Orbital 2022 Milestone 2 Report

Team Name: W

Team Members: Farrel Dwireswara Salim & Elbert Benedict

Proposed level of achievement: Artemis

Application name: Chattoku

Poster: Link

Video: Link

Chattoku GitHub repository: Link

Motivation

As an anime lover, it is not rare for us, members of team W, we often found ourselves in a situation where we want to find a new anime to watch, but do not know where to find a good anime which suits our preference. We also often found ourselves to feel lost regarding trending topics of anime because we don’t know who to discuss it with. There are some who have tried to do this such as Reddit and MyAnimeList. However, both MyAnimeList and Reddit do not have a feature where anime lovers can find a new anime which is specially tailored for them. Moreover, MyAnimeList, does not have a mobile application for it’s forum while Reddit is not specially made for anime lovers, which sometimes cause us to feel a bit lost. Considering all of this, our team decided to create Chattoku (Chat + Otaku), a platform where fellow anime lovers can both interact with the anime community and to also get an anime recommendation based on their preference.

Aim

We aim to create a cross-platform mobile application for anime lovers that will provide them with anime databases, a recommendation system based on their favorite genre and anime series, as well as a platform to discuss various topics with other users.

Project scope

Chattoku utilizes mobile applications (for Android and iOS) as the front-end for users to interact with each other through chats and forums, as well as providing them with anime databases. This will be done in React Native, and deployed in the Expo host.

Machine learning algorithms will be used to implement the anime recommendation system. This will be done using python and Flask. Furthermore, the recommendation system will be deployed in Heroku.

A set of back-end APIs will be used for authentication and database. This will be done using Firebase.

<u>User stories</u>

Priority level: *** - **Must** have, ** - **Should** have, * - **Could** have

Priority Level	As a …	I would like to …
***	User	Create an account and login to Chattoku.
***	User	Input my anime genre preferences
***	User	Input my favorite anime
***	User	Customise my username, profile picture, and bio
***	User	Be able to log out from Chattoku
***	User	Be able to delete my account in case I longer want to use Chattoku
***	User	Create a new forum
***	User	Be a super-admin of my created forum so that I can control all activity in this forum.
***	Forum super-admin	Do everything that forum admin can do in my forum
***	Forum admin	Ban or unban non-admins from my forum
***	User	Create, update, and delete my post and comment in a forum
***	User	Like or dislike on posts made in the forum
***	User	Follow a forum that interests me so that I can receive notifications every time there is a new post in this forum
***	User	Unfollow a forum that I previously followed when I no longer interested
***	User	Know the currently trending anime
**	User	Initiate a private chat with other users
**	User	Receive message notifications

		Become friend with other user so that I can view other’s favorite anime and genre
**	User	Send a friend request to other user
**	User	Cancel friend request that I’ve sent
**	User	Accept or reject friend request that I’ve received
**	User	Receive notification when other user sends a friend request
**	User	Receive notification when my friend request is accepted or rejected
**	User	Unfriend another user when I no longer want
**	User	Create a group with other users so that I can message everyone in this group
**	User	Be a group super-admin of my created group so that I can regulate group’s activity
**	Group super-admin	Do everything that group admin can do in my group
**	Group super-admin	Change user’s role from member to admin and vice versa
**	Group super-admin	Remove group admins from my group
**	Group admin	Invite other user to my group
**	Group admin	Cancel user invitation to my group
**	Group admin	Remove other non-admins from my group
**	User	Leave a group when I no longer interested in becoming it’s member
**	User	Receive notification when a group send me an invitation
**	User	Receive notification when one of my group has new member

**	User	Decline private chat requests with other users
**	Forum admin	Delete posts or comments made by non-admins in my forum
**	Forum super-admin	Change user’s role from member to admin and vice versa
**	Forum super-admin	Ban or unban forum admins from my forum
**	Forum super-admin	Delete posts or comments made by admins in my forum
**	User	Sort the forum posts by latest or trending
**	User	Find my previous forum posts easily
**	User	Find a detail of a specific anime
**	User	Get an anime recommendation based on my chosen genres and favorite anime series
*	User	Be able to switch to dark mode
*	User	Get a tutorial to use the Chattoku app

Progress (as of Milestone 2)

1. Authentication
    - Login page
    - Register page
    - Forgot Password page
    - Logout functionality
    - Resend authentication email functionality
2. Forum system
    - Allow users to create, update, and delete post and comments
    - Allow users to create their own forum
    - Allow users to search forum and forum post
    - Like and dislike post
    - Allow forum admin to ban users
    - Allow forum admin to delete any posts and comments on forum he/she owns
    - Allow admin to edit forum details
3. Anime database
    - Search a specific anime by its title
    - Add and remove anime from user's favorite list
    - Get currently airing and top anime
    - Get anime recommendations based on user's favorite genre or favorite anime
4. Friend system
    - Send friend request to other user
    - Accept or reject friend request sent by other user
    - Cancel friend request that has been sent
    - Search friend by username
    - Allow user to view friend's favorite genre & anime
5. Group system
    - Create group
    - Allow group admin to customize group's info
    - Allow group admin to invite other user
    - Allow group admin to remove group member (excluding other admins)
    - Allow group admin to cancel group invitation to other user
    - Allow group owner to delete group
    - Allow user to accept or reject group invitation
    - Allow user to leave group

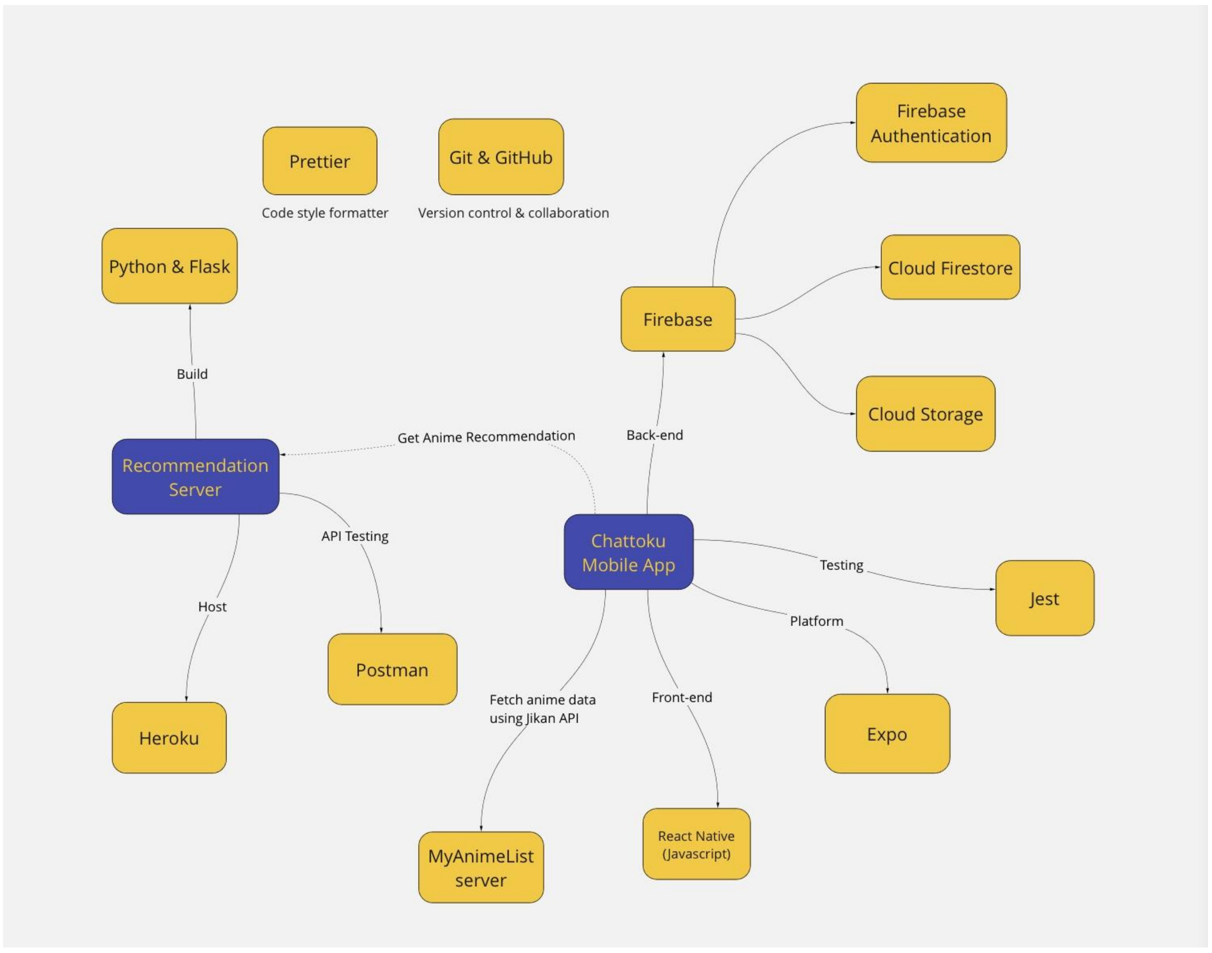
6. Chat system
    - One-on-One messaging
    - Group messaging
    - View active private & group chat lists
    - Remove specific chat from chat lists
    - Labels on unread message
7. Profile page
    - Allow user to customize its username, bio, and profile picture
    - Fetch user’s favorite anime and genre
    - Stack navigation to “edit favorite genre” section
    - Allow users to see past post
8. Notification System
    - Allow user to get notification when receiving friend request
    - Allow user to get notification when receiving group invitation
    - Allow user to get notification when receiving new message
    - Allow user to get notification when there is a new post in a forum that he/she followed

For more detailed explanation of our current feature, check our user’s guide

For feature’s explanation as well as implementation detail, check our developer’s guide

