SW Engineering CSC648/848 FALL 2020

DoReMeet

Team 02

Milestone 2

Nimiksha Mahajan (nmahajan@mail.sfsu.edu) - Team Lead
Meet Patel (mpatel6@mail.sfsu.edu) - Backend & Database Lead
Luke Chang-Knezevich (lchangknezevich@mail.sfsu.edu) - Frontend Lead
Mike Bolanos (mbolanos1@mail.sfsu.edu) - GitHub Master & Backend Engineer
Jungsun Eoh (jeoh@mail.sfsu.edu) - Database & Backend Engineer
Vincent Tse (vtse3@mail.sfsu.edu) - Server Admin & Frontend Engineer

History Table

Version	Date	Comments
M2V2	12/05/2020	Submission updated with Feedback
M2V1	11/01/2020	Initial Submission
M1V2	10/11/2020	Submission updated with Feedback
M1V1	09/30/2020	Initial Submission

Table Of Contents

1. Main Data Items and Entities V2

1. **Registered User/User** - A user is any individual who has created an account on the website using their name, a valid email, date of birth and password.

2. **Unregistered User** - Any individual who has not created an account is an unregistered user. An unregistered user is eligible to create a free or premium account.

3. **Landing Page** - The first page any individual, both registered and unregistered, will see when they visit the website. The page shall lead to a sign up or login page.

4. **Account** - An account is an entity associated with a user, and the site sees each user in terms of their valid account.
   a. *Active/Activated account* - An active account is a user account that is in service and the information is stored on our database. An account is active when the user first creates and sets it up. Also, the account is activated after the user unpauses their account.
   b. *Paused Account/Inactive Account* - Users have the ability to pause their account or inactivate it without deleting their information. This can be done if the user wants to take a break from using the website, but don't want to leave the platform all together.
   c. *Disabled Account* - Users with an account have the option to delete their account and remove all their information from our database. Such an account is a disabaled account.

5. **Account Type** - Every registered user can have a free account or a premium account. A free account is created by default, but the user has the option to opt for a premium account at a small fee, and access the premium features of our website. A user has the ability to switch between a free account and premium account, and vice versa.

6. **Free Account** - A free account is a regular account that every registered user gets by default. Unlike the premium account, the free account does not have paid features.

7. **Premium Account** - A premium account is a paid account which in addition to all the features a regular account offers, offers additional features.
   a. *Premium Features* - A premium account includes premium features including unlimited number of likes, extended search radius to 150 miles, increased portfolio size (from 80mb in free account to 120mb), highlighted user profile, ability to rewind on a passed potential match and no ads while using the website.
   b. *Rewind* - As a part of their premium membership, users can "rewind" on their last potential match. This means that if they passed on a user, but didn't mean to, or they changed their mind, they have the ability to go back and look at the profile again, and like or pass as they wish.

8. **Profile -** A profile is the user's information on the website. They can set it up in any way they like, displaying whatever information they feel is relevant to them. The site provides means for them to curate their profile and optimize it to gather more traction.
    a. *Bio* - A bio is a brief text description that the user can add to their profile to describe what they are looking for, or talk more about their work.
    b. *Photos* - Users have the ability to add pictures of their work or anything relevant that showcases their interest or what they are looking for.
    c. *Videos* - Similar to photos, users can add short videos to their profile showing their work as well. This is especially useful for musicians, dancers and filmmakers, who will get a chance to showcase a snippet of their work.
    d. *Links* - Users can add links to their website or our portfolio pages on their bio to endorse their work.
    e. *Linked Accounts* - Users have the ability to link other accounts to our website. For instance, they can link their Instagram, Youtube or Spotify account to show more of their portfolio.

9. **Preferences** - Users can customize their profile by adding preferences to it. These preferences will make their account more personalized and will help them get more targeted potential matches. The users can add a variety of preferences from the given choices:
    a. *Art Categories* - The users will be able pick one of the five art categories - *painting, dancing, music, photography, cinema* - and their profile will get hits in that category only, i.e. dancers will see others dancers while singers will be put in a pool of other singers. This will help users get direct access to the community they are seeking to join.
    b. *Work Type* - The site provides a platform for both passion projects and work. Users can filter their potential matches based on their preference and choose to work with others on serious projects or just collaborate on a passion project/hobby.
    c. *Distance* - The users have the option to set distance preferences if they are open to meet other people in person. They can choose any radius between 1 mile to 100 miles (150 miles for premium account users) of their geographical location, and they will see potential matches only in the specified distance. Premium account holders can extend their radius upto 150 miles.
    d. *Meet online only* - Users also have the option to set their profile preference to meet online only - i.e. they can opt to see only users who want to meet online and collaborate virtually. They don't have to share their location in this case, but will still have the option to filter what countries or cities they want to see the artists from.
    e. *Gender* - Users will have the option choose artists based on gender (Mael, Female, Trans)
    f. *Age* - Users can also set an age range as a preference criteria for their potential matches. They can choose any range between 18 and 100 years of age. Users have the ability to set *minimum* and *maximum* age filters for their potential matches.
    g. *Skill* - Users have the ability to set their skill level (novice, competent, experiences, expert) on their profile and can choose to only see potential matches based on their skill level. For instance, they can choose to learn with other novices or work with an expert.

10. **Tags** - Users will be able to add tags to their profile, which highlight key aspects about them, for instance a user is interested in art, portrait painting in particular and wants to meet fellow artists younger than 30 years in 50 miles of his radius. Also this is just a hobby for him, so he can add portraits, hobby, meet in person, as more such things as his tags, which help condense the information on his profile and get it more traction from other users looking for similar things.
    a. *Custom Tags* - Users can create their own custom tags to really highlight what they want. Some instances could be “unpaid”, “urgent”, “fun”, “volunteer partner”, “LGBTQ friendly” or anything that they think is relevant to their profile. The custom tags help share more about the user with potential matches and make more fitting connections.

11. **Potential Matches** - Potential Matches are the users that a user sees based on their preferences. The potential matches fit whatever criteria the user has set for art category, distance, age, skill or gender. The user has the option to either pursue these matches or forego them. These are just people who the system thinks would be a good match with the user.
    a. *Waiting State -* Potential Matches that are liked by one user are in a waiting state till the other user either likes or passes on them.
    b. *Filtered Results* - The potential matches can be filtered based on the preferences set by the user. For example, if a user sets distance limitations, all potential matches outside that radius will be filtered out.

12. **Likes** - Likes are the potential matches that the user wants to pursue. If the user does want to work with these potential matches and if they like them back, then the two of them will be connected. Regular account holders can like upto 100 people per day, whereas premium account users will have unlimited number of likes.

13. **Pass** - Passing are the potential matches that the user does not want to pursue, because they don’t match their interest or they don’t think would be a good match for them. Even if these potential matches, that the user passed, likes them back, they will not be connected, given it is not a mutual interest to work together.

14. **Matches/Matched users** - Matches are the potential matches that the user liked and who liked them back. Given their mutual interest in being connected, the site matched the 2 users and provided them a platform to interact with text messages, share photos/media etc and collaborate on whatever project they were both seeking to pursue.

15. **Community Page -** The community page holds joint projects that users have uploaded. Users who match with others on the site and go on to create something have the option to upload that project here and browse the projects that others have made. Users can vote on which project they think deserves to be the project of the month.

16. **Highlights of the Month -** The highlights of the month will be community voted projects. Each month the projects with the highest number of votes will be showcased as the highlights of the month for all users.

17. **Report** - A user has the ability to report any potential matches or matches if they engage in any illegal, violent or indecent activity. A reported user will be investigated and can have their account suspended if found guilty.

18. **Flag** - A user has the ability to flag any inappropriate content that they see from any potential matches or matches. The content will be reviewed, and if found in violation of the site's policies will be removed from the site. Also the user who posted it, will be notified with a warning regarding the content.
    a. *Inappropriate Content* - Content that violates our websites Community Guidelines, and/or aims to promote violence, hate or bias is considered inappropriate.

19. **Block** - A user has the ability to block any potential matches or matches if they feel don't want to engage with that person. Any blocked user will not be seen by the user and will not be able to connect with or contact the user.

20. **Unmatch** - A user can choose to unmatch with any of their matches, if they realize they don't want to pursue that connection, or don't want to stay connected for any reason. An unmatched user is not blocked, and can still appear in the users potential matches.

21. **Message List** - The message list page shows a list of all the chat messages of a user with all their matches. The list has a little preview showing the users match's name and their text exchanges.

22. **Messaging/Chat Window** - When connected with a user (a potential match who the user likes and who liked them back), the user can connect with the match using a messaging or chat window, where they can share information via texts, photos or short videos.

23. **Conversation prompts** - The chat window gives initial conversion prompts when the users are first matched. This is to promote engagements and help users reduce the anxiety of coming up with a conversation starter. These prompts are suggested to promote engaging and productive conversations between the matches.

24. **Password** - The user needs to set a password when they create an account. The user then uses this password to log into the website from any device or browser. The password needs to meet certain security requirements (password length, special characters etc. as per the standard), to ensure that it is a strong password that cannot be easily cracked.
    a. *Forgotten Password* - The user might forget their password, and they have the ability to say they forgot their password, at which point they'll be given a special link to reset their password via email.

   b. *Reset Password* - The user can choose to reset their existing password (even if it is not forgotten) and will have the ability to do so.

25. **Administrator/Admin** - An administrator or admin is someone who has access to the site's backend is able to investigate any flagged or reported activities on the website. An admin has the ability to remove any content/user that is found in violation of the website's policies.

26. **Customer Service** - Customer Service is a support team for the website that can help the users with any technical or logistical difficulties and can help them navigate the website better. Aso the customer service is able to support the user with any settings, password updates etc.

2. Functional Requirements V2

Priority 1

Registered User (also referred to as user):

1. Users shall be able to sign in to the website.
2. Users shall be able to sign out from the website.
3. Users shall be able to change their existing password.
7. Users shall be able to Like other users.
8. Users shall be able to Pass other users.
9. Users shall be able to view potential matches' profiles.
10. Users shall be able to view their matches' profiles.

Unregistered User:

15. An unregistered user shall be able to create an account using a valid email.
16. An unregistered user shall be able to only see the landing page of the website.
18. An unregistered user shall be required to enter a date of birth upon account creation.
25. An account shall allow its user to modify the password.

Account:

19. An account shall have one valid email linked to it.
24. An account shall be active to allow users to continue matching with others.

Profile:

32. A profile shall allow users to upload pictures.
33. A profile shall allow users to delete pictures.
34. A profile shall allow users to upload text descriptions as bios.
35. A profile shall allow users to update their bio.
36. A profile shall allow users to delete their bio.
37. A profile shall allow users to upload short videos.
38. A profile shall allow users to link their Youtube account.
39. A profile shall allow users to link their Spotify account.
40. A profile shall allow users to link their Instagram account.
41. A profile shall allow users to link any websites.
42. A profile shall allow users to change what is displayed publicly on their account.

Preferences:

43. Preferences shall allow users to filter potential matches based on art categories.
44. Preferences shall enable users to choose if they want to meet online.
45. Preferences shall enable users to choose if they want to meet in person.
46. Preferences shall enable users to share their location.
47. Preferences shall enable users to change their match preferences to online only.
48. Preferences shall enable users to change their match preferences to local only.

49. Preferences shall let users set a location radius for meeting in person.
52. Preferences shall enable users to adjust preference of minimum age for potential matches.
53. Preferences shall enable users to adjust preference of maximum age for potential matches.
54. Preferences shall let users to filter match preferences based on gender.
55. Preferences shall let users to set skill levels.
56. Preferences shall allow users to update their skill levels.
57. Preferences shall let users to filter potential matches based on skill levels.

Community Page:

58. The community page shall contain work that has been created by a collaboration of users.
59. The community page shall contain highlights of the month.

Password:

67. Passwords shall be updatable via email in case the user forgets their password.

Matches:

70. Matches' profile shall be accessible to the user.
71. Matched users shall be able to communicate via text message.
77. Matches shall be able to unmatch their existing matches.
78. Matches shall be made when both users like each other.

Potential Matches:

79. Potential Matches shall be liked by both users to become a match.
80. Potential Match shall be removed if a user passes on them.
81. Potential matches shall be matched when both users have liked.

Chat Window:

82. Message List shall show a list of all the conversations between matches.

Priority 2

Registered User (also referred to as user):

11. Users shall be able to report other users.
12. Users shall be able to flag inappropriate content.
13. Users shall be able to block other users.
14. Users shall be able to vote on collaborations posted on the community page.

Account:

20. An account shall be paused by the user anytime they want.
21. An account shall allow users to login while it is paused.
22. An account shall be disabled by users through the settings page.
23. An account shall be reactivated by the user anytime they want.

Premium Account:

26. A premium account users' profile shall be highlighted.
27. A premium account users' shall have access to premium features.

Admin/Administrator:

63. Administrators shall be able to see a flagged user's account.
64. Administrators shall be able to see a reported user's account.
65. Administrators shall be able to delete a user's account.
66. Administrators shall notify a flagged user about any removed content.

Tags and Custom Tags:

68. Tags shall be added to the users profile to highlight preferences.
69. Custom Tags shall allow users to list special requirements.

Matches:

72. Matches shall be able to share photos via chat.
73. Matches shall be able to share videos via chat.
74. Matches shall be able to search within their match list by first name.
75. Matches shall be able to search within their match list matches by username.

Chat Window:

83. Chat window shall allow users to sort matches by name.
84. Chat windows shall allow users to sort matches chronologically.

Priority 3

Registered User (also referred to as user):

4. Users shall be able to troubleshoot login problems.
5. Users shall be able to activate 2 factor identification.
6. Users shall be able to deactivate 2 factor identification.

Unregistered User:

17. Unregistered users shall have to confirm their account registration via a valid email.

Premium Account:

28. A premium account user shall have to pay for premium features.
29. A premium account user shall be able to save their payment information.
30. A premium account user shall be able to update their payment information.
31. A premium account user shall be able to cancel their subscription.

Preferences:

50. Preferences shall let users filter by country if they choose to meet online.
51. Preferences shall enable users to filter by city if they choose to meet online.

