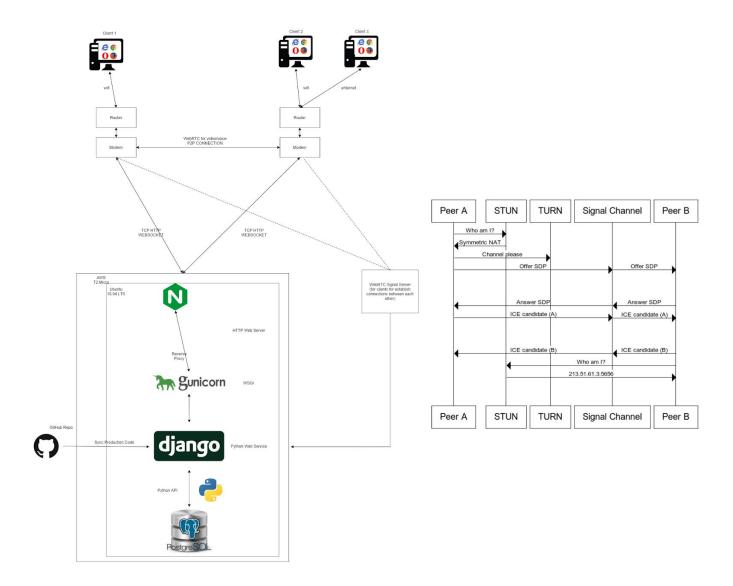
Matching System



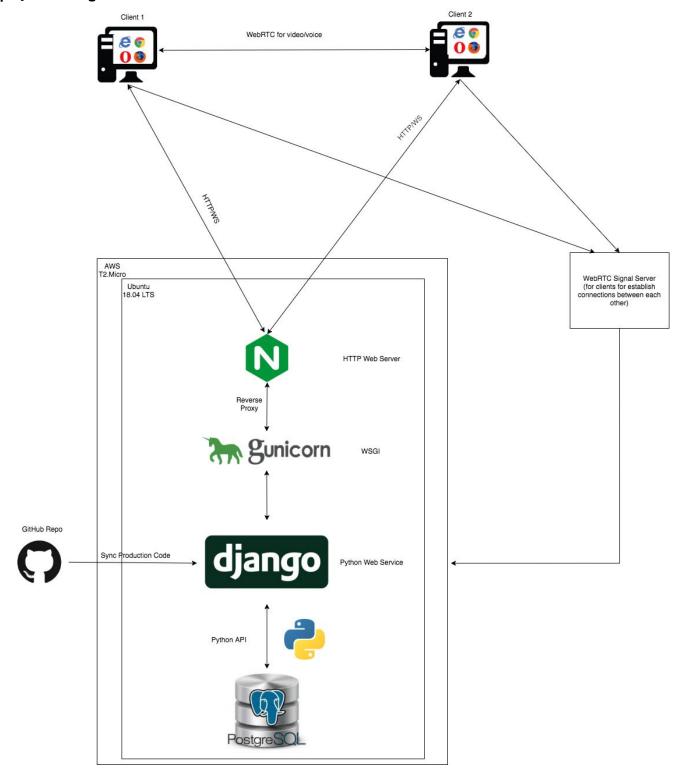
High Level Application Network and Deployment Diagrams

Application Network Diagram





Deployment Diagram



Identify actual key risks for your project at this time

Schedule Risk

Group meetings - Since, all face to face meetings became online, it is taking some time for us to organize the group meeting. We cannot meet in person and this is a group project and it is very important in this class to meet with your group to do the project smoothly. To solve this problem, we are doing discord meetings to assign and discuss all the problems we are having with the project. We are texting and figuring out what time is suitable for each member in the group and doing discord meetings.

Skills Risk

Video chat - We are thinking to implement video chat as one of the features of our website. There are chances of having problems in implementing it, therefore we may decide to start with a basic wordpress website for now. Later on, by looking at the source/html code, we may inherit the specific skills to accomplish the task we wish to obtain. A simple wordpress website will serve our purpose very well at an initial stage of our project.

Teamwork Risk

Because of the COVID-19, we were not able to continue our in-person weekly meeting. So we are forced to go for online meetings through ZOOM. Which may reduce productivity and lower the quality of product. To resolve this issue, we are doing an online meeting and discussing all the issues and progress of the project. We are continuously communicating via text and updating about the project with our team members. With everything going on, there just might not be enough time to implement all of the things we want to.

Legal/Content Risk Not applicable

Project management

To manage M2 tasks we split up each section and each section had approx. two workers. We did all of our work on Google Drive that way the team members could collaborate on their section. We followed the same outline from M1. Because our team is using Google Docs to collaborate off of eachother, it is challenging for each member to push their work on the doc through github. To manage future tasks we are going to implement the use of Trello to create Kanban boards to manage individual tasks. During our scrum meetings we will be able to create tasks, and move them from in progress to finished in order to track the current state of our application. It will also allow us to keep track of each team member's efficiency in completing tasks and when they might need assistance.