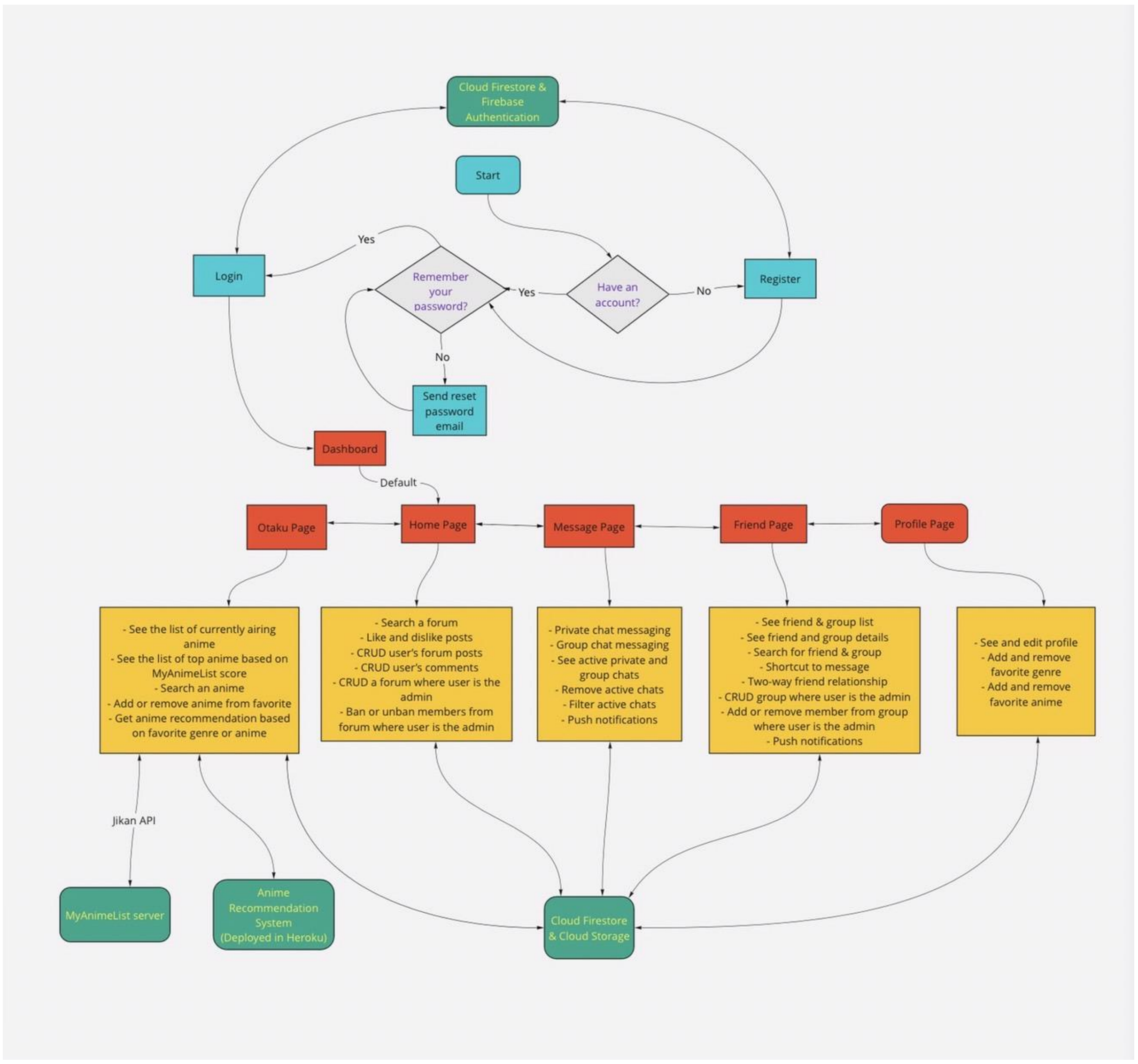
Software Architecture

Shown below is Chattoku's software architecture.



Application design flow

Shown below is Chattoku’s design flow.


Cloud Firestore & Firebase Authentication
Start
Login
Yes
Remember your password?
Yes
Have an account?
No
Register
No
Send reset password email
Dashboard
Default
Otaku Page
Home Page
Message Page
Friend Page
Profile Page
- See the list of currently airing anime
- See the list of top anime based on MyAnimeList score
- Search an anime
- Add or remove anime from favorite
- Get anime recommendation based on favorite genre or anime
- Search a forum
- Like and dislike posts
- CRUD user's forum posts
- CRUD user's comments
- CRUD a forum where user is the admin
- Ban or unban members from forum where user is the admin
- Private chat messaging
- Group chat messaging
- See active private and group chats
- Remove active chats
- Filter active chats
- Push notifications
- See friend & group list
- See friend and group details
- Search for friend & group
- Shortcut to message
- Two-way friend relationship
- CRUD group where user is the admin
- Add or remove member from group where user is the admin
- Push notifications
- See and edit profile
- Add and remove favorite genre
- Add and remove favorite anime
Jikan API
MyAnimeList server
Anime Recommendation System (Deployed in Heroku)
Cloud Firestore & Cloud Storage

Tech stack

Technology	Purposes	Reasons
Git & GitHub	Version control, collaboration, and code review to maintain code quality	It facilitates a seamless collaboration
React Native (Javascript)	Build the front-end of the mobile application	1. Popular framework (Lot of resources online) 2. Cross-platform (Easier to develop Android and iOS application together) 3. Members' familiarity with ReactJS
Firebase	Development of the back-end of the application, including storage and user authentication.	1. Popular backend service (Many resources on the internet) 2. Variety of APIs which are useful for the development of Chattoku 3. Has a free tier pricing plan
Expo	Deployment of the mobile application	Ease of deployment, especially for React Native app
Jest	Software testing for the Chattoku app	1. Popular testing framework (Various resources online) 2. Works well with React Native program
Python	Implement the machine learning algorithm to get anime recommendations for users	Python has various libraries that are useful in developing a machine learning model.
Heroku	Deployment of the anime	Ease of deployment and

	recommendation server	members' past experience with Heroku
Postman	API testing for the anime recommendation server	It provides API testing
Jikan API	Fetch anime data from myanimelist server.	It is a free and open-source API which makes getting anime data much easier
Prettier	Code auto formatter	It eases the standardization of code styling

Software Engineering Practices

- ### *GitHub Version Control*

To easily collaborate with each other, we use GitHub to maintain the project's integrity. Everytime we want to create a new feature, fix bugs, or add more testing, we create a new issue first where we explain concisely what is going to be changed.
Then, we branch off the *master* branch, and continue to work on the new branch. When we are ready to merge the modified branch into our *master* branch, we will open a new Pull Request where we explain the updated features that the branch contains. The other user will then review the branch to ensure that there is no bug in the program, as well as maintaining that the code follows Software Engineering principles such as *Abstraction* or *Separation of Concerns*. After the other user has approved the change in this branch, the branch will be merged to the *master* branch.

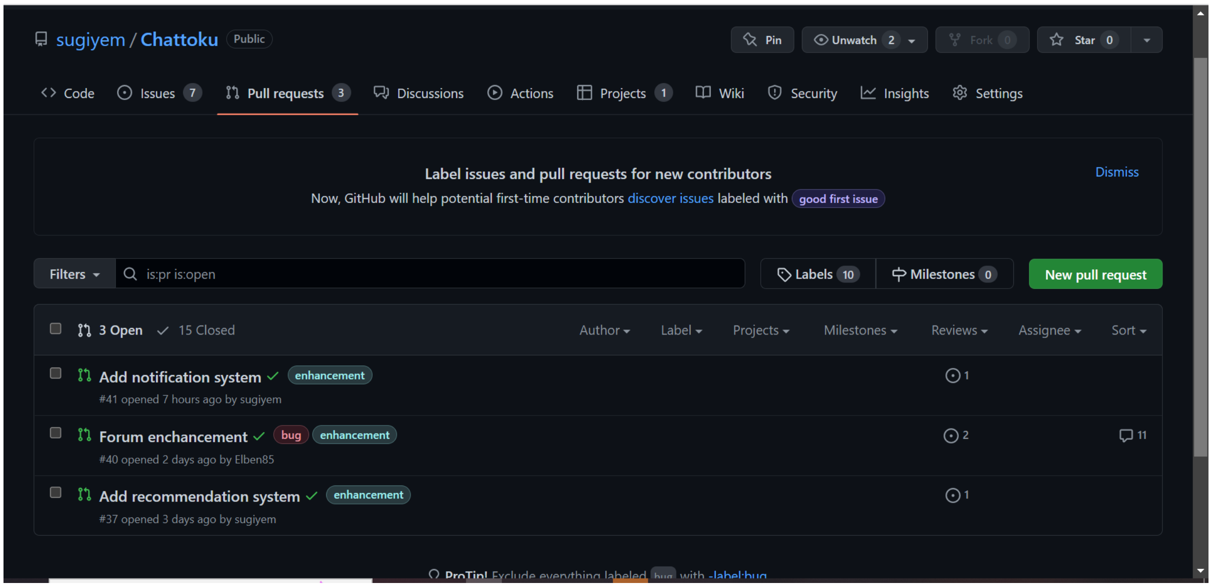
- ### *Continuous Integration with GitHub Actions*

To further ensure that any changes to the *master* branch are not breaking the entire codebase, we use GitHub Actions to set up our own workflow. This workflow will be automated as follows,

1. Do a clean install of the project dependencies on the latest Ubuntu version
2. Run all written tests in the project (currently, unit and UI tests)

The workflow will pass if and only if all the tests in the project pass. A badge will appear to indicate whether the workflow passed or failed.