Highlights of the Month:

60. The Highlights of the Month shall contain one work that has the most votes for Uniqueness/Creativity.
61. The Highlights of the Month shall contain one work that has the most votes for Effort.
62. The Highlights of the Month shall contain one work that has the most votes for Quality.

Matches:

76. Matches shall be able to use default conversation prompts in chat.

Chat Window:

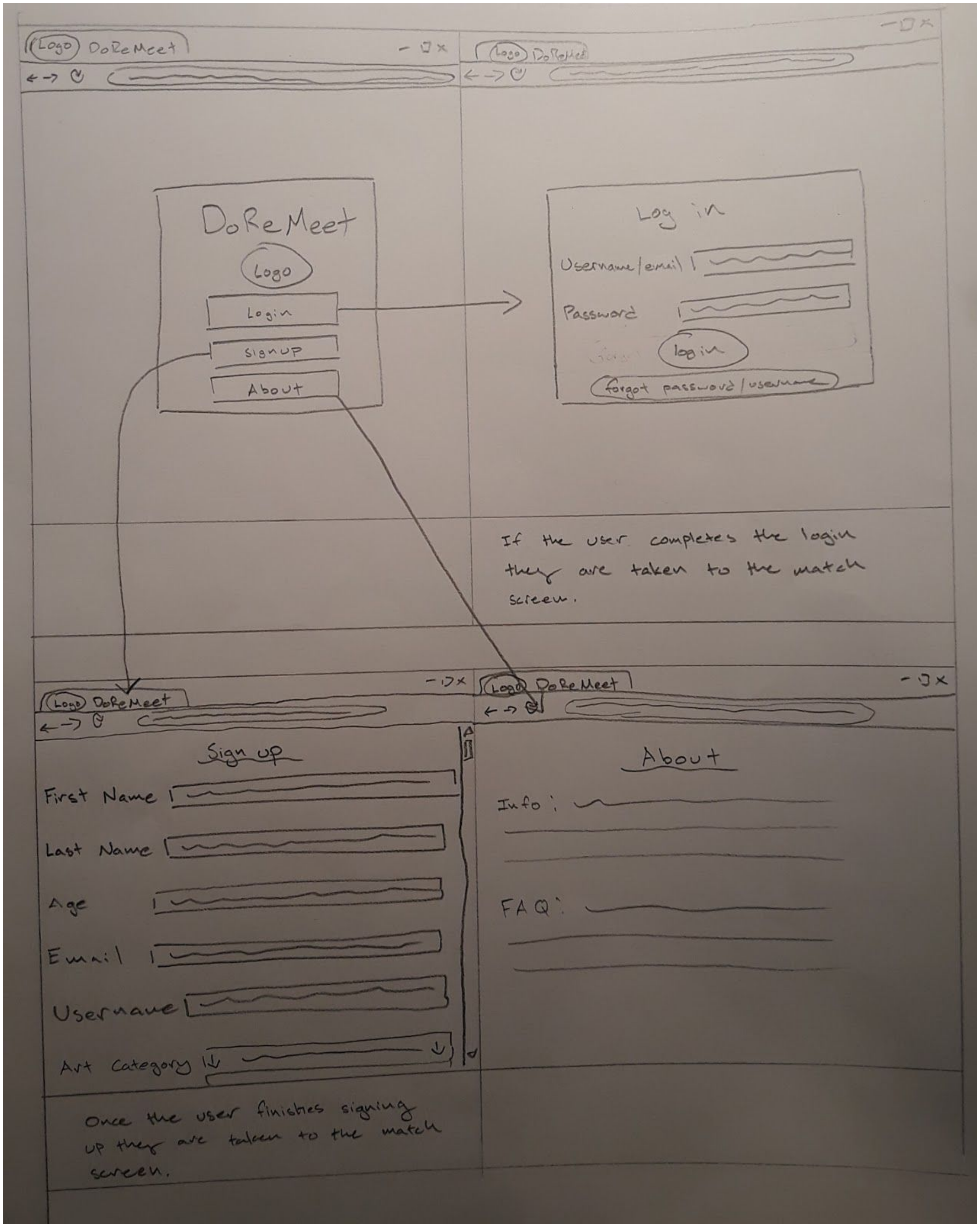
85. Chats with each match shall have the option to interact via audio call.
86. Chats with each match shall have the option to interact via video call.

Customer Service:

87. Customer Service shall be able to chat with users.

3. UI Mockups and Storyboards

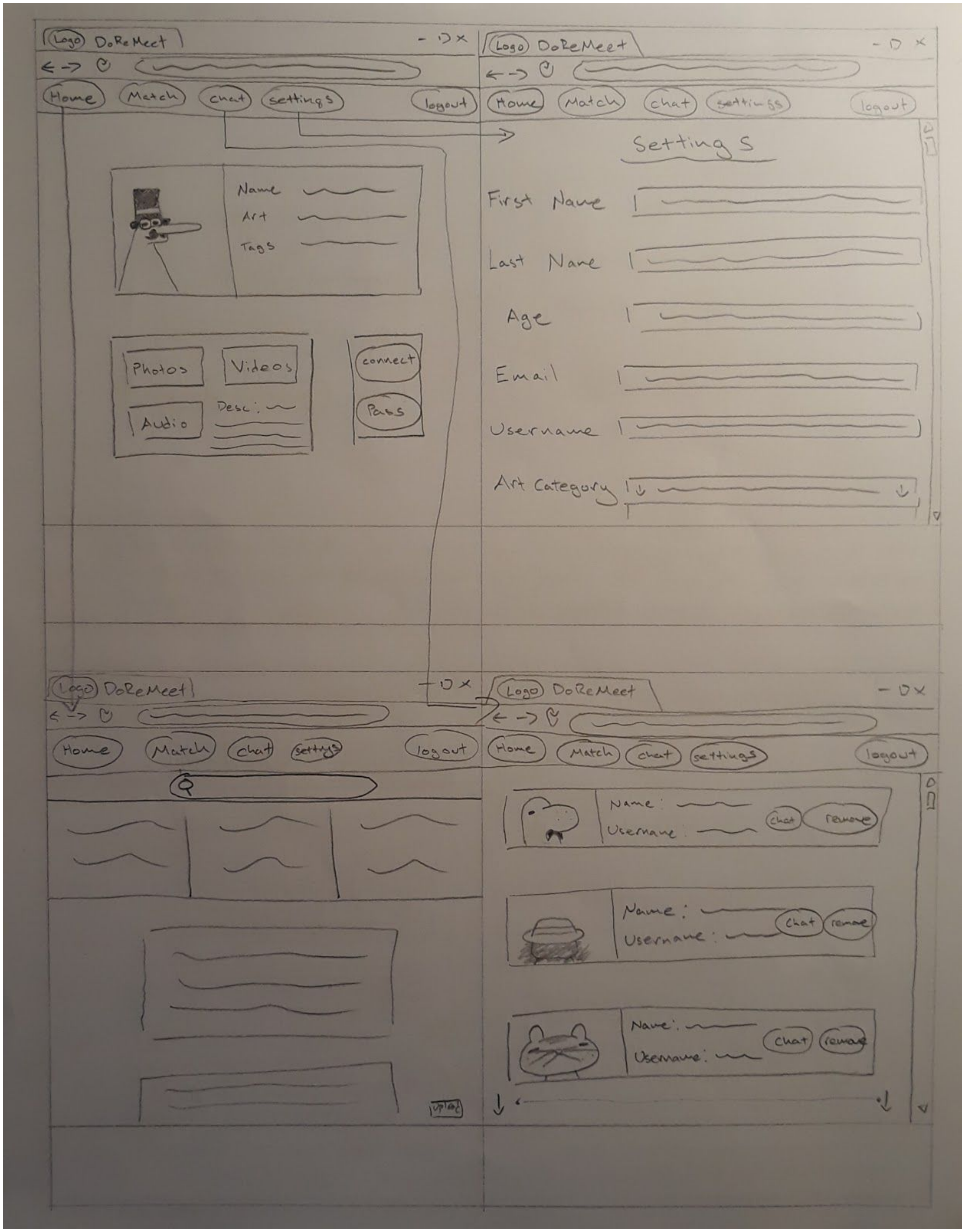
General Site flow (1)



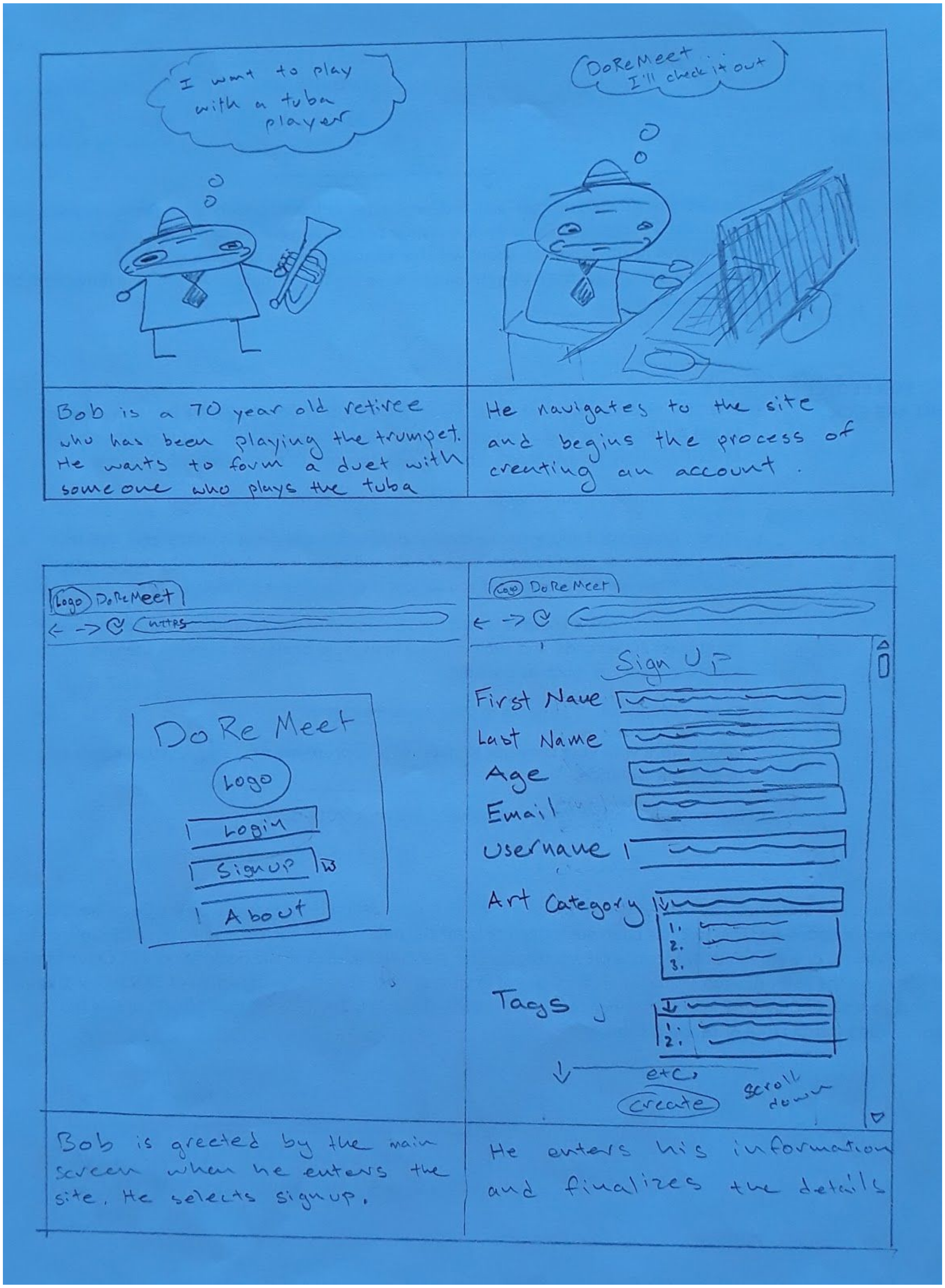
If the user completes the login they are taken to the match screen.

Once the user finishes signing up they are taken to the match screen.

General Site flow (2)



Use Case 1: Forming a duet

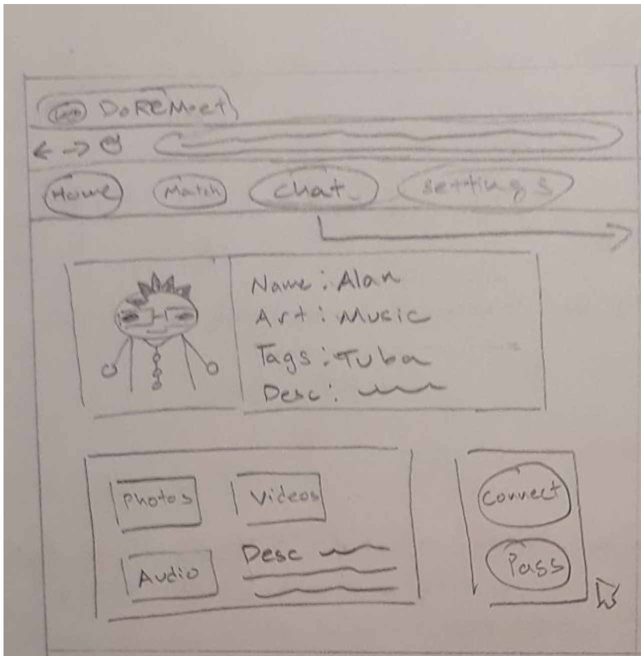


Bob is a 70 year old retiree who has been playing the trumpet. He wants to form a duet with some one who plays the tuba

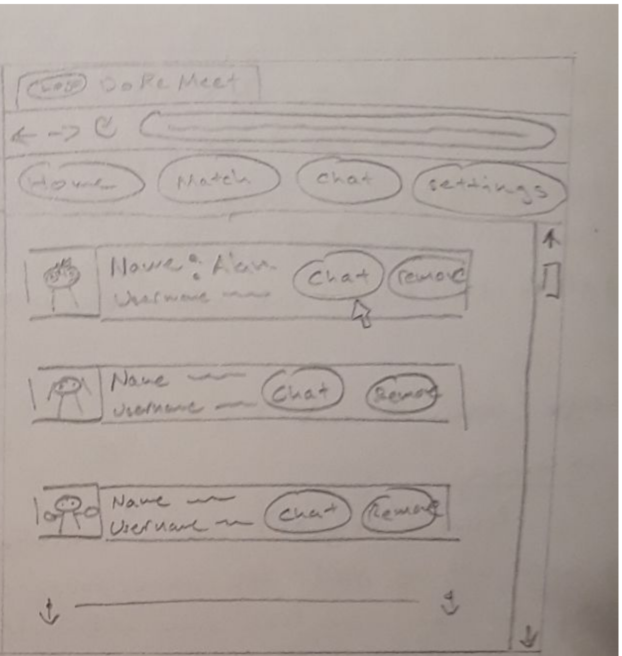
He navigates to the site and begins the process of creating an account.

