

Frindr - Product Backlog

Haris, CJ, Mayur, Aeson, Joe

Problem Statement

Create an application that allows users from all around the world to connect and create new friendships with people located in their area with similar interests and hobbies.

Background Information

Moving to a new place may be hard sometimes, and finding friends can't always be the easiest thing in a new place. That's where Frindr comes in, this app allows people to be able to find new friends based on mutual interests and location. Users will be able to enter their interests and location. Using a match finding algorithm we will bring the most similar users together. Once they have been connected they can begin chatting. If the chat doesn't go well, then either of the users can "unmatch" the other and cause the chat to end. If the chat goes well then these two users could potentially have a new friendship. The goal of Frindr is to connect awesome people with other awesome people. Now moving to a new city, and finding new friends doesn't seem that bad.

Environment

We want to make sure that this website will be able to run on as many computers as possible. Therefore, we have chosen to use commonly used frameworks so that as many browsers as possible will be able to access and use the website. We will be using React, Node, and Javascript, which are all commonly used frameworks for web development. We will use MySQL for the database and we will be hosting the server and database using Digital Ocean.

Functional Requirements

Backlog ID	Functional Requirement	Hours	Status
1	As a user, I would like to create an Frindr profile	2	Sprint 1
2	As a user, I would like to login to my account (using email)	5	Sprint 1
3	As a user, I would like to be able to edit my profile	3	Sprint 1
4	As a user, I would like the Ability to add a bio	3	Sprint 1
6	As a user I would like the Ability to edit a bio	1	Sprint 1
7	As a user I would like the Ability to add interests	3	Sprint 1
8	As a user I would like to Remove an interest	1	Sprint 1
9	As a user I would like to Edit interests	1	Sprint 1
10	As a user I would like to Add characteristics	3	Sprint 1
11	As a user I would like the Ability to edit characteristics	1	Sprint 1
12	As a user I would like to be able to Remove characteristics	1	Sprint 1
13	As a user i would like the ability to delete my profile	3	Sprint 1
14	As a user i would like the ability to edit my name	1	Sprint 1
15	As a user i would like the ability to add my location	3	Sprint 1
16	As a user i would like the ability to edit my location	1	Sprint 1
17	As a user i would like the ability to hide myself from matching process	2	
18	As a user i would like the ability to enable email notifications	2	
19	As a user i would like the ability to disable email notifications	1	
20	As a user I would like to have a layout to find friends	3	

21	As a user I would like to be matched with random people based on our info	4	Sprint 1
22	As a user I would like to display a profile picture	1	
23	As a user I would like to upload my own photos	4	
24	As a user I would like to have a place to view my photos	3	
25	As a user I would like to be able to remove my photos	2	
26	As a user, I would like to add a chat	3	
27	As a user, I would like to 'like' a chat	2	
28	As a user, I would like to unlike a chat	1	
29	As a user, I would like to remove a chat	3	
30	As a user, I would like to be able to change my password	2	
31	As a user, I would like to be able to change the email associated with my account	2	

Non-Functional Requirements

Architecture

The frontend of the site will be created using React. It will connect through a server to the SQL database. Both the server and the database will be hosted through DigitalOcean.

Security

Security is a very important concern for both the performance of our application and privacy of our end users. Users will be able to securely login into our application through login API implemented into our application, and all of our passwords will be stored with encryption. Our web application will also prevent common vulnerabilities to our database such as SQL injections.

Usability

Frindr will include a very intuitive user experience with user's having the ability to seamlessly register for an account and begin matching with other users within minutes of signing up.

Performance

This web application will be hosted on a remote server provided through digital ocean and will be connected to a sql database, which will be able to handle thousands of requests to our application a minute. This application will require access to the internet. The back-end will be able to hold user data for 10,000 users. The queries that the user will be able to make from the application will take approximately 400 ms to receive a response.

Use Cases

Case 1:

Create a Frindr account (Profile)

Action	System Response
1. On the landing page click create profile	2. Display sign up page
3. Enter Email / Password /username	4. Send request to Backend, verify on frontend
	5. Backend will handle account creation, and will then log the user in using those credentials.
	6.Home page will be loaded
7.Homepage will be shown to user	

Case 2:

Login to account

Action	System Response
1.On the landing page click log in	2.Display the login page
3.User enters their already created login information	4. Info is sent to back end to verify that the user with those credentials exists

	5. Home page for specific user will be loaded
6.Home page for the user will be displayed	

Case 3:

Edit my profile

Action	System Response
1.Click my profile at top of page	2. Expand profile options drop down
3.Click on edit profile	4. Sends request to back end for all user information
5. A page will be displayed with all the users information	6. Information from back end is displayed on the edit profile page
7.Click a field(i.e name, age) to edit it	8. If a user clicks on an editable field, it can be changed in place
9. Click "confirm button once all changes have been made	10. Send all of the new data to the backend so that the database can be updated to reflect the new user information

Case 4:

add a bio

Action	System Response
1.Navigate to the edit profile page	2.necessary data will be loaded, and then displayed on the edit page
3.Click add Bio	4.A text box will appear and then the user can enter text
5.User adds bio	6.The program will limit the bio to 200 words, after that typing will not be allowed.
7.User click confirm	8.Send the information to the back end to be added to the data base

Case 6:

edit a bio

Action	System Response
1.Navigate to the edit profile page	2.necessary data will be loaded, and then displayed on the edit page
3.Click edit Bio	4.The bio text box will not become an editable text box as opposed to a static one
5.User edits bio	6.The program will limit the bio to 200 words, after that typing will not be allowed.
7.User click confirm	8.Send the information to the back end to modify the data base