Each creation of a pull request will trigger this workflow to help developers spot any issues with the code before merging to the main branch. Furthermore, each push to the *master* branch of the project repository will also trigger this workflow.


sugiyem / Chattoku Public

Pin | Unwatch 2 | Fork 0 | Star 0

Code | Issues 7 | Pull requests 3 | Discussions | Actions | Projects 1 | Wiki | Security | Insights | Settings

Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

Dismiss

Filters | is:pr is:open | Labels 10 | Milestones 0 | New pull request

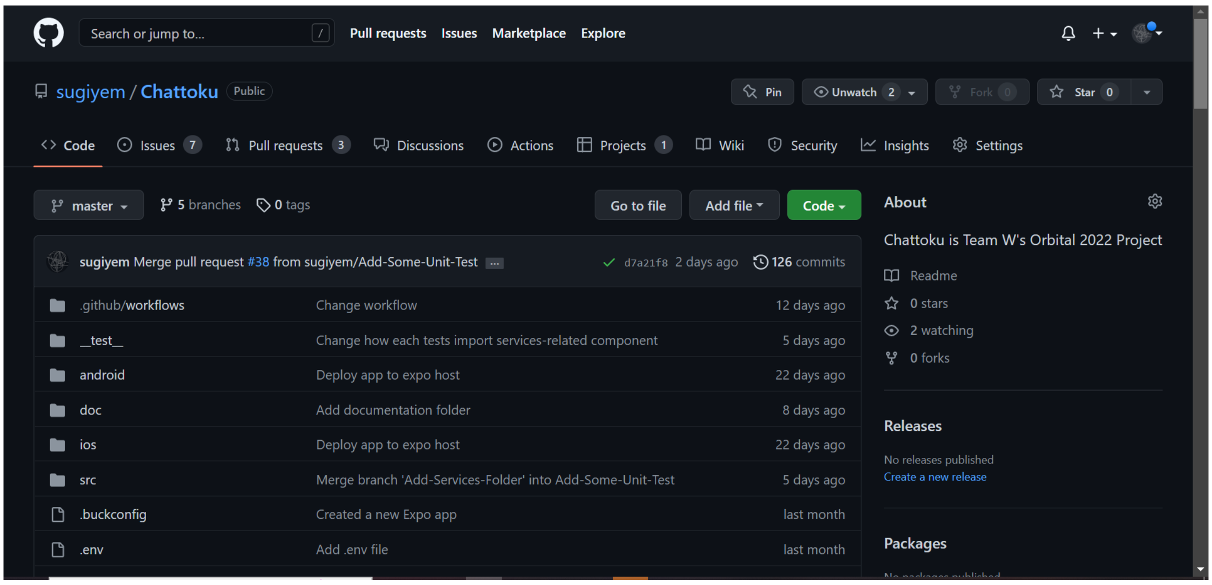
3 Open ✓ 15 Closed | Author | Label | Projects | Milestones | Reviews | Assignee | Sort

Add notification system ✓ enhancement — 1
#41 opened 7 hours ago by sugiyem

Forum enchancement ✓ bug enhancement — 2 — 11
#40 opened 2 days ago by Elben85

Add recommendation system ✓ enhancement — 1
#37 opened 3 days ago by sugiyem

ProTip! Exclude everything labeled bug with -label:bug


Search or jump to... | Pull requests | Issues | Marketplace | Explore

sugiyem / Chattoku Public

Pin | Unwatch 2 | Fork 0 | Star 0

Code | Issues 7 | Pull requests 3 | Discussions | Actions | Projects 1 | Wiki | Security | Insights | Settings

master | 5 branches | 0 tags | Go to file | Add file | Code

sugiyem Merge pull request #38 from sugiyem/Add-Some-Unit-Test ... ✓ d7a21f8 2 days ago 126 commits

.github/workflows	Change workflow	12 days ago
__test__	Change how each tests import services-related component	5 days ago
android	Deploy app to expo host	22 days ago
doc	Add documentation folder	8 days ago
ios	Deploy app to expo host	22 days ago
src	Merge branch 'Add-Services-Folder' into Add-Some-Unit-Test	5 days ago
.buckconfig	Created a new Expo app	last month
.env	Add .env file	last month

About

Chattoku is Team W's Orbital 2022 Project

Readme
0 stars
2 watching
0 forks

Releases

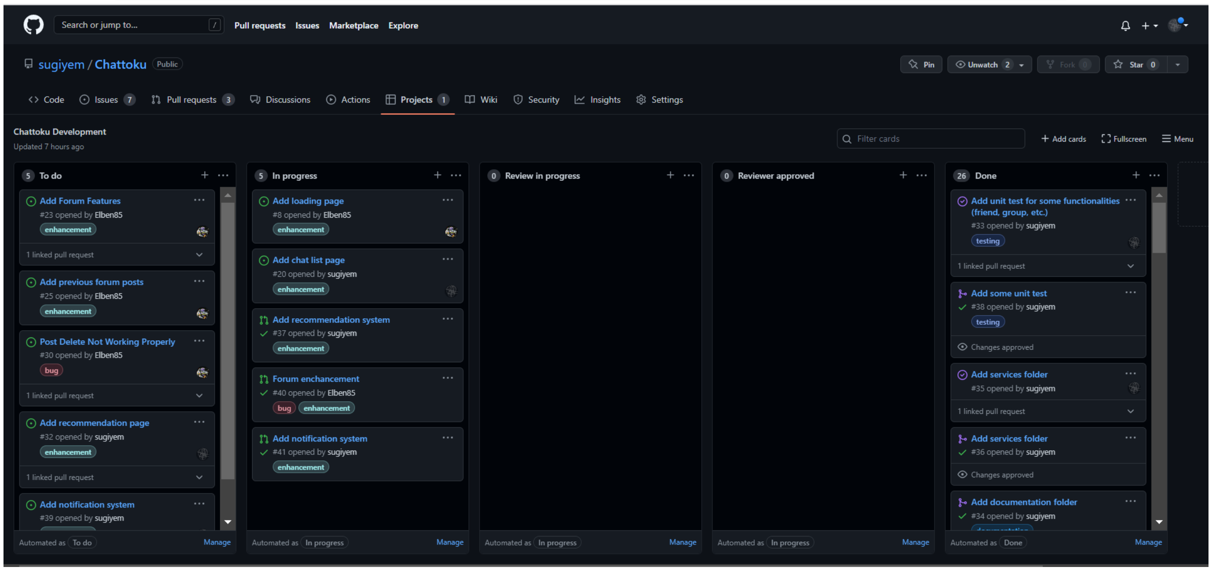
No releases published
Create a new release

Packages

No packages published

- ***Planning with GitHub Projects***

Using GitHub Projects, we employ a Card-Based task allocation system to keep track of our current progress. This helps us to keep each other to work in an organized manner.



sugiyem / Chattoku Public

Chattoku Development
Updated 7 hours ago

5 To do

Add Forum Features
#23 opened by Elben85
enhancement
1 linked pull request

Add previous forum posts
#25 opened by Elben85
enhancement

Post Delete Not Working Properly
#30 opened by Elben85
bug
1 linked pull request

Add recommendation page
#32 opened by sugiyem
enhancement
1 linked pull request

Add notification system
#39 opened by sugiyem

Automated as To do — Manage

5 In progress

Add loading page
#8 opened by Elben85
enhancement

Add chat list page
#20 opened by sugiyem
enhancement

Add recommendation system
✓ #37 opened by sugiyem
enhancement

Forum enchancement
✓ #40 opened by Elben85
bug enhancement

Add notification system
✓ #41 opened by sugiyem
enhancement

Automated as In progress — Manage

0 Review in progress

Automated as In progress — Manage

0 Reviewer approved

Automated as In progress — Manage

26 Done

Add unit test for some functionalities (friend, group, etc.)
#33 opened by sugiyem
testing
1 linked pull request

Add some unit test
✓ #38 opened by sugiyem
testing
Changes approved

Add services folder
#35 opened by sugiyem
1 linked pull request

Add services folder
✓ #36 opened by sugiyem
Changes approved

Add documentation folder
✓ #34 opened by sugiyem

Automated as Done — Manage

- *Code Style*

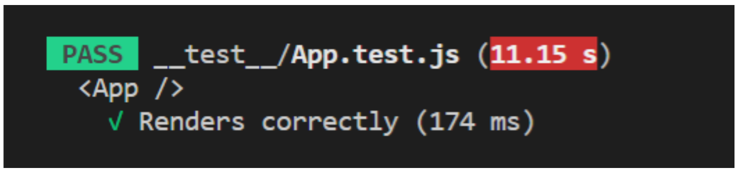
To maintain a standardized code styling, we both use Prettier, a JavaScript code formatter, in our IDE. With this standardized approach, the code will become more readable, and hence we can understand each other’s codes more efficiently,

Testing

As of milestone 2, some testing has been done for our project. We've written some unit tests using Jest framework to ensure the quality and correctness of our app. The test results are shown below.