Bob is greeted by the main screen when he enters the site. He selects signup.

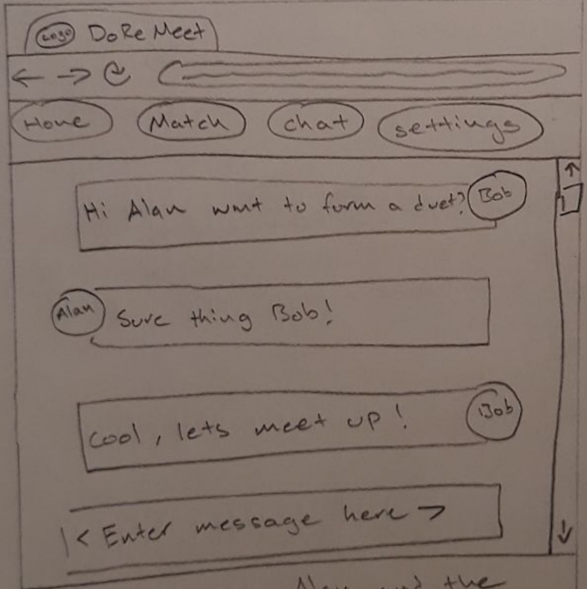
He enters his information and finalizes the details



Bob is taken to the match page after his account has been created. He chooses to connect or pass with the people he matches with.



Bob has chosen to match with Alan, similarly Alan has matched with Bob. Bob clicks the chat button and finds Alan in his chat page.



Hi Alan want to form a duet?

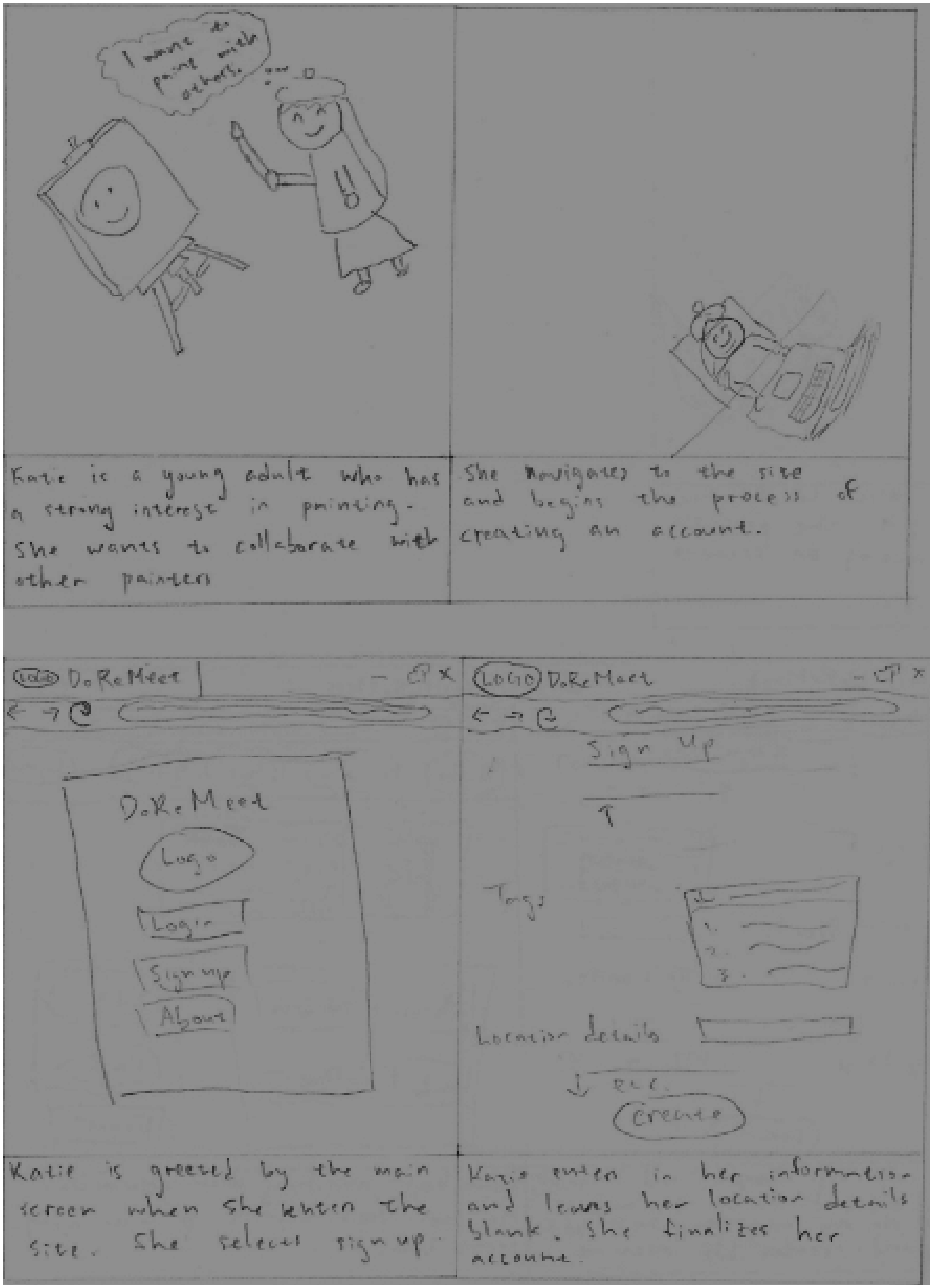
Sure thing Bob!

cool, lets meet up!

< Enter message here >

Bob messages Alan and the two of them agree on the details of their meet up.

Use Case 2: Unwanted account details - sharing location



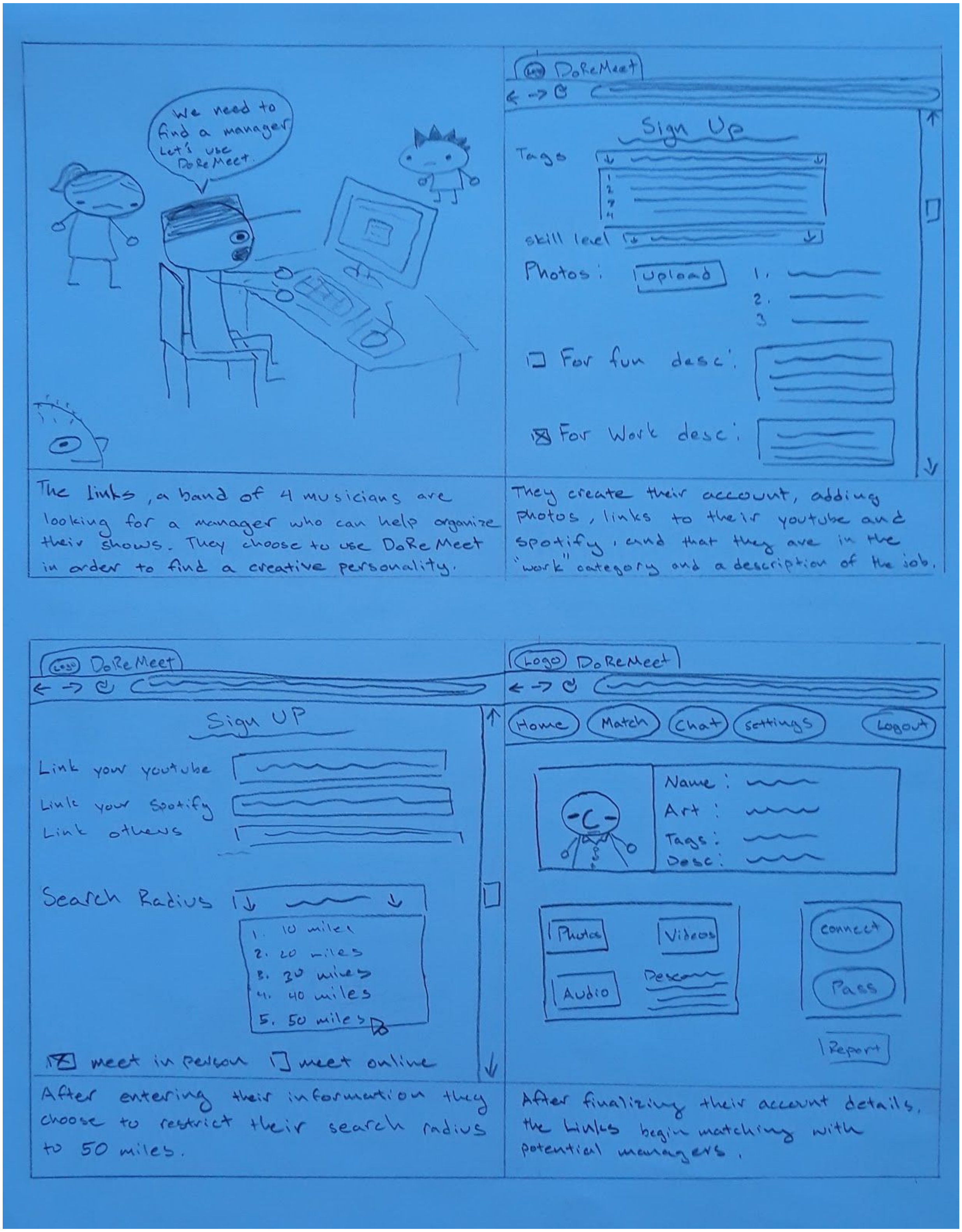
Katie is a young adult who has a strong interest in painting. She wants to collaborate with other painters

She navigates to the site and begins the process of creating an account.

Katie is greeted by the main screen when she enters the site. She selects sign up.

Katie enters in her information and leaves her location details blank. She finalizes her account.

Use Case 3: Band Looking for Manager



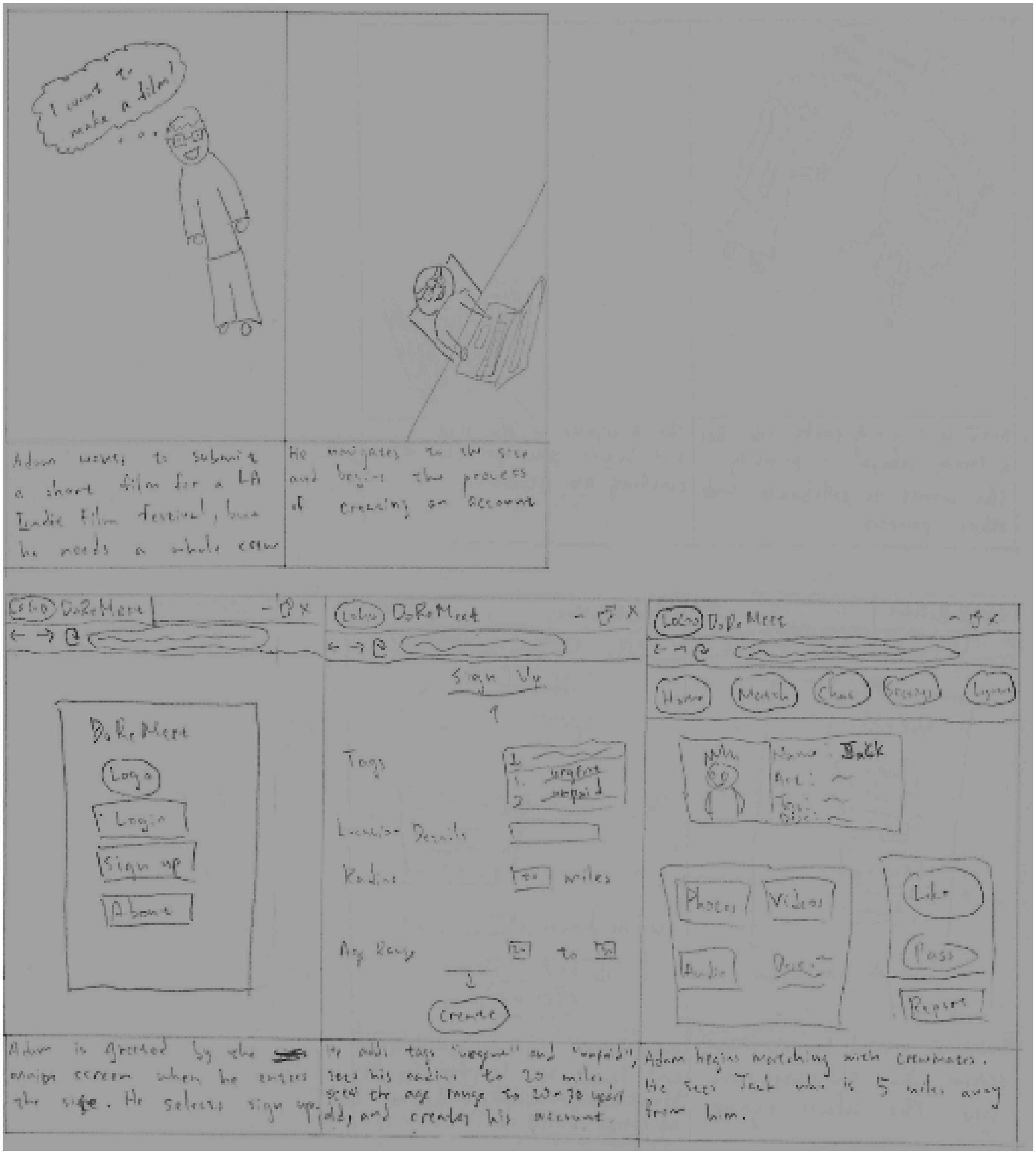
The Links, a band of 4 musicians are looking for a manager who can help organize their shows. They choose to use DoReMeet in order to find a creative personality.