Case 7:

add interests

Action	System Response
1.Navigate to the edit profile page	2.necessary data will be loaded, and then displayed on the edit page
3. Click add interest	4. two new editable text boxes will appear, for the user to add information
5.Add interest name and description	6.One textbox is for the name of the interest, the other textbox is for the description. Name and interest fields will be limited to 240 characters
7. Click confirm	8.The data from the textboxes will be sent to the backend, to be added to the user profile

Case 8:

Remove an interest

Action	System Response
1.Navigate to the edit profile page	2.necessary data will be loaded, and then displayed on the edit page
3.Click remove interest next to a specific interest	4.Delete the interest from the database for the user profile

Case 9:

Edit interests

Action	System Response
1.Navigate to the edit profile page	2.necessary data will be loaded, and then displayed on the edit page
3. Click edit interest on a specific interest	4. Two editable text boxes will appear(populated with old info), for the user to change information
5.edit interest name and description	6.One textbox is for the name of the interest, the other textbox is for the description. Name will be limited to 240 characters
7. Click confirm	8.The data from the textboxes will be sent to the backend, to replace the old data

Case 10:

Add characteristics

Action	System Response
1.Navigate to the edit profile page	2.necessary data will be loaded, and then displayed on the edit page
3. Click add characteristics	4. an editable text box will appear, for the user to add the characteristic
5.Add characteristic name	6.Textbox is to type in the name of the characteristic, will be limited to 240 characters
7. Click confirm	8.The data from the characteristic will be sent to the backend, to be added to the user profile

Case 11:

edit characteristics

Action	System Response
1.Navigate to the edit profile page	2.necessary data will be loaded, and then

	displayed on the edit page
3. Click edit characteristics on a specific characteristics	4. an editable text box will appear(populated with old info), for the user to edit the characteristic
5.Add characteristic name	6.User edits the name of the characteristic
7. Click confirm	8.The data from the characteristic edit will be sent to the backend, to modify the user profile

Case 12:

Remove characteristics

Action	System Response
1.Navigate to the edit profile page	2.necessary data will be loaded, and then displayed on the edit page
3.Click remove characteristic, next to a specific characteristic	4.Delete the characteristic from the database for the user profile

Case 13:

Delete Frindr Profile

Action	System Response
1. Delete button	2. System deletes profile

Case 14:

Edit profile name

Action	System Response
1. Text field on front-end to change name	2. Field populates on front end as user types
3. Submit button	4. Database updated with new name for the current user

Case 15:

Add location

Action	System Response
1. Location text-box on registration page	2. Field populates as user types
3. User presses register button	4. Backend creates user profile and sets location for the user

Case 16:

Edit location

Action	System Response
1. Text field requesting location	2. Text field fills as user types
3. Submit location change button	4. User location updated in database and success message delivered on front-end

Case 17:

Hide from matching process

Action	System Response
1. Toggle hide button	2. Toggle turns off (front-end)
3. Submit form	4. User is marked to be hidden in database and success message delivered on front-end

Case 18:

Enable email notifications

Action	System Response
1. Toggle on email notifications	2. Toggle turns on (front-end)
3. Press submit on the form	4. Notifications enabled for current user

	in database and success message shown on front-end
--	--

Case 19:

Disable email notifications

Action	System Response
1. Toggle off email notifications	2. Toggle turns off on front-end
3. Press submit on the form	4. Notifications disabled in database for current user and success msg prompted on front-end

Case 20:

Page for finding friends

Action	System Response
Start	Show picture and bio of first person
Skip button	Show picture and bio of another person
Friend button	Sets interested status on target to user
Profile button	Shows full profile of target

Case 21:

Matching algorithm

Action	System Response
Request from skip or start	Compiles information about a user (tags, location) and puts people inside an accepted range for that user, with the requirements getting looser as time goes on, randomly selecting people from the list (but not the same people)

Case 22:

Profile picture

Action	System Response
Information shown	When displaying a user to others, this picture will be shown
Pressing picture	Will re-route to profile page

Case 23:

Upload photos

Action	System Response
Upload	Will open file manager and get options for uploading a photo
Submit	Will send the photo to it's storage location

Case 24:

Place to view photos

Action	System Response
Profile	Will have tab for photos
(photos)	Will show a certain number of photos, and will have an option to display them or not

Case 25:

Remove Photos

Action	System Response
Remove	Will ask for confirmation then delete photo, page will then refresh without that photo

Case 26:

Add a chat (to a match)

Action	System Response
Tap on matches page	Page changes from home page to matches page

Click on matched profile	Page changed to matched user profile chat page
Click chat dialogue box, type, press enter	Chat will show up on dialogue box

Case 27:

Like a chat (to a match)

Action	System Response
Tap on matches page	Page changes from home page to matches page
Click on matched profile	Page changed to matched user profile chat page
Hit like button next to chat message	Chat will show “like” symbol next to it

Case 28:

Unlike a chat (to a match)

Action	System Response
Tap on matches page	Page changes from home page to matches page
Click on matched profile	Page changed to matched user profile chat page
Hit unlike button next to chat message	Chat “like” symbol next to it will disappear

Case 29:

Remove a chat (to a match)

Action	System Response
Tap on matches page	Page changes from home page to matches page

Click on matched profile	Page changed to matched user profile chat page
Hit remove button next to chat message	Chat will disappear

Case 30:

Changing a password

Action	System Response
1. Navigate to the profile page and click the "change password" button.	2. Switch pages to the password changing page.
3. User enters a new password into the field.	4. Backend updates with new password.
5. User is returned to the home page.	

Case 31:

Changing the associated email.

Action	System Response
1. Navigate to the profile page and click the "change email" button.	2. Switch pages to the email changing page.
3. User enters a new email into the field.	4. Backend updates with new email.
5. User is returned to the home page.	