- ***Test Entire Application***

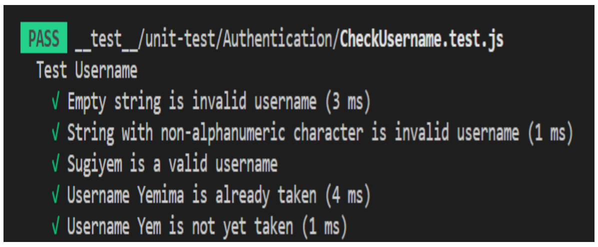
Objective	Result (Pass / Fail)
To test that entire application run without error	Pass



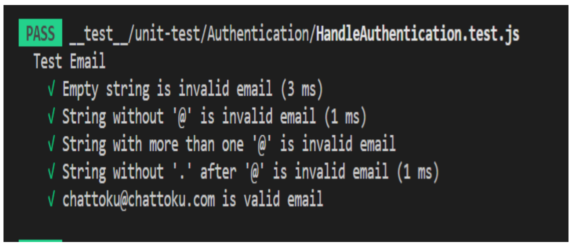
```
PASS  __test__/App.test.js (11.15 s)
  <App />
    √ Renders correctly (174 ms)
```

- ***Test Authentication System***

Objective	Result (Pass / Fail)
Test that user's username must be a non-empty string	Pass
Test that user's username must consists of alphanumeric characters	Pass
Test that user's username must be unique	Pass
Test that user's email must be valid	Pass



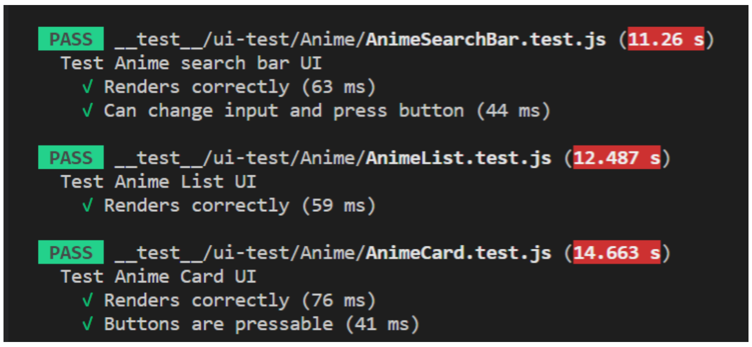
```
PASS  __test__/unit-test/Authentication/CheckUsername.test.js
  Test Username
    √ Empty string is invalid username (3 ms)
    √ String with non-alphanumeric character is invalid username (1 ms)
    √ Sugiyem is a valid username
    √ Username Yemima is already taken (4 ms)
    √ Username Yem is not yet taken (1 ms)
```



```
PASS  __test__/unit-test/Authentication/HandleAuthentication.test.js
  Test Email
    √ Empty string is invalid email (3 ms)
    √ String without '@' is invalid email (1 ms)
    √ String with more than one '@' is invalid email
    √ String without '.' after '@' is invalid email (1 ms)
    √ chattoku@chattoku.com is valid email
```

- ***Test Anime Front-End Component***

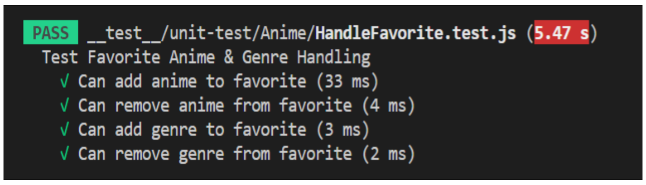
Objective	Result (Pass / Fail)
To test that Anime List component renders as expected	Pass
To test that Anime Card component renders as expected	Pass
To test that Anime Search Bar component renders as expected	Pass



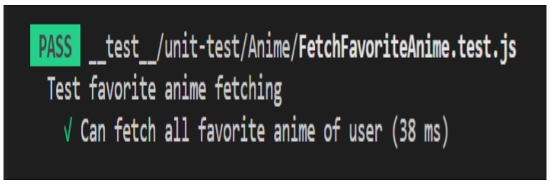
```
 PASS  __test__/ui-test/Anime/AnimeSearchBar.test.js (11.26 s)
  Test Anime search bar UI
    √ Renders correctly (63 ms)
    √ Can change input and press button (44 ms)

 PASS  __test__/ui-test/Anime/AnimeList.test.js (12.487 s)
  Test Anime List UI
    √ Renders correctly (59 ms)

 PASS  __test__/ui-test/Anime/AnimeCard.test.js (14.663 s)
  Test Anime Card UI
    √ Renders correctly (76 ms)
    √ Buttons are pressable (41 ms)
```

- ***Test Anime & Genre Back-End System***

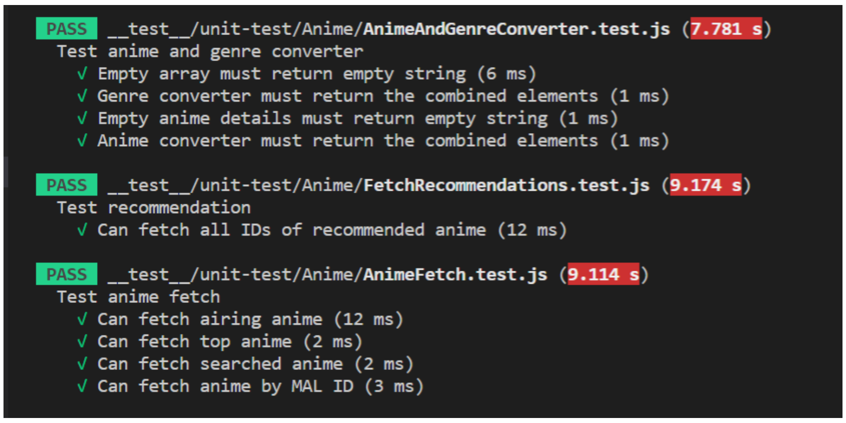
Objective	Result (Pass / Fail)
To test that user can add or remove anime from favorite	Pass
To test that user can add or remove genre from favorite	Pass
To test that user can fetch its favorite anime	Pass
To test that user can fetch currently airing anime	Pass
To test that user can fetch top anime	Pass
To test that user can fetch searched anime	Pass
To test that user can fetch specific anime based on its MyAnimeList ID	Pass
To test that the system can convert anime and genres to its string representation	Pass
To test that user can fetch the data of recommended anime	Pass



```
PASS  __test__/unit-test/Anime/HandleFavorite.test.js (5.47 s)
  Test Favorite Anime & Genre Handling
    √ Can add anime to favorite (33 ms)
    √ Can remove anime from favorite (4 ms)
    √ Can add genre to favorite (3 ms)
    √ Can remove genre from favorite (2 ms)
```



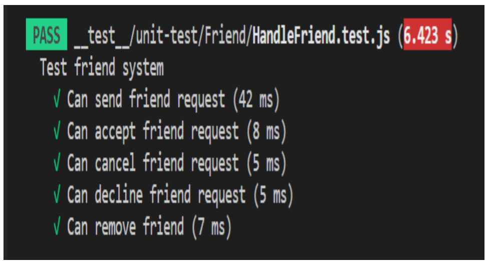
```
PASS  __test__/unit-test/Anime/FetchFavoriteAnime.test.js
  Test favorite anime fetching
    √ Can fetch all favorite anime of user (38 ms)
```



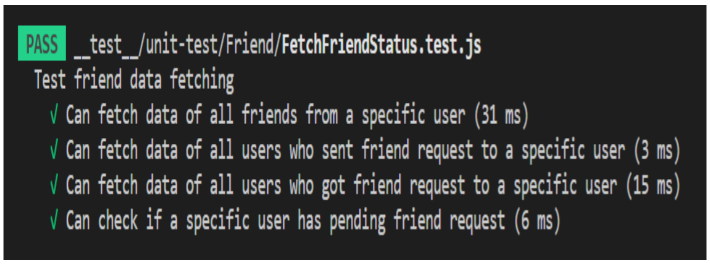
```
PASS  __test__/unit-test/Anime/AnimeAndGenreConverter.test.js (7.781 s)
  Test anime and genre converter
    √ Empty array must return empty string (6 ms)
    √ Genre converter must return the combined elements (1 ms)
    √ Empty anime details must return empty string (1 ms)
    √ Anime converter must return the combined elements (1 ms)

PASS  __test__/unit-test/Anime/FetchRecommendations.test.js (9.174 s)
  Test recommendation
    √ Can fetch all IDs of recommended anime (12 ms)

PASS  __test__/unit-test/Anime/AnimeFetch.test.js (9.114 s)
  Test anime fetch
    √ Can fetch airing anime (12 ms)
    √ Can fetch top anime (2 ms)
    √ Can fetch searched anime (2 ms)
    √ Can fetch anime by MAL ID (3 ms)
```

- ***Test Friend System***

Objective	Result (Pass / Fail)
To test that user can send friend request	Pass
To test that user can accept / reject friend request	Pass
To test that user can cancel friend request	Pass
To test that user can unfriend	Pass
To test that user can fetch its friends data	Pass
To test that user can fetch its incoming requests	Pass
To test that user can fetch its outgoing requests	Pass



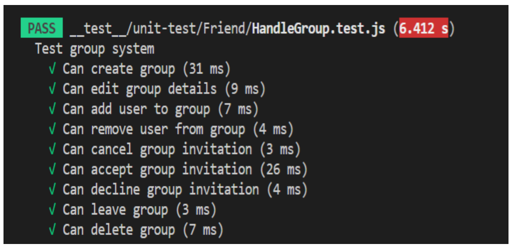
```
PASS  __test__/unit-test/Friend/HandleFriend.test.js (6.423 s)
  Test friend system
    √ Can send friend request (42 ms)
    √ Can accept friend request (8 ms)
    √ Can cancel friend request (5 ms)
    √ Can decline friend request (5 ms)
    √ Can remove friend (7 ms)
```



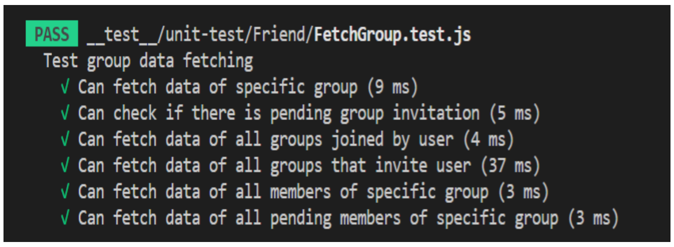
```
PASS  __test__/unit-test/Friend/FetchFriendStatus.test.js
  Test friend data fetching
    √ Can fetch data of all friends from a specific user (31 ms)
    √ Can fetch data of all users who sent friend request to a specific user (3 ms)
    √ Can fetch data of all users who got friend request to a specific user (15 ms)
    √ Can check if a specific user has pending friend request (6 ms)
```