They create their account, adding photos, links to their youtube and spotify, and that they are in the "work" category and a description of the job.

After entering their information they choose to restrict their search radius to 50 miles.

After finalizing their account details, the Links begin matching with potential managers.

Use Case 4: Director looking to create movie for an Indie Film Festival



Adam wants to submit a short film for a LA Indie Film Festival, but he needs a whole crew

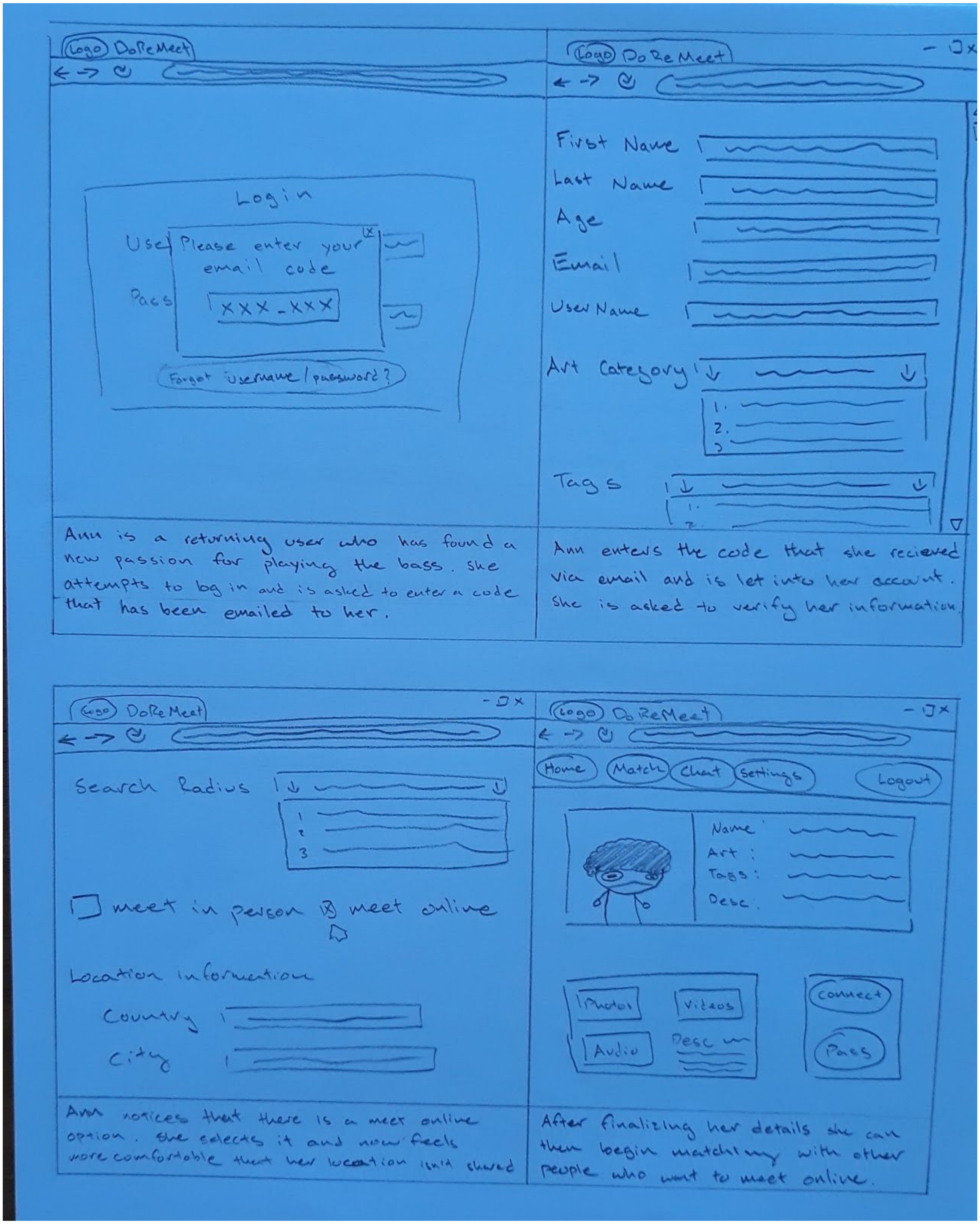
He navigates to the site and begins the process of creating an account

Adam is greeted by the main screen when he enters the site. He selects sign up.

He adds tags "urgent" and "unpaid", sets his radius to 20 miles, sets the age range to 20-30 years old, and creates his account.

Adam begins matching with crewmates. He sees Jack who is 5 miles away from him.

Use Case 5: Returning user



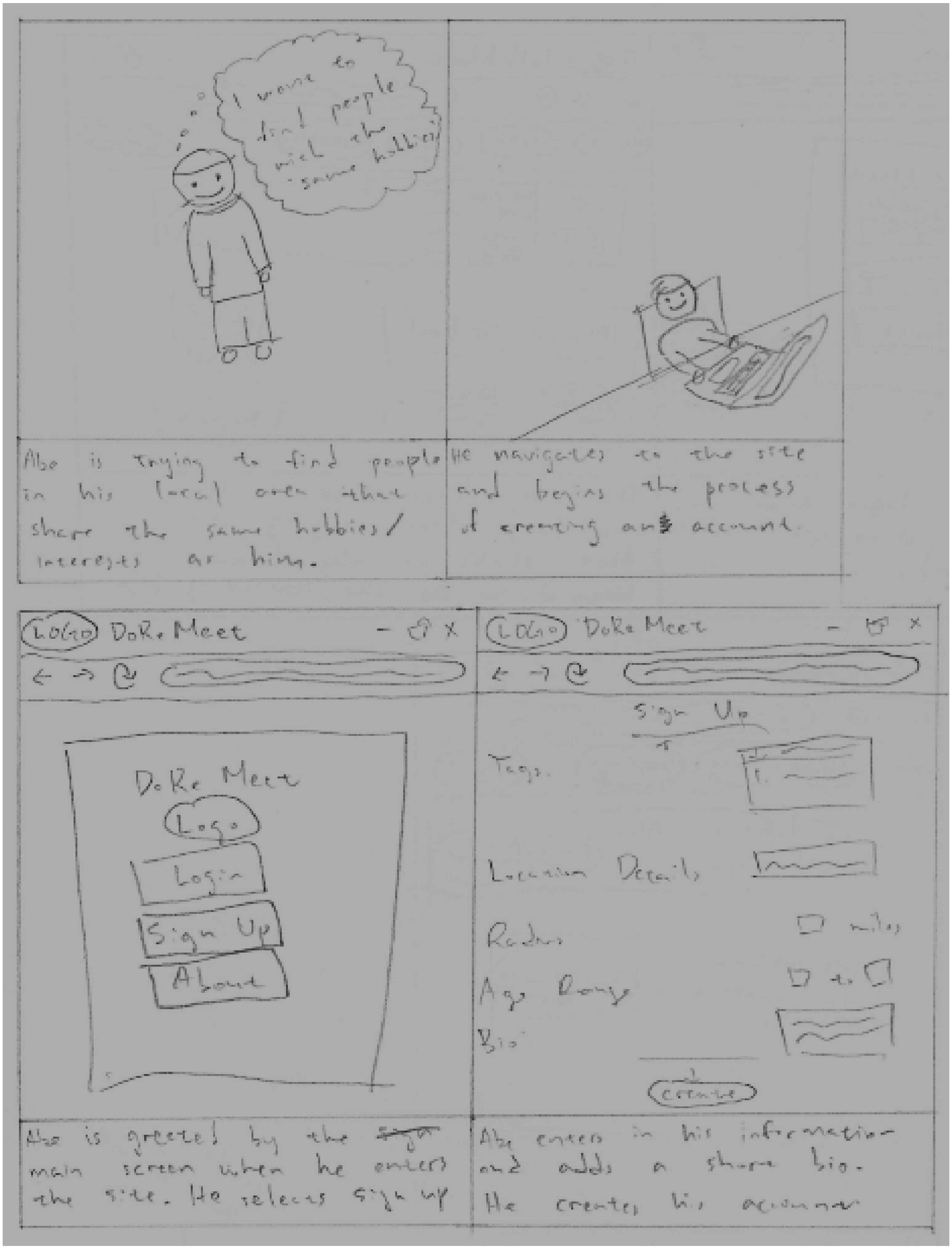
Ann is a returning user who has found a new passion for playing the bass. She attempts to log in and is asked to enter a code that has been emailed to her.

Ann enters the code that she recieved via email and is let into her account. She is asked to verify her information.

Ann notices that there is a meet online option. She selects it and now feels more comfortable that her location isn't shared

After finalizing her details she can then begin matching with other people who want to meet online.

Use Case 6: Account setup



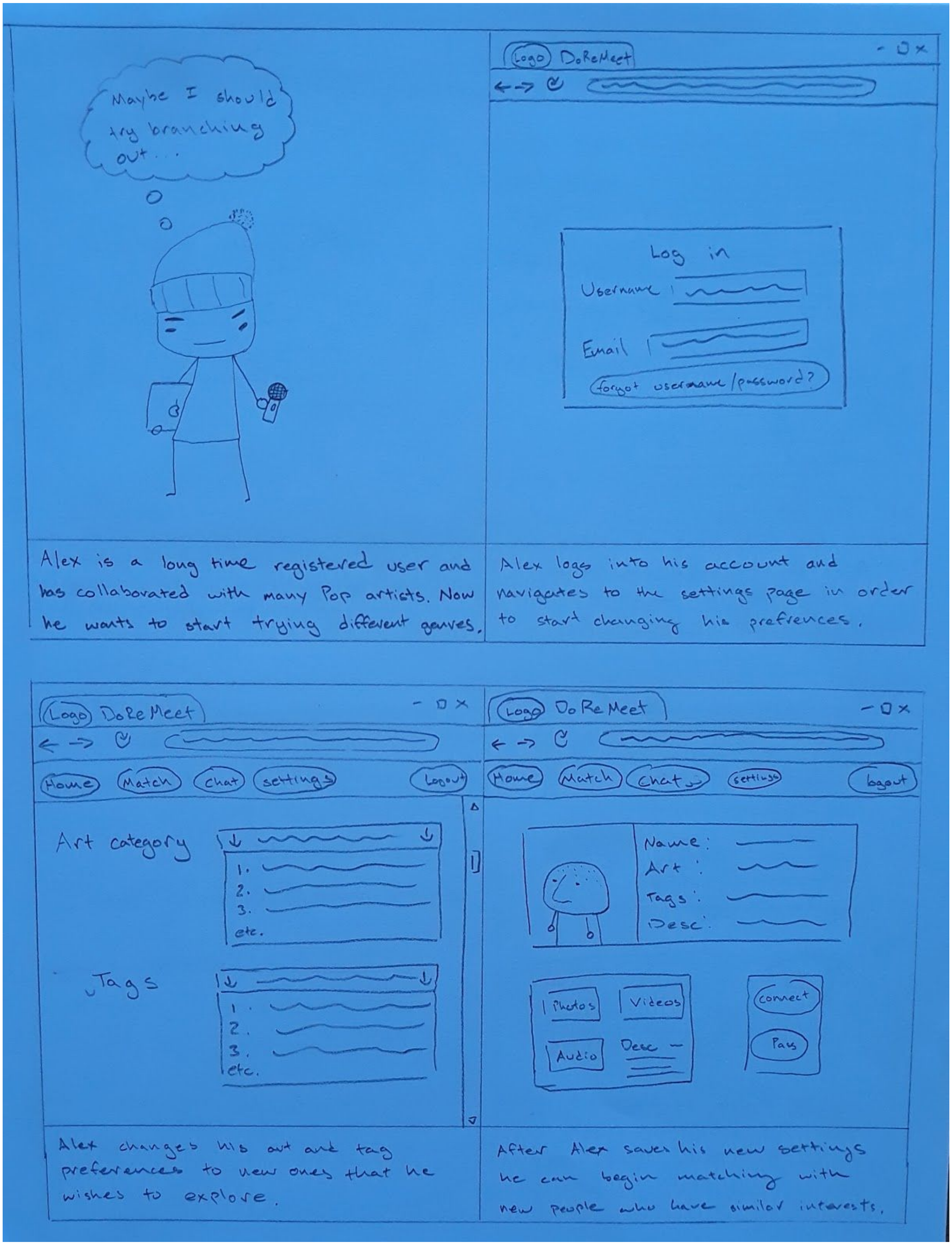
Abe is trying to find people in his local area that share the same hobbies/interests as him.

He navigates to the site and begins the process of creating an account.

Abe is greeted by the ~~sign~~ main screen when he enters the site. He selects sign up

Abe enters in his information and adds a short bio. He creates his account

Use Case 7: Updating preferences



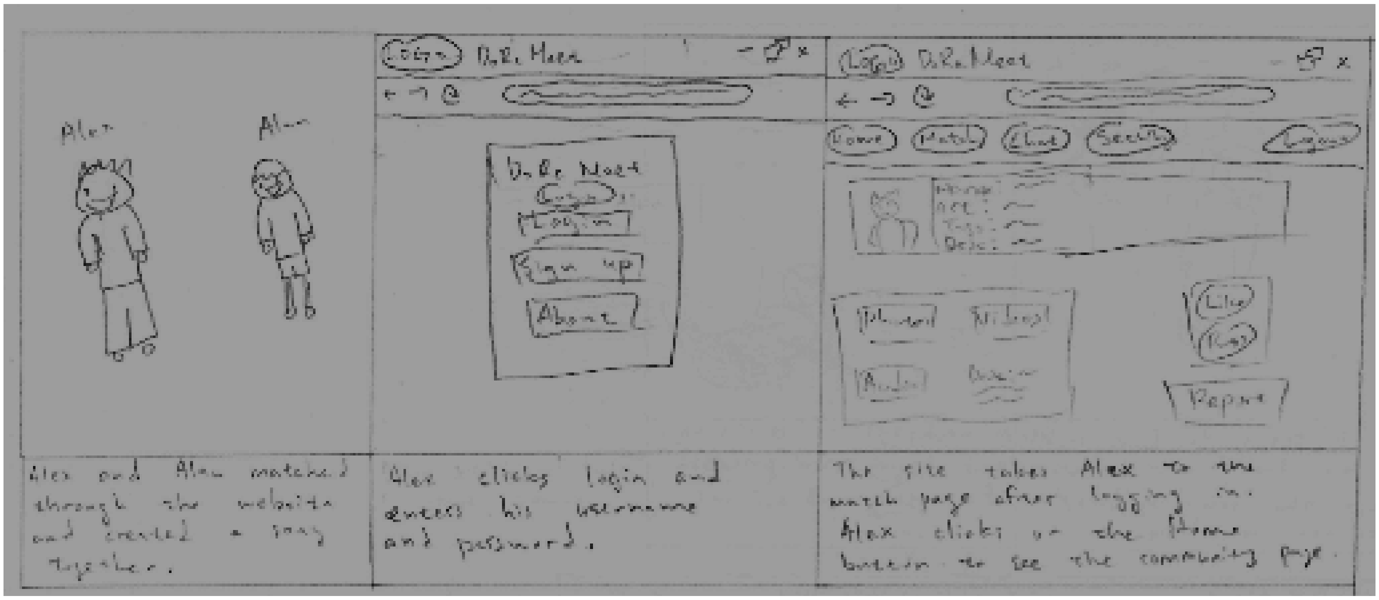
Alex is a long time registered user and has collaborated with many Pop artists. Now he wants to start trying different genres.

Alex logs into his account and navigates to the settings page in order to start changing his prefrences.

Alex changes his art and tag preferences to new ones that he wishes to explore.

After Alex saves his new settings he can begin matching with new people who have similar interests.

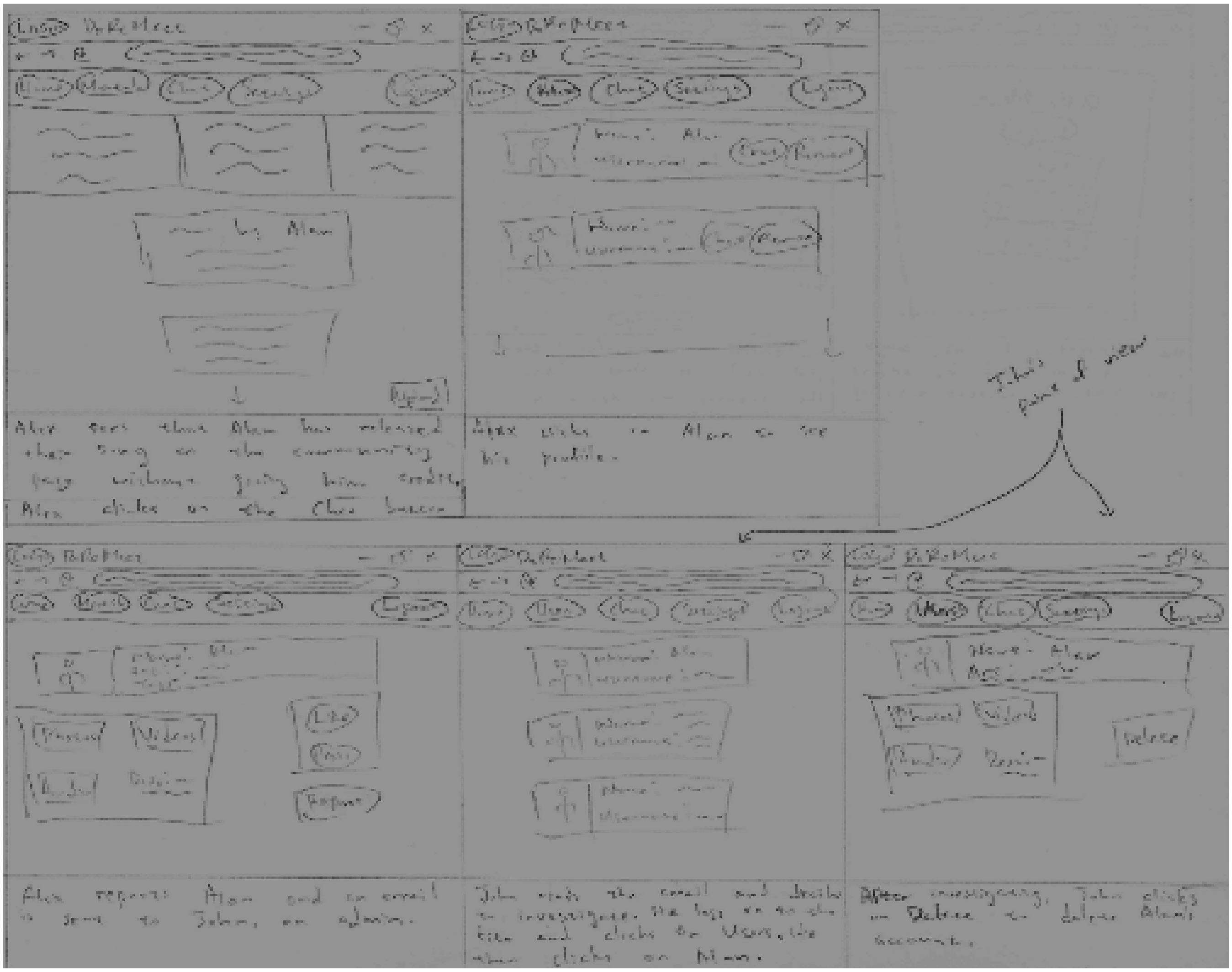
Use Case 8: Reporting a user



Alex and Alan matched through the website and created a song together.

Alex clicks login and enters his username and password.

The site takes Alex to the match page after logging in. Alex clicks on the Home button to see the community page.



Alex sees that Alan has released their song on the community page without giving him credit. Alex clicks on the Chat button.

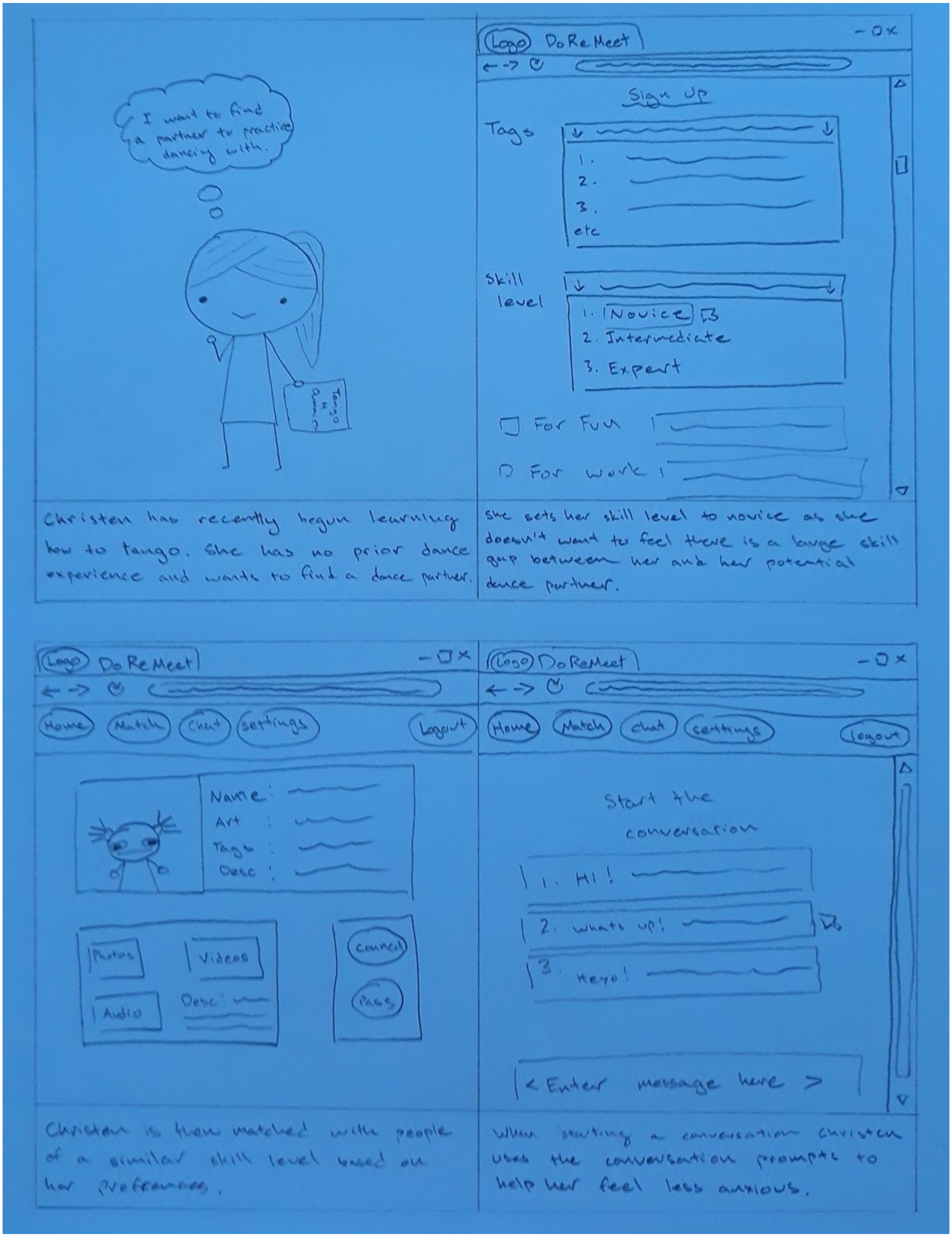
Alex clicks on Alan to see his profile.

Alex reports Alan and an email is sent to John, an admin.

John reads the email and decides to investigate. He logs on to the site and clicks on Users, he then clicks on Alan.

After investigating, John clicks on Delete to delete Alan's account.

Use Case 9: Set skill level and Conversation Prompts



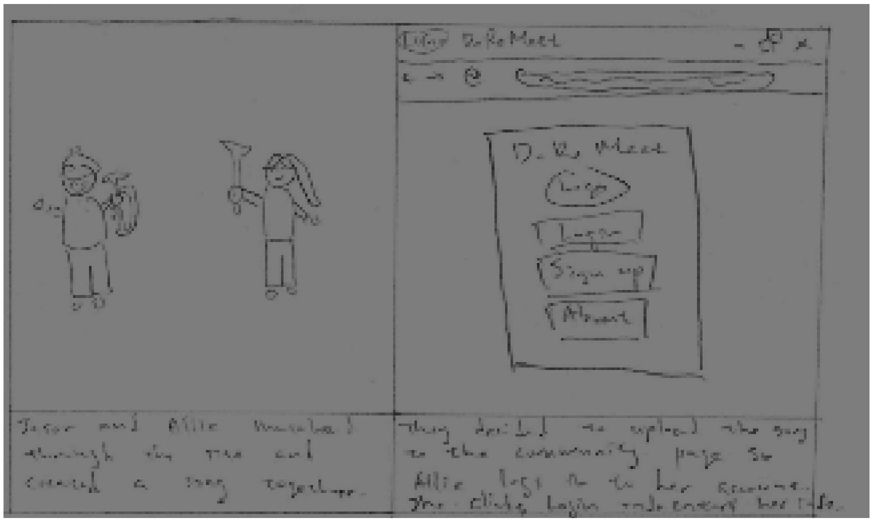
Christen has recently begun learning how to tango. She has no prior dance experience and wants to find a dance partner.

She sets her skill level to novice as she doesn't want to feel there is a large skill gap between her and her potential dance partner.

Christen is then matched with people of a similar skill level based on her preferences.

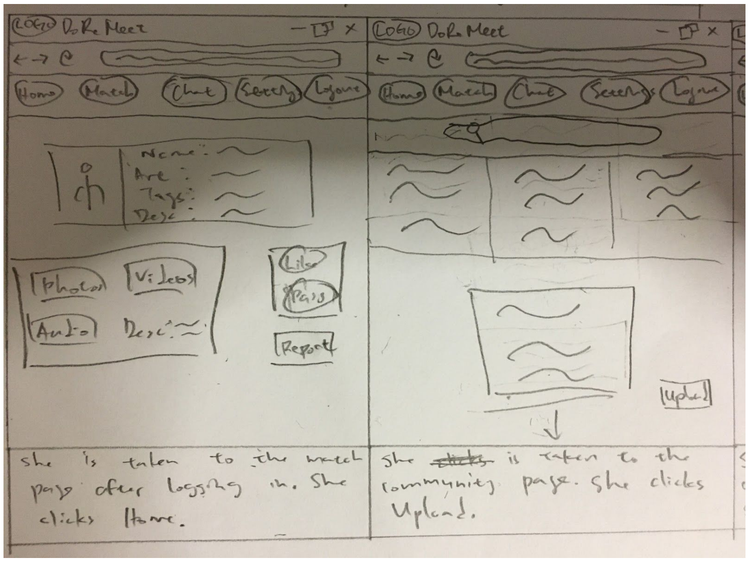
When starting a conversation christen uses the conversation prompts to help her feel less anxious.

Use Case 10: Submitting a project to the community highlights page



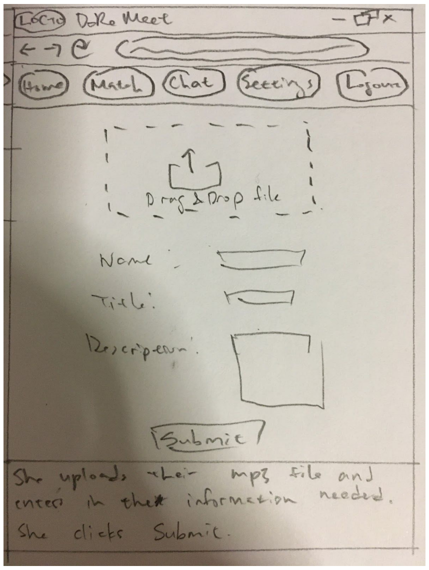
Jason and Allie musicked through the site and created a song together.