- ***Test Group System***

Objective	Result (Pass / Fail)
To test that user can create and edit group	Pass
To test that user can add another user to a group	Pass
To test that user can remove another user from a group	Pass
To test that user can cancel group invitation	Pass
To test that user can accept / reject group invitation	Pass
To test that user can leave group	Pass
To test that user can delete group	Pass
To test that user can fetch data of all its groups	Pass
To test that user can fetch all groups that invite it	Pass



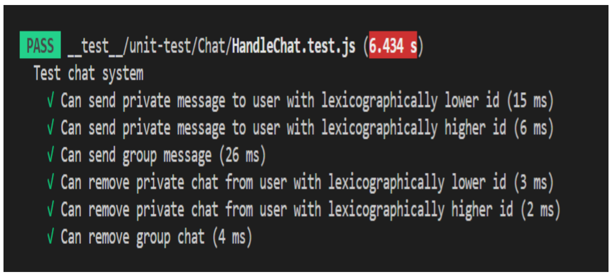
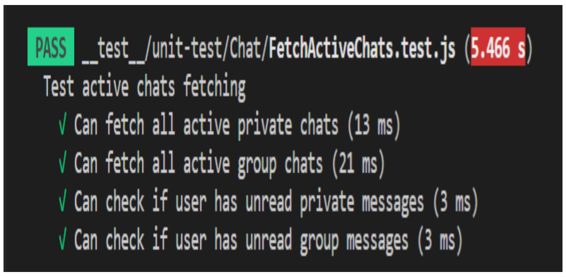
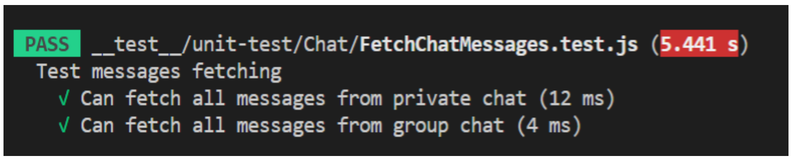
```
PASS  __test__/unit-test/Friend/HandleGroup.test.js (6.412 s)
  Test group system
    √ Can create group (31 ms)
    √ Can edit group details (9 ms)
    √ Can add user to group (7 ms)
    √ Can remove user from group (4 ms)
    √ Can cancel group invitation (3 ms)
    √ Can accept group invitation (26 ms)
    √ Can decline group invitation (4 ms)
    √ Can leave group (3 ms)
    √ Can delete group (7 ms)
```



```
PASS  __test__/unit-test/Friend/FetchGroup.test.js
  Test group data fetching
    √ Can fetch data of specific group (9 ms)
    √ Can check if there is pending group invitation (5 ms)
    √ Can fetch data of all groups joined by user (4 ms)
    √ Can fetch data of all groups that invite user (37 ms)
    √ Can fetch data of all members of specific group (3 ms)
    √ Can fetch data of all pending members of specific group (3 ms)
```

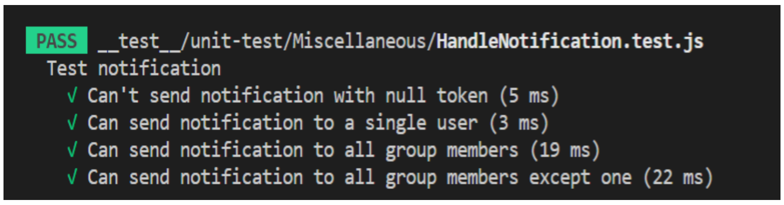
- ***Test Message System***

Objective	Result (Pass / Fail)
To test that user can send a private and group message	Pass
To test that user can remove private and group chat from active chat lists	Pass
To test that user can fetch all its active private and group chats	Pass
To test that user can check if there is unread private or group message	Pass


```
PASS __test__/unit-test/Chat/HandleChat.test.js (6.434 s)
  Test chat system
    √ Can send private message to user with lexicographically lower id (15 ms)
    √ Can send private message to user with lexicographically higher id (6 ms)
    √ Can send group message (26 ms)
    √ Can remove private chat from user with lexicographically lower id (3 ms)
    √ Can remove private chat from user with lexicographically higher id (2 ms)
    √ Can remove group chat (4 ms)
```

```
PASS __test__/unit-test/Chat/FetchActiveChats.test.js (5.466 s)
  Test active chats fetching
    √ Can fetch all active private chats (13 ms)
    √ Can fetch all active group chats (21 ms)
    √ Can check if user has unread private messages (3 ms)
    √ Can check if user has unread group messages (3 ms)
```

```
PASS __test__/unit-test/Chat/FetchChatMessages.test.js (5.441 s)
  Test messages fetching
    √ Can fetch all messages from private chat (12 ms)
    √ Can fetch all messages from group chat (4 ms)
```

- ***Test Notification System***

Objective	Result (Pass / Fail)
To test that notification can be pushed to a specific user	Pass
To test that notification can be pushed to all users in a specific group	Pass


```
PASS __test__/unit-test/Miscellaneous/HandleNotification.test.js
  Test notification
    √ Can't send notification with null token (5 ms)
    √ Can send notification to a single user (3 ms)
    √ Can send notification to all group members (19 ms)
    √ Can send notification to all group members except one (22 ms)
```

- ***Test Anime Recommendation System API***

Additionally, using Postman, we have also tested out HTTP requests sent to our Anime Recommendation System endpoint as summarized below.

Base URL: https://chattoku-recommendation-system.herokuapp.com

Endpoint	Method	Body	Expected Status	Expected Output	Result (Pass / Fail)
/predict	GET	-	405 (method not allowed)	-	Pass
/predict	POST	-	400 (bad request)	-	Pass
/predict	POST	{ “genres”: “” }	200 (Ok)	Array containing 5 IDs	Pass
/predict	POST	{ “genres”: “Action, Comedy” }	200 (Ok)	Array containing 5 IDs	Pass
/predict	POST	{ “genres”: “Action, Comedy”, “anyData”: [] }	200 (Ok)	Array containing 5 IDs	Pass

Our future plans?

Features to be completed by mid of July:

1. User authentication: delete account system
2. Friend: allow user to block specific users
3. Forum: allow forum owner to add / remove admin
4. Forum: allow users to sort posts based on timestamp
5. Start to add more detailed testing

Features to be completed by end of July:

1. Testing finalization
2. Documentation
3. Style improvement

Problems Encountered

1. Familiarization - there are some tech stacks which none of our team members had used before, which causes some delay in the project due to the need to familiarize ourselves with the tech stacks.
2. Many bugs - we have encountered a lot of advanced bugs which do not affect the workflow of the app, but causes some warnings to appear.

Deployment details

As of milestone 2, Chattoku has been deployed for users to take a look at what this app can do.

To test this app, first you need to download Expo Go on your mobile phones.

Then, for iOS users only, there is one additional step that you need to do. Simply do any one of two choices given below.

- Create an Expo Go account using your email and send your registered email to our telegram account (@sugiyemm or @elben85)

- Login to Exgo Go using our test account (username: chattoku-test , password: chattokucp2106)

After that, you simply need to scan the QR code below to try using Chattoku. Alternatively, you can also click on the link below.

Link to our deployed app: Link

Note: After signing up, you need to verify your email first. Do check your spam folder for that.

Project Log

Task	Date	Farrel	Elbert	Remarks
Refinement of ideas	9 May 2022	2h	2h	1. Revise project proposal
Refinement of ideas	10 May 2022	2h	2h	1. Revise project proposal
Lift-Off	11 May 2022	3h	1h	1. Poster and video creation
Project consultation	12 May 2022	3h	1h	1. Project consultation with advisor 2. Picking up react native & firebase
Picking up relevant technology	13 May 2022	2h	0h	1. Picking up react native & firebase
Mission control 1 & Team meeting	14 May 2022	5h	4h	1. React Native tutorial 2. Discuss what to do for the next two weeks 3. Create the github repository 4. Configure firebase with the app
Application development	15 May 2022	1h	0h	1. Continue configuring firebase to the app
Application development	16 May 2022	2h	1h	1. Create navigation bar for dashboard screen 2. Enable fetching anime data 3. Check firebase configuration
Application development	17 May 2022	2h	0h	1. Create anime search bar
Application development	18 May 2022	2h	0h	1. Anime screen UI

Application development	19 May 2022	2h	0h	1. Anime screen UI
Application development	20 May 2022	1h	0h	1. Create navigation to login and signup screen
Mission control 2	21 May 2022	5h	6h	1. Software Engineering tutorial 2. Login & Sign Up 3. Database Discussion 4. Code Review 5. Profile screen UI
Application development	22 May 2022	4h	8h	1.Login & Sign Up 2. Profile screen UI 3. Enable app to use camera & library
Application development	23 May 2022	2h	0h	1. Favorite anime & genre UI
Application development	24 May 2022	4h	1h	1.Login & Signup Bug Fixes 2. Allow user to add and remove a specific anime to favorite 3. Edit profile system
Application development	25 May 2022	3h	0h	1. Edit genre screen
Application development	26 May 2022	2h	0h	1. Clean up code 2. Prototype friend system
Application development	27 May 2022	5h	1h	1. Prototype friend system 2. Code Review 3. Fix bug in iOS
Mission control 3	28 May 2022	7h	8h	1. Machine learning tutorial 2. Code review 3. Forum prototype