They decided to upload the song to the community page so Allie logs in to her account. She clicks login and enters her info.



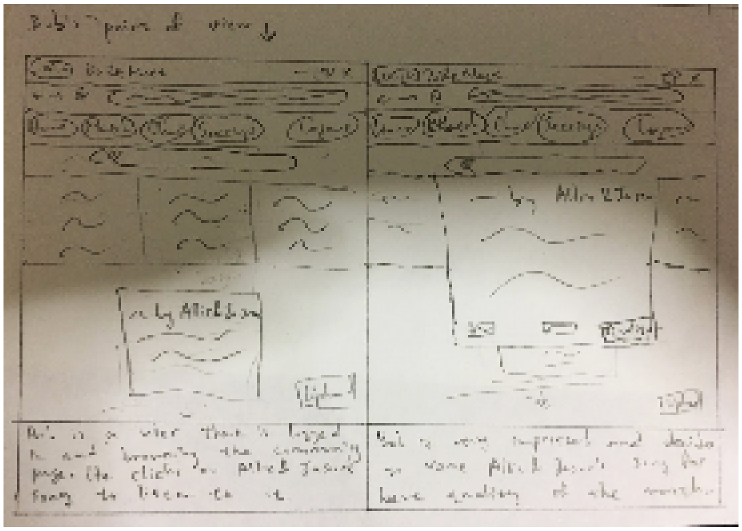
She is taken to the match page after logging in. She clicks Home.

She ~~clicks~~ is taken to the community page. She clicks Upload.



She uploads their mp3 file and enters in the information needed. She clicks Submit.

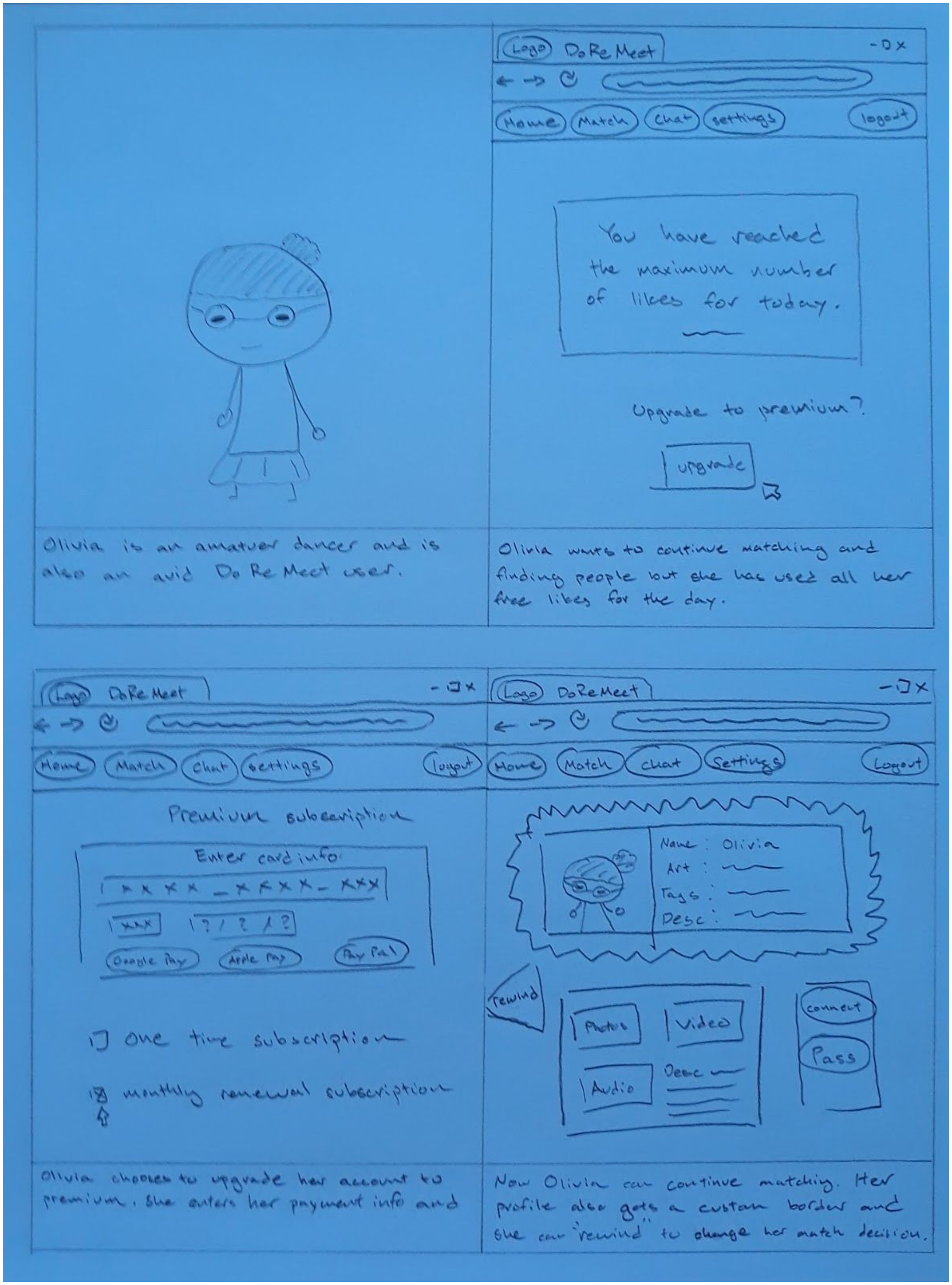
Bob's point of view ↓



Bob is a user that is logged in and browsing the community page. He clicks on Allie & Jason's song to listen to it.

Bob is very impressed and decides to name Allie & Jason's song for have [illegible] of the month.

Use Case 11: Premium features



Olivia is an amatuer dancer and is also an avid Do Re Meet user.

You have reached the maximum number of likes for today.

Upgrade to premium?

Olivia wants to continue matching and finding people but she has used all her free likes for the day.

Premium subscription

Enter card info:

xxxx_xxxx_xxx

xxx ?/?/?

□ one time subscription

☒ monthly renewal subscription

Olivia chooses to upgrade her account to premium. She enters her payment info and

Name: Olivia

Art:

Tags:

Desc:

Now Olivia can continue matching. Her profile also gets a custom border and she can "rewind" to change her match decision.

4. High level database architecture and organization

1. Database Organization

- **Business Rules**

A. Users (strong)

- A new user shall be able to create one and only one account.
- A registered user shall be able to register as only one account type.
- A user shall have a unique id.
- A user shall have a first name.
- A user shall have a last name.
- A user shall have a unique email address.
- A user shall define gender.
- A user shall have a birthdate.
- A user shall have a phone number.
- A user shall have an art category.
- A user shall have at least one tag.
- A user shall have a skill level.

B. User Address (strong)

- A user address shall have a unique id.
- A user address shall have a street.
- A user address shall have a city.
- A user address shall have a state.
- A user address shall have a country.
- A user address shall have a zip code.

C. Account

- An account shall be available to one and only one user.
- An account shall have one unique id.
- An account shall have a username.
- An account shall have an account type.
- An account shall have user id.
- An account shall have one encrypted password.
- An account shall have one account creation date.
- An account shall have account status.

D. Account Type

- An account type shall have one unique id.
- An account type shall have one description.

E. File Paths

- File paths shall have one unique id.
- File paths shall have a path for profile picture.
- File paths shall have a path for pictures.
- File paths shall have description for pictures.
- File paths shall have paths for videos.
- File paths shall have description for videos.
- File paths shall have a path for audio.
- File paths shall have description for audio.
- File paths shall have a path to their YouTube profile.
- File paths shall have a path to their Spotify profile.
- File paths shall have a path to their Instagram profile.
- File paths shall have a path to their website.

F. Preferences

- Preferences shall have one unique id.
- Preferences shall have interest on minimum age range.
- Preferences shall have interest on maximum age range.
- Preference shall have interest on gender preference.
- Preference shall have interest in the meeting category.
- Preference shall have interest on minimum location range.
- Preference shall have interest on maximum location range.
- Preference shall have user’s match interests on art categories.
- Preference shall have the user's interest on skill level.

G. Matches

- Matches shall have one unique id.
- Matches shall have a username.
- Matches shall have a match_status.

H. Chats

- Chats shall have a unique id.
- Chats shall have client username.
- Chat shall have a creation date.

I. Chat Messages

- Chat messages shall have a unique id.
- Chat messages shall have user_id.
- Chat messages shall have client username.
- Chat messages shall have file id.
- Chat messages shall have a creation date.
- Chat messages shall have text.

J. Chat File

- Chat files shall have a unique id.
- Chat files shall have a unique name.
- Chat file shall have a file path.

K. Community page

- Community page shall have a unique id.
- Community page shall have a user's name.
- Community page shall have a post title.
- Community page shall have a post field.
- Community page shall have post total likes.
- Community page shall have a post creation date.

- **Entities, their attributes, relationship, and domains at the high level**

A. Users (strong)

- **User_id**: key, numeric
- **First_name**: multivalue, alphanumeric
- **Last_name**: multivalue, alphanumeric
- **DOB**: multivalue, timestamp
- **Gender**: multivalue, alphanumeric
- **Username**: multivalue, alphanumeric
- **Email**: key, alphanumeric
- **Phone**: multivalue, alphanumeric
- **Art_category**: multivalue, alphanumeric
- **Tag**: multivalue, alphanumeric
- **Skill_lvl**: multivalue, numeric

B. User Address (strong)

- **User_add_id**: key, numeric
- **Street**: alphanumeric
- **City**: alphanumeric
- **State**: alphanumeric
- **Zipcode**: alphanumeric
- **Country**: alphanumeric

C. userAdd (weak)

- **Users**: weak key, numeric
- **User_address**: weak key, numeric

D. Account (weak)

- **User_acc_id**: key, numeric
- **Users**: weak key, numeric
- **Type**: weak key, numeric
- **Password**: multivalue, alphanumeric
- **Created**: composite, alphanumeric
- **Activate**: multivalue, alphanumeric

E. Account Type (weak)

- **User_accType_id**: key, numeric
- **User_description**: multivalue, alphanumeric

F. File Paths (weak)

- **filePath_id**: key, numeric
- **profile_pic_path**: multivalue, alphanumeric
- **pic_path**: multivalue, alphanumeric
- **pic_path_desc**: alphanumeric
- **video_path**: multivalue, alphanumeric
- **video_path_desc**: alphanumeric
- **audio_path**: multivalue, alphanumeric
- **audio_path_desc**: alphanumeric
- **profile_1**: multivalue, alphanumeric
- **profile_2**: multivalue, alphanumeric
- **profile_3**: multivalue, alphanumeric
- **profile_4**: multivalue, alphanumeric

G. Preferences (weak)

- **preference_id**: key, numeric
- **min_age**: numeric
- **max_age**: numeric
- **gender_pref**: alphanumeric
- **min_loc**: numeric
- **max_loc**: numeric
- **art_pref**: multivalue, alphanumeric
- **meeting_pref**: multivalue, alphanumeric
- **skill_lvl_pref**: numeric

H. Matches (weak)

- **match_id**: key, numeric
- **match_client_username**: weak key, alphanumeric
- **match_status**: multivalue, alphanumeric

I. Chats (weak)

- **chat_id**: key, numeric
- **matches**: weak key, numeric
- **chat_message**: weak key, numeric

J. Chat Messages (weak)

- **Chat_msg_id**: key, numeric
- **Chat_user_id**: weak key, numeric
- **Chat_msg_client_username**: weak key, alphanumeric
- **Chat_msg_file_id**: weak key, alphanumeric
- **Chat_msg_creation**: multivalue, timestamp

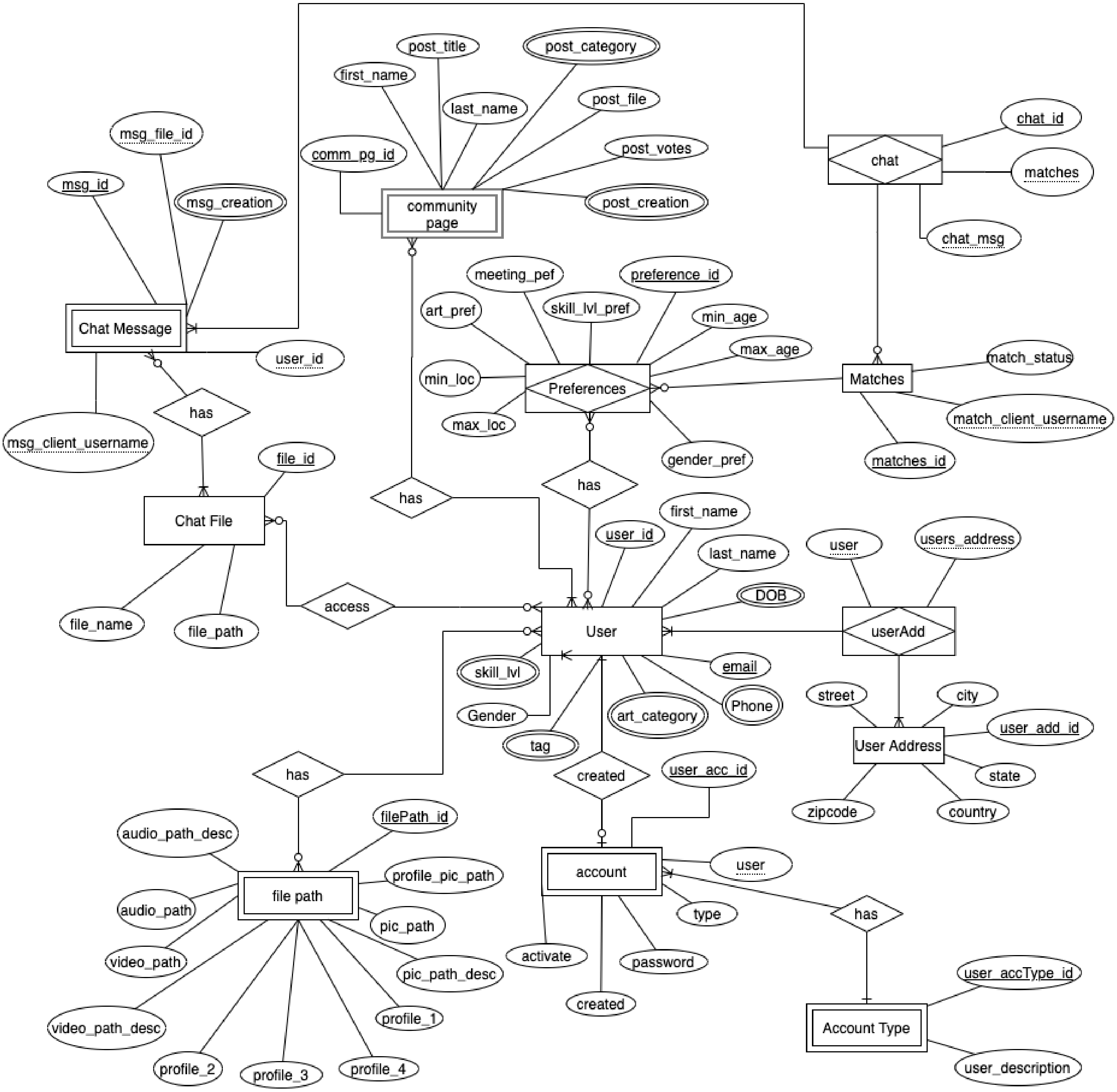
K. Chat File (strong)

- **Chat_file_id**: key, numeric
- **Chat_file_name**: multivalue, alphanumeric
- **Chat_file_path**: multivalue, alphanumeric

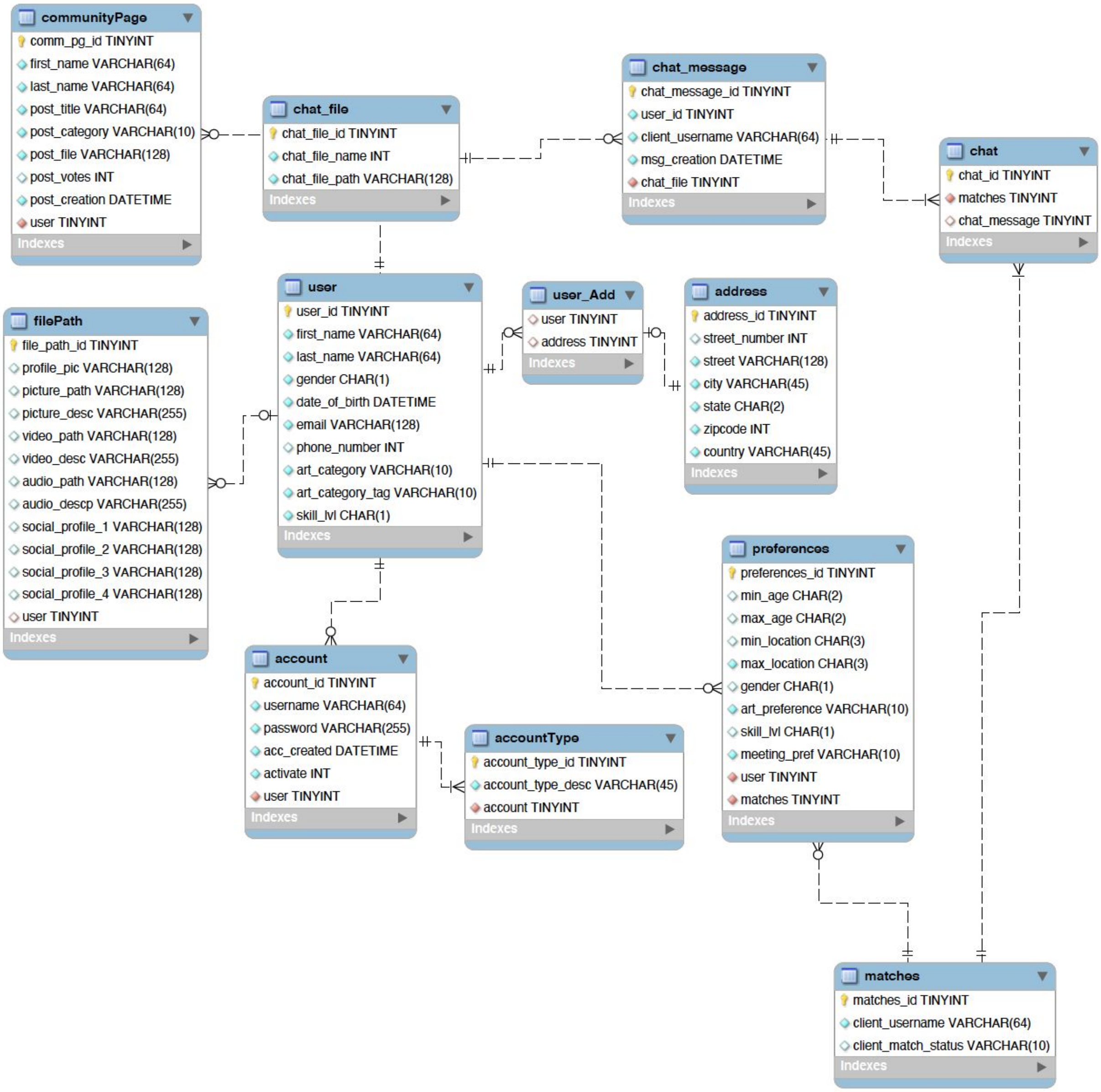
L. Community page (weak)

- **Comm_pg_id**: key, numeric
- **First_name**: multivalue, alphanumeric
- **Last_name**: multivallue, alphanumeric
- **Post_title**: multivalue, alphanumeric
- **Post_file**: multivalue, alphanumeric
- **Post_votes**: numeric
- **Post_creation**: multivalue, timestamp

- ERD Diagram


post_title
post_category
first_name
last_name
post_file
comm_pg_id
post_votes
community page
post_creation
chat_id
chat
matches
chat_msg
msg_file_id
msg_id
msg_creation
meeting_pef
preference_id
art_pref
skill_lvl_pref
min_age
max_age
Chat Message
user_id
match_status
min_loc
Preferences
Matches
max_loc
match_client_username
has
msg_client_username
gender_pref
matches_id
file_id
has
has
Chat File
first_name
user_id
last_name
user
users_address
DOB
access
file_name
file_path
User
userAdd
skill_lvl
email
Phone
Gender
art_category
street
city
user_add_id
tag
User Address
has
created
user_acc_id
state
zipcode
country
filePath_id
audio_path_desc
account
user
profile_pic_path
file path
audio_path
type
has
pic_path
video_path
activate
password
pic_path_desc
user_accType_id
profile_1
created
video_path_desc
Account Type
profile_2
profile_3
profile_4
user_description

- **EER Diagram (made using MySQL Workbench)**



- **DBMS**

We will be using **MySQL workbench** to create the database of our website, and there are many reasons to use MySQL workbench.
Actions will be directly carried out on the tables; it is also free and open source. It also has enhanced entity-relationship modeling support. Lastly, it is compatible with multiple operating systems, such as windows, Linux, and OS X.

2. Media Storage

For the media/file storage we have decided to use/store them in a **file system**.
We made this decision as a team after doing some research on the media storage. Our website allows each user to upload multiple photos, videos and media files (upto 80mb for free account and 120mb for premium account) as a primary form of showing their work. Storing media files in a database will cause load on the database, and negatively impact the performance, which we do not want so we chose to store media files in a file system.
In the database system, we will have a path stored which will take/help users to access the media storage. Using these file paths the user will be able to access their media files easily and much quickly.

3. Search/filter architecture and implementation

Our product's searching system is fairly simple, wherein the website allows users to search the people they are connected to. With the help of a search algorithm, users will be able to search people in their match list only, by their names, or by their usernames.

In addition to letting people search the list of people they are connected with, users also have the ability to search the community page for posts based on the username of the person making the post or the art category they want to see the post in.

5. High Level APIs and Main Algorithms

APIs to be used

Location API: For matching 2 registered users together based on their location, we will be using locationIQ (https://locationiq.com/) API, which will let us accurately present the user with valid matches that fits their location search criteria. For our website, we will have a search radius preference for matching two users together. Based on the radius specifications set by the user, we will present them with potential matches to pick and choose from.

Encryption API: Encryption should encrypt a user's password before it is stored in the database to secure the information. The password will be stored in a database in encrypted form, and will not be accessible to anyone. When a user tries to login a user's account, the Encrypt API will fetch the encrypted password, decrypt to see if it matches with what user types in. We will be creating our own API for encryption and not using any 3rd party libraries/APIs to uphold security standards.

Authentication API: Authentication should verify a user's identity using a unique identifier. A unique identifier is a set of a valid email-address as an User ID and associated password that user input when they registered the site. We are also planning to implement two factor authentication and OAuth 2.0 to enforce stronger security measures. Using these standard authorization practices, we will try to ensure that the users' account is secure and cannot be breached easily.

Payment API: We will also be implementing a payment API, since we will need to securely take payment information from the users' who opt to enroll in a premium account. We are still debating between using a 3rd party APi vs building our own API. While third party APIs will already have the security aspect built in, it is a risk since we will be ultimately sharing sensitive user information with an outside company. Also most third party APIs are paid (we did find a free one: Noodlio-Pay and are testing its integrity) so we are not sure if we can go that route. With implementing our API, we need to ensure that it is secure, all the information is encrypted, along with making sure that the API is compatible for at least all the major cards/payment methods.

Algorithms

Sorting: Given the expected amount of matches a user can have. We will rely on insertion sort for organizing the list of matches a user has. The amount of likes will also be sorted in this manner.

Matching: Matching will be done based mostly on preferences selected by the user. We will prioritize search results based on the meeting criteria first (in person or online). If it is a meet-in-person preference, we will then filter the results based on the minimum and maximum radius set by the user. Next, any tags that the user has listed will be prioritized and we will present results based on the custom tags/requirements set by the user. If a user has a premium account, they will have priority when being matched by other users, meaning that they will show up first and their highlighted profiles will gain more traction.

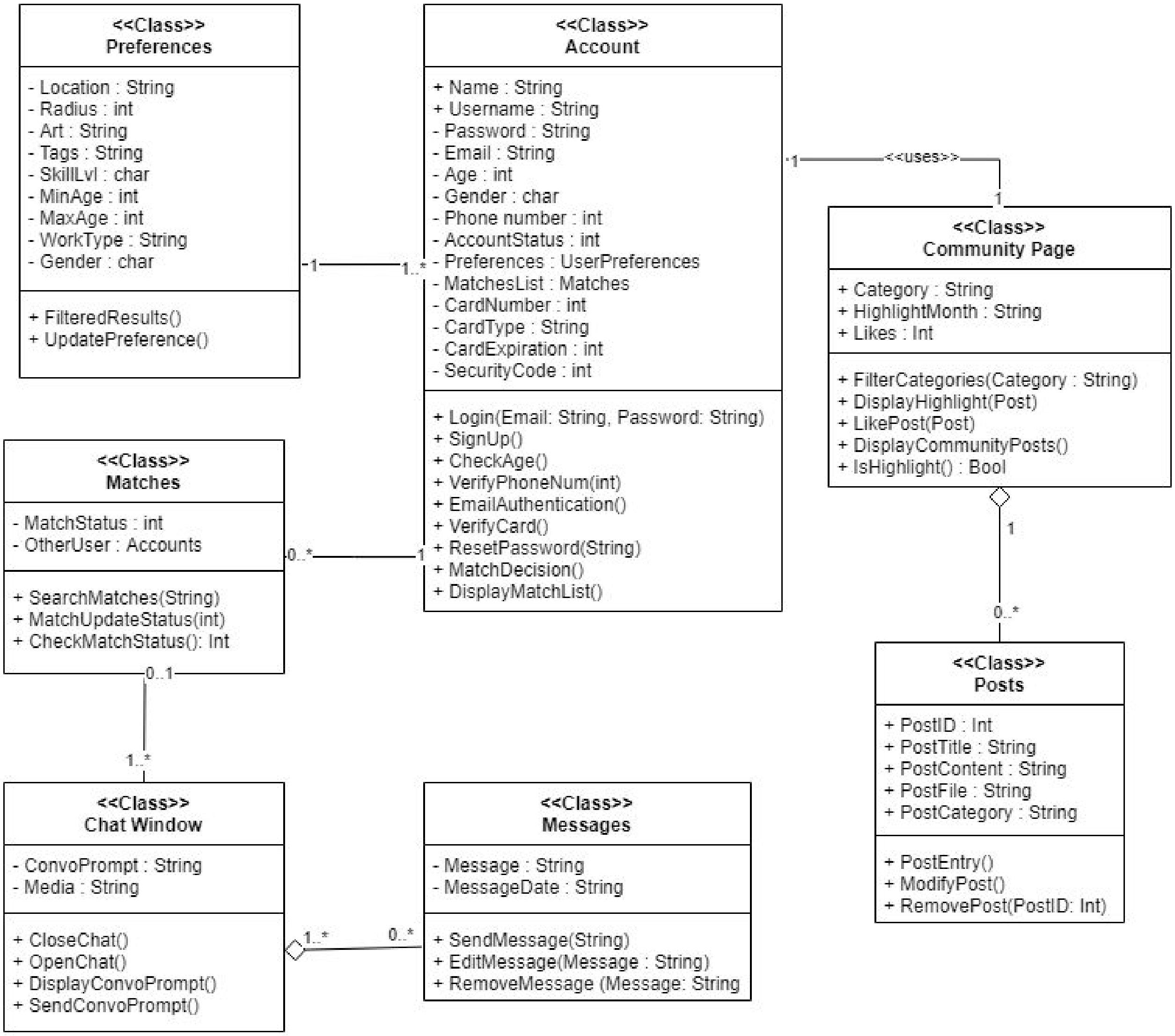
Searching: Our product provides users only relevant results without hassle. To provide these experiences, the product automatically filters out search results by user's predefined references such as, prefer art area or skill level of the art, and geographic information. Users will only get the result around the user's area. The only search users can do is within the list of people they are already matched with.