				4. Basic chat functionality 5. Add README to github repo
Application development	29 May 2022	8h	15h	1. Forum Prototype 2. Basic chat functionality 3. Add README to github repo 4. Add more feature to profile customization system
Deployment	30 May 2022	10h	2h	1. Debug code 2. Add more styling to the app 3. Write milestone 1 report and user's guide 4. Deploy app to expo host
Break	31 May 2022	0h	0h	rest
Break	1 June 2022	0h	0h	rest
Application development	2 June 2022	5h	0h	1. Add prototype chat list
Application development	3 June 2022	2h	0h	1. Add forgot password feature
Mission control 4	4 June 2022	6h	6h	1. Machine learning tutorial 2 2. Code review 3. Dividing Task 4. Modify friend system 5. Fix iOS testing error
Application development	5 June 2022	5h	6h	1. Add Edit Feature 2. Research on like + dislike system, debouncing 3. Fix Post Delete

				Bug 4. Modify friend system
Application development & testing	6 June 2022	2h	0h	1. Configure jest and add simple testing 2. Add workflow
Application development	7 June 2022	6h	2h	1. Code review 2. Add group system
Application development	8 June 2022	6h	0h	1. Add group system 2. Research on recommendation system
Recommendation system development	9 June 2022	2h	2h	1. Convert data from myAnimeList from json to csv using node.js 2. Create prototype anime recommendation system
Recommendation system development	10 June 2022	2h	0h	1. Refine the recommendation system
Recommendation system development & testing	11 June 2022	5h	2h	1. Deploy recommendation system to heroku 2. Test recommendation system API
Application development & testing	12 June 2022	3h	6h	1. Unit testing 2. Add Like System 3. Add Forum and post search 4. Add Create Forum Skeleton
Application development & documentation	13 June 2022	4h	0h	1. Add recommendation page 2. Add

				documentation folder
Application development	14 June 2022	5h	2h	1. Code Review 2. Add recommendation page 3. Refine group system code
Application development	15 June 2022	2h	0h	1. Create services folder for managing Backend-related functionality 2. Fix group invitation label bug
Application testing	16 June 2022	3h	0h	1. Add testing for Anime-related component
Application development	17 June 2022	5h	0h	1. Add active group chat list
Application development	18 June 2022	1h	6h	1. Styled create forum screen 2. Add edit forum screen 3. Allow owner to delete users post 4. Add forum banner 5. Clean up Chat-related code
Application development	19 June 2022	1h	8h	1. Research on notification for Expo app 2. Created Edit Forum Screen 3. Implemented ban / unban system
Application testing	20 June 2022	3h	0h	1. Add testing for Chat-related component
Application development	21 June 2022	5h	0h	1. Code review 2. Add notification on some friend and

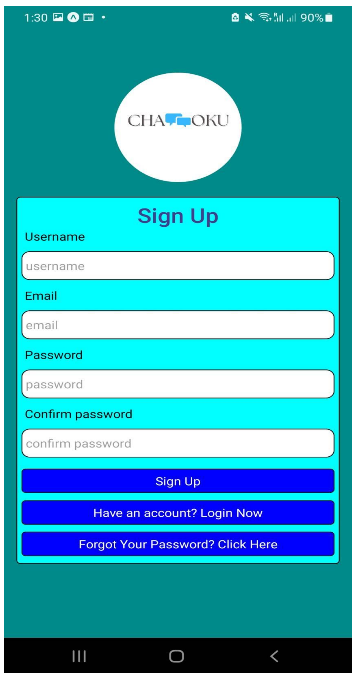
				group component
Application development & documentation	22 June 2022	3h	3h	1. Write milestone 2 report & developer's guide 2. Code Review 3. Reorganized some code
Application development & documentation	23 June 2022	7h	2h	1. Code review 2. Update notifications 3. Write milestone 2 report & developer's guide 4. Add notification unit testing
Application development & documentation	24 June 2022	6h	0h	1. Code review 2. Add chat notifications 3. Write testing report
Application development & documentation	25 June 2022	6h	8h	1. Code review 2. Update project's poster 3. Continue working on the developer's guide 4. Fix chat bug 5. forum re-styling 6.start on creating past forum post list
Application development & documentation	26 June 2022	5h	15h	1. Code review 2. Create app demo video 3. Fix forum bug 4. Add forum follow system 5. Add list of past forum post
Application testing & documentation	27 June 2022	6h	2h	1. Code review 2. Add unit test for signup, login, logout 3. Fix anime search

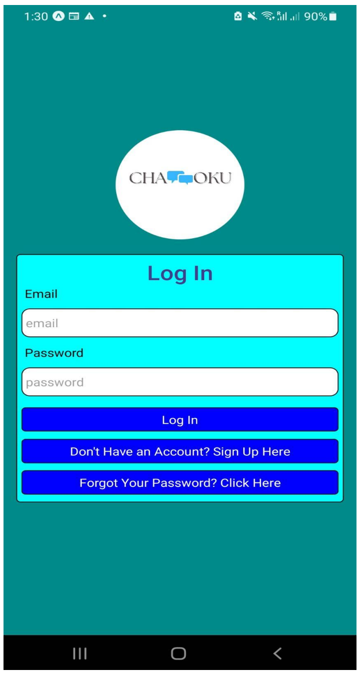
				bar bug 4. Publish the newest version of app to expo server 5. Finalize milestone 2 submission
Total		189h	112h	

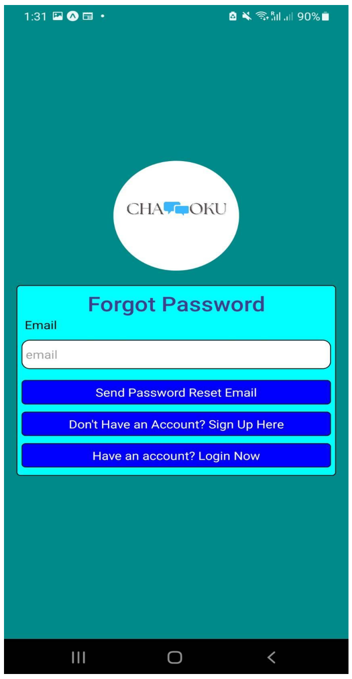
Sample Screenshots

Shown below are some screenshots of feature which has been implemented in our application as of milestone 2.

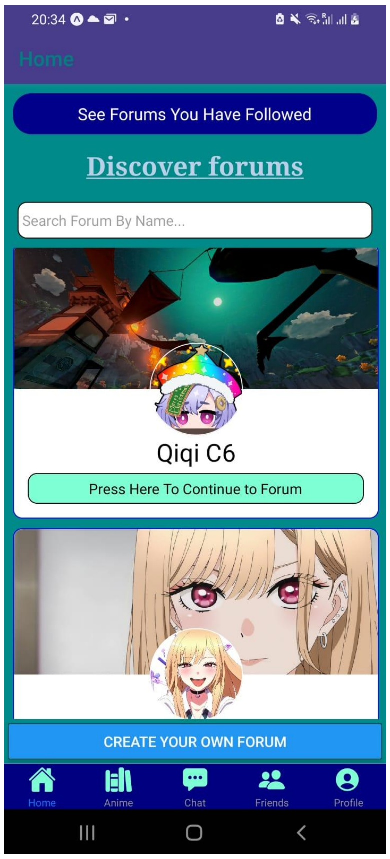
- Login & Sign Up

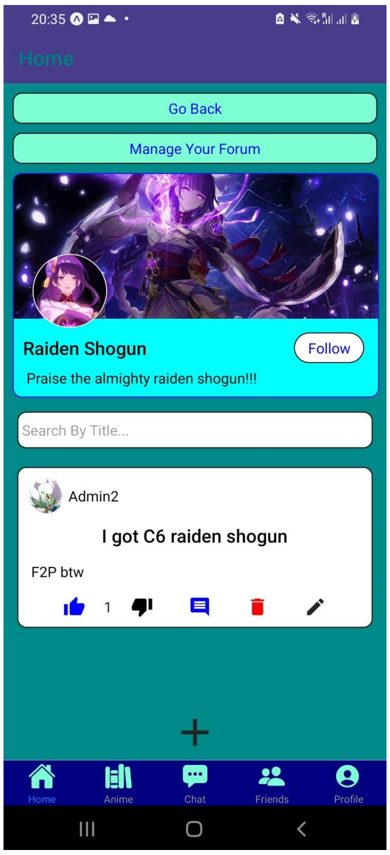


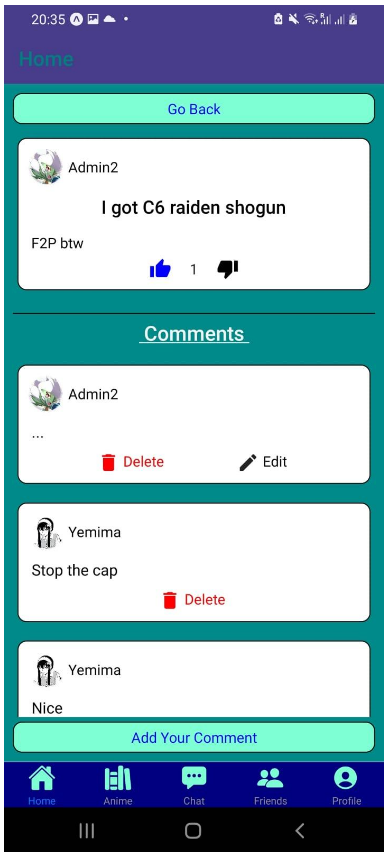

1:30
90%
CHATOKU
Log In
Email
email
Password
password
Log In
Don't Have an Account? Sign Up Here
Forgot Your Password? Click Here


1:31
90%
CHATOKU
Forgot Password
Email
email
Send Password Reset Email
Don't Have an Account? Sign Up Here
Have an account? Login Now