Ranking: Our website has a feature that highlights the most popular post that users have created with whomever they were matched with. The highlighted post will be chosen based on the amount of "likes" the post has. These "likes" will have a weight tied to them. The weights of these "likes" will degrade over time at certain time intervals.

Changes to Stack:

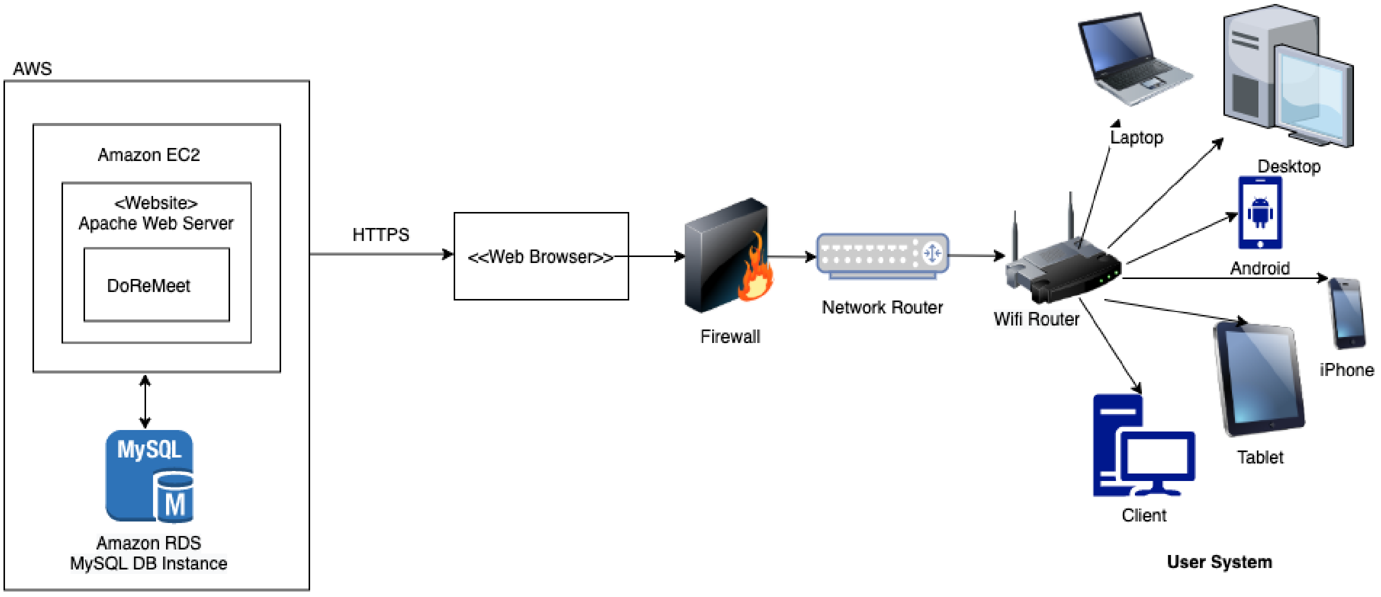
Instead of using React framework, we are using **Express** framework for our backend.

6. High Level UML Diagrams

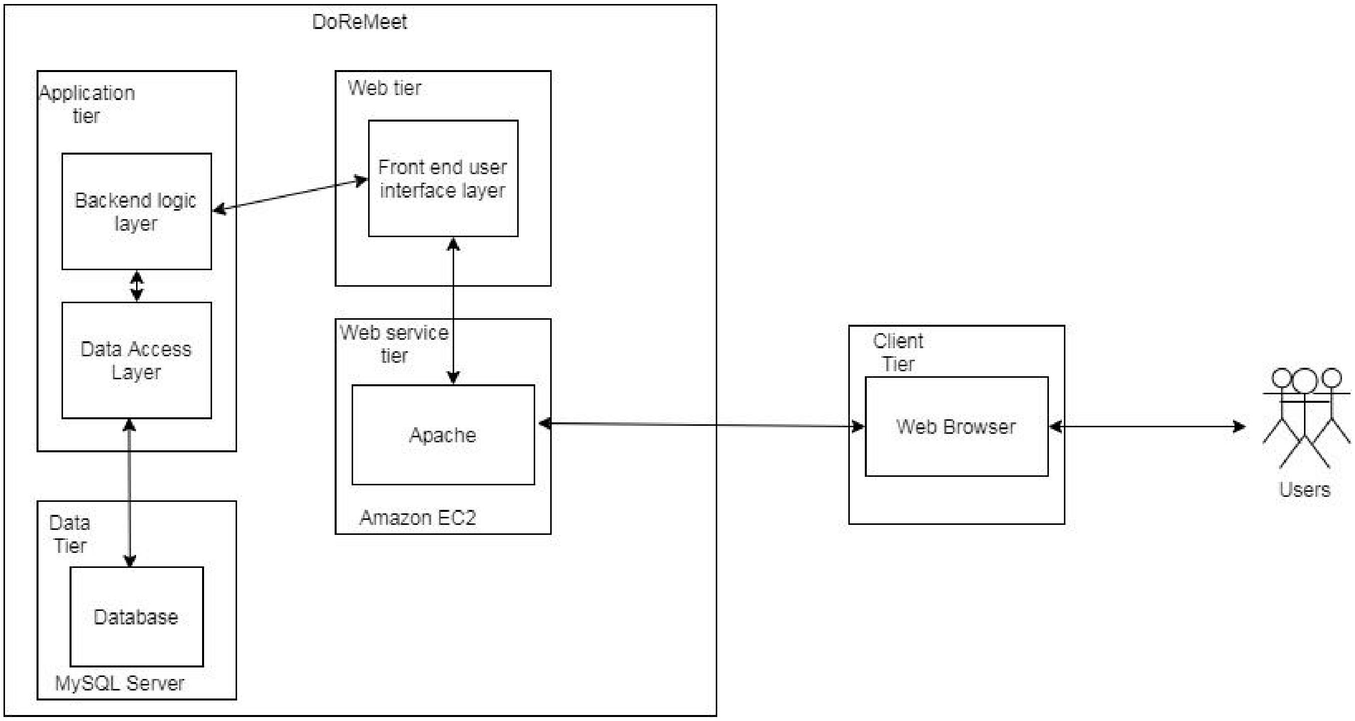


7. High Level Application Network and Deployment Diagrams

Network Diagram



Deployment Diagram



8. Key risks

Skills risks

- Lack of experience in UI/UX design.
  Our team is fairly new to building a whole product from scratch and the limited experience in terms of building good UI/UX design is a risk that will likely create roadblocks for us..
  Resolution: We aim to utilize good practices and look to other sites for inspiration when it comes to site design.
- Lack of experience in the database and using API.
  Similar to UI/UX, we have limited experience in implementing 3rd party APIs and in creating our own APIs. Also, there is only one person on our team who has some knowledge of databases, so navigating that and building an effective database will be a bit of a challenge.
  Resolution: To overcome this limitation, we aim to learn in class from Professor Ortiz Costa, and utilize online learning materials (video, documentation, learning websites, etc.)

Schedule risks

- Our goals for DoReMeet might be a bit lofty, and we may be overestimating our own abilities and may not be able to deliver on all of our priority 1 requirements.
  Resolution: Properly manage out time and our team in order to maximize productivity and accomplish all the things we want to get done.
- Our timeline is too tight to implement our product’s priority 3 level requirements, so we are worried if we’ll get to test our product properly.
  Resolution: We think we might need to come up with a few edge cases and start implementing tests at an early stage in our development process.

Technical risks

- Relying on external API’s might present issues in a lack of needed functionality or unforeseen bugs.
  Resolution: Properly research all APIs before committing to using them, and using only stable versions to prevent any unforeseen bugs.

Teamwork risks

- Loading too much work on a single teammate could lead to a backlog of work.
  Resolution: Make sure to properly distribute work and keep schedules in mind.
- We are doing a team project in a virtual environment. Remote situations give us more time to work, but at the same time, it makes it hard to focus on the task.
  Resolution: Make sure to prioritize tasks and deload every work to every member. Use a task manager (Monday) to keep tracking small tasks and share the progress with other members. We are trying to prioritize communication within the team, to make sure that everyone is on the same page about all our work.

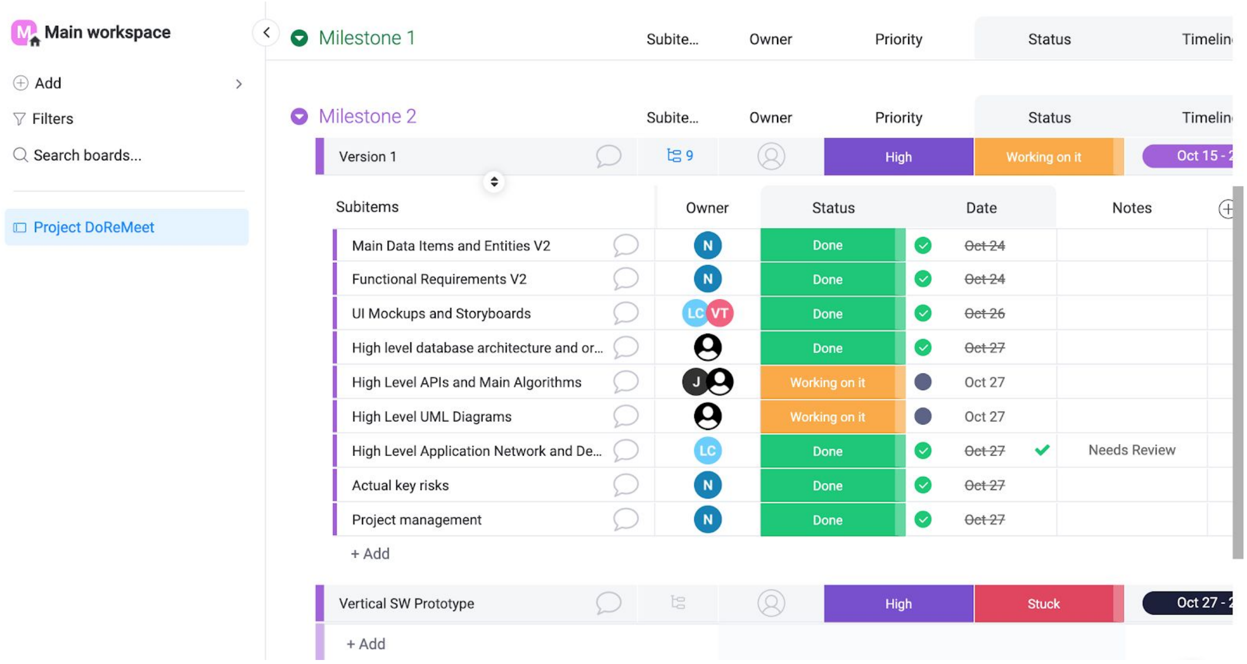
Legal/Content risks

- We are handling user's location information, potentially even their credit card information (for premium users) and other personal information. There might be a chance people misuse our application with ulterior motivation, and try to cause harm to someone. *Resolution*: We need to be extra careful with user's information, and we will give the users' ability to choose what information they want to share and not share.

9. Project management

For our product we are choosing to use monday.com.
We've set up our Project Board and use it to split tasks, assign work, and update the status of the work. Also, we are using Monday.com to keep track of deadlines and the timeline for our project, along with categorizing the work based on priorities.

This platform will help us keep track of our own progress, and see what stuff our team members are working on. It'll help us all communicate better and see the overall progress of the team.



Milestone 1	Subite...	Owner	Priority	Status	Timelin

Milestone 2	Subite...	Owner	Priority	Status	Timelin
Version 1	9		High	Working on it	Oct 15 - 2

Subitems	Owner	Status	Date	Notes
Main Data Items and Entities V2	N	Done	✓ ~~Oct 24~~	
Functional Requirements V2	N	Done	✓ ~~Oct 24~~	
UI Mockups and Storyboards	LC VT	Done	✓ ~~Oct 26~~	
High level database architecture and or...		Done	✓ ~~Oct 27~~	
High Level APIs and Main Algorithms	J	Working on it	● Oct 27	
High Level UML Diagrams		Working on it	● Oct 27	
High Level Application Network and De...	LC	Done	✓ ~~Oct 27~~	✓ Needs Review
Actual key risks	N	Done	✓ ~~Oct 27~~	
Project management	N	Done	✓ ~~Oct 27~~	
+ Add				

Vertical SW Prototype			High	Stuck	Oct 27 - 2
+ Add					

10. Team Contributions

Member	Role	Contributions
Nimiksha	Team Lead & Documentation Lead	Documentation: Main Data Items and Entities Functional Requirements Database architecture (Review only) High level APIs and Algorithms High Level UML Diagram (Design) Key Risks Project Management Vertical Prototype: Code Review
Meet	Backend and Database Lead	Documentation: Main Data Items and Entities Functional Requirements High level database architecture (Design and Implementation) ERD, EER Diagram, DBMS, Media Storage and Search/filter architecture High level APIs and Algorithms High Level UML Diagram (Design) Key Risks Vertical Prototype: Database implementation
Luke	Frontend Lead	Documentation: Main Data Items and Entities Functional Requirements UI Mockups and Storyboard (General Flow and 6 use cases) High level APIs and Algorithms High Level UML Diagram (Design) Network and Deployment Diagrams Key Risks Vertical Prototype: Frontend code Backend to Frontend connection
Mike	GitHub Master & Backend Engineer	Documentation: Main Data Items and Entities Functional Requirements High level APIs and Algorithms High Level UML Diagram (Design and Implementation) Key Risks Vertical Prototype: Backend Code (search from database and insert to database) Backend to Frontend connection
Jungsun	Database & Backend Engineer	Documentation: Main Data Items and Entities

		Functional Requirements High level APIs and Algorithms High Level UML Diagram (Design) Key Risks Vertical Prototype: Backend Code (connect to database, insert into database)
Vincent	Server Admin & Frontend Engineer	Documentation: Main Data Items and Entities Functional Requirements UI Mockups and Storyboard (5 use cases) High level APIs and Algorithms High Level UML Diagram (Design) Key Risks Vertical Prototype: Connect backend to frontend Search and Insert to Database

The team worked on all sections of Milestone 2 together during Zoom meetings, and collaboratively created this report.