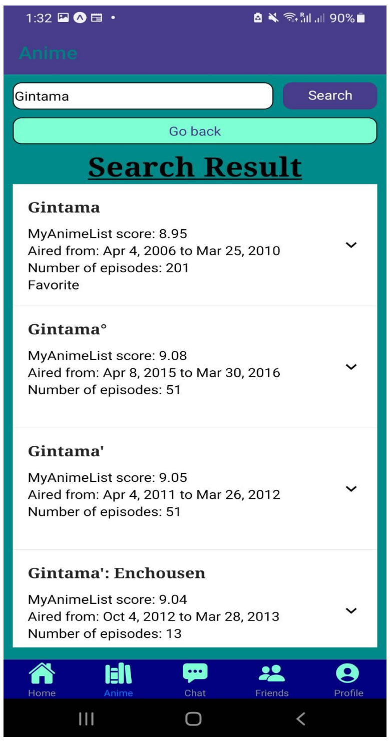
- Forum section

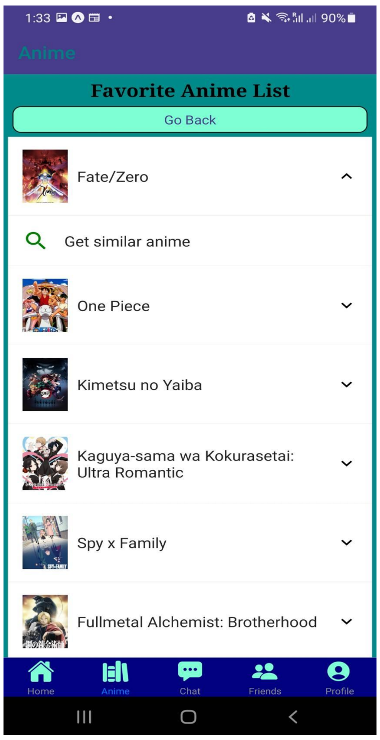


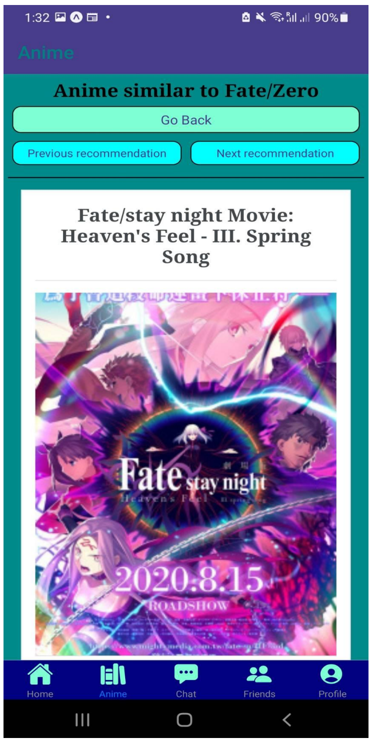




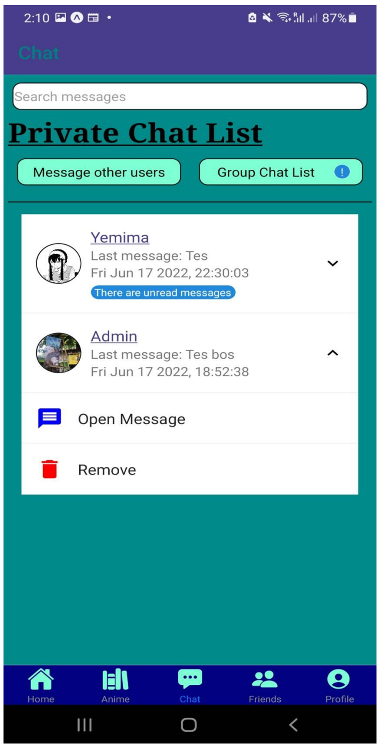
- Anime section







- Chat section



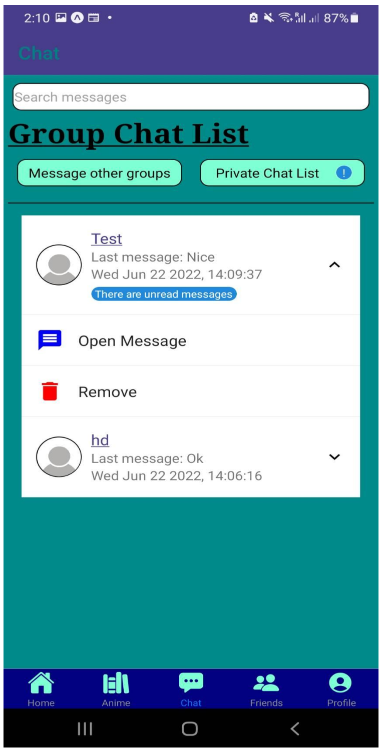
Private Chat List

Yemima
Last message: Tes
Fri Jun 17 2022, 22:30:03
There are unread messages

Admin
Last message: Tes bos
Fri Jun 17 2022, 18:52:38

Open Message

Remove



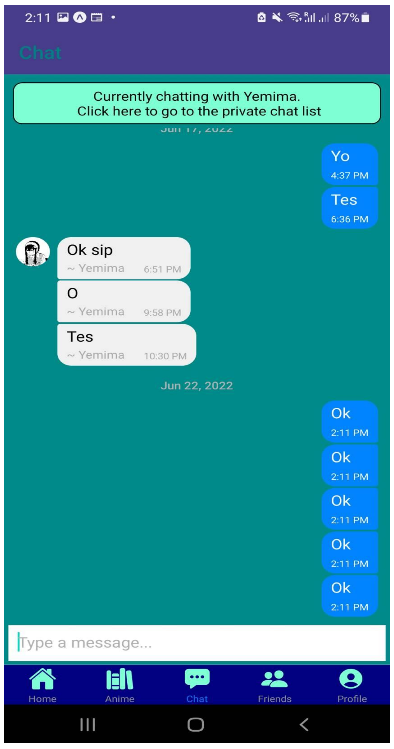
Group Chat List

Test
Last message: Nice
Wed Jun 22 2022, 14:09:37
There are unread messages

Open Message

Remove

hd
Last message: Ok
Wed Jun 22 2022, 14:06:16



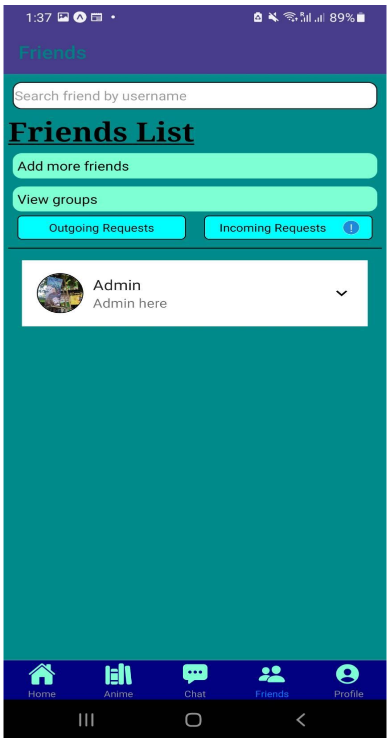
Currently chatting with Yemima.
Click here to go to the private chat list

Yo 4:37 PM
Tes 6:36 PM
Ok sip ~ Yemima 6:51 PM
O ~ Yemima 9:58 PM
Tes ~ Yemima 10:30 PM

Jun 22, 2022

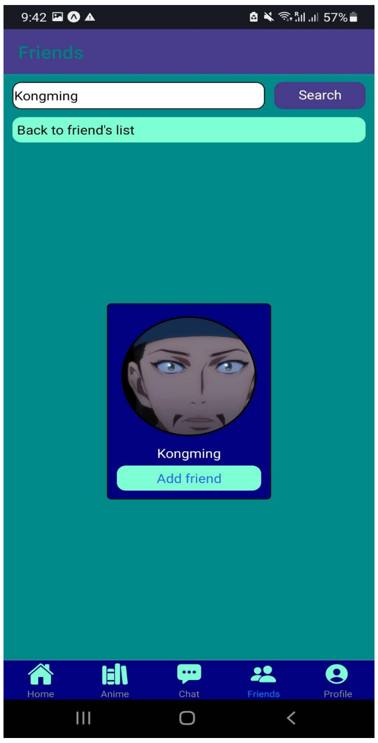
Ok 2:11 PM
Ok 2:11 PM
Ok 2:11 PM
Ok 2:11 PM
Ok 2:11 PM

- Friend section



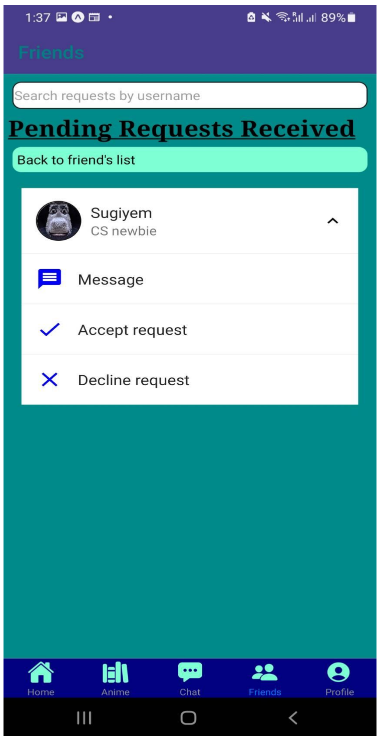
Friends List

Admin
Admin here



Kongming

Kongming



Pending Requests Received

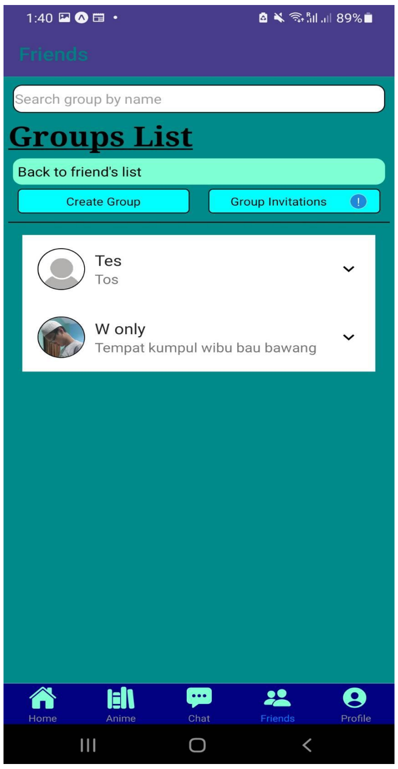
Sugiyem
CS newbie

Message

Accept request

Decline request

- Group section


Friends

Search group by name

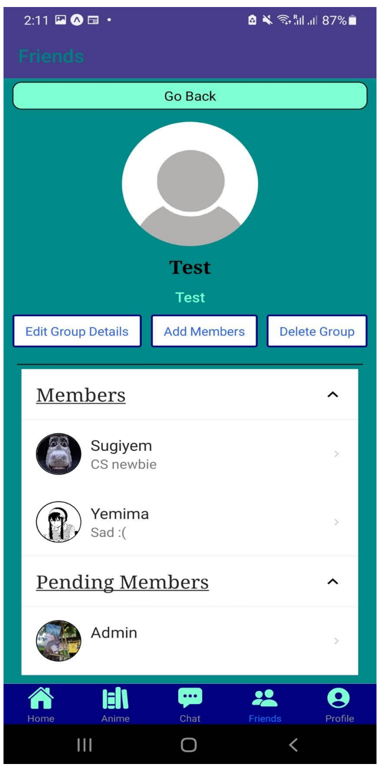
Groups List

Back to friend's list

Create Group | Group Invitations

Tes
Tos

W only
Tempat kumpul wibu bau bawang


Friends

Go Back

Test
Test

Edit Group Details | Add Members | Delete Group

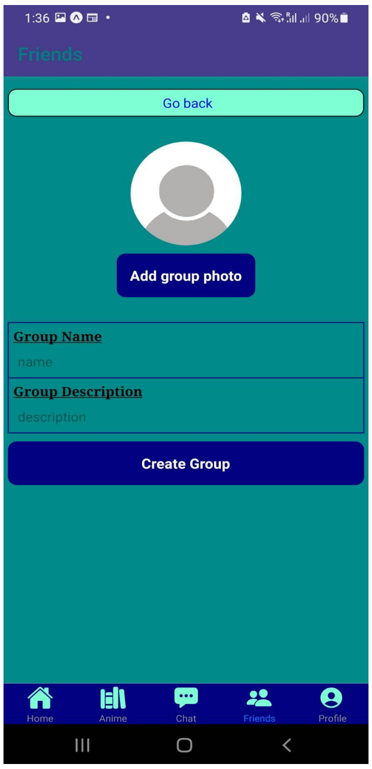
Members

Sugiyem
CS newbie

Yemima
Sad :(

Pending Members

Admin


Friends

Go back

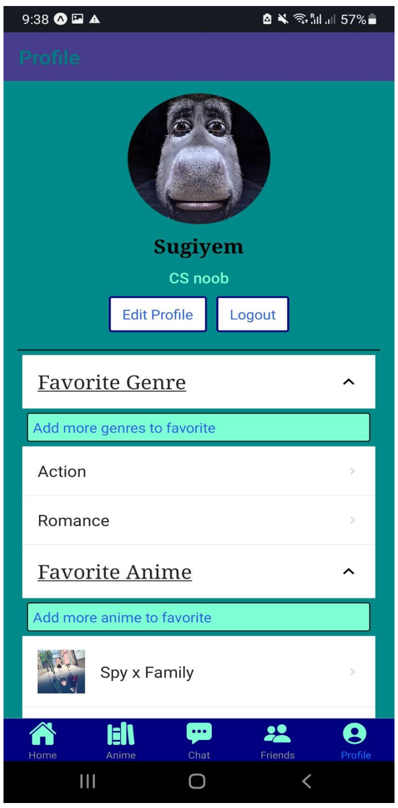
Add group photo

Group Name
name

Group Description
description

Create Group

- Profile Section


Profile

Sugiyem
CS noob

Edit Profile | Logout

Favorite Genre

Add more genres to favorite

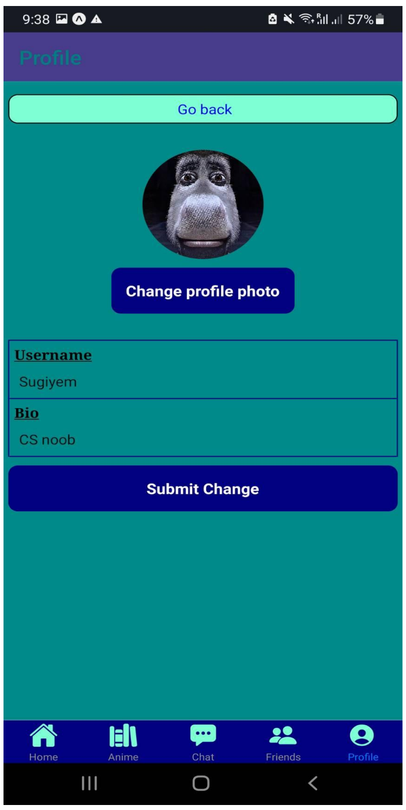
Action

Romance

Favorite Anime

Add more anime to favorite

Spy x Family


Profile

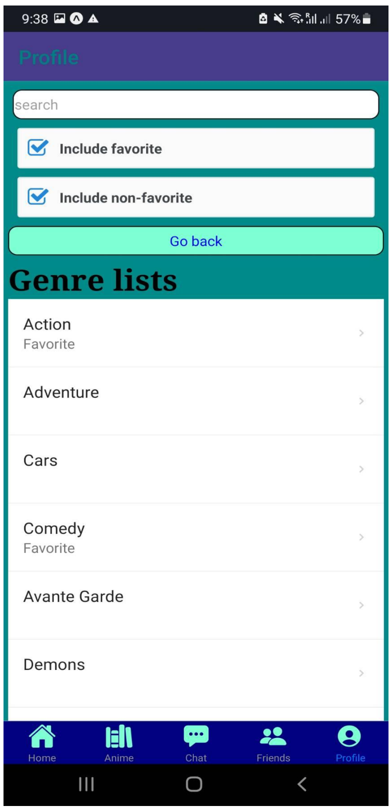
Go back

Change profile photo

Username
Sugiyem

Bio
CS noob

Submit Change


Profile

search

☑ Include favorite

☑ Include non-favorite

Go back

Genre lists

Action
Favorite

Adventure

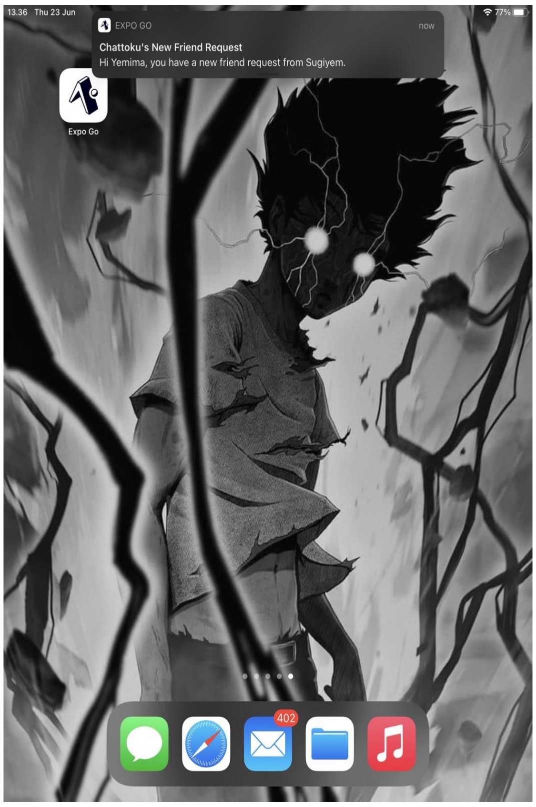
Cars

Comedy
Favorite

Avante Garde

Demons

- Notification System

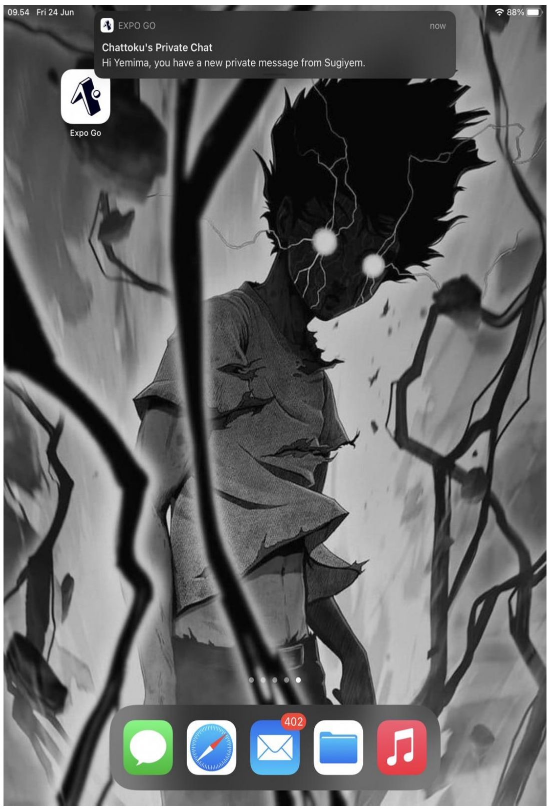


13.36 Thu 23 Jun

EXPO GO now

Chattoku's New Friend Request

Hi Yemima, you have a new friend request from Sugiyem.



09.54 Fri 24 Jun

EXPO GO now

Chattoku's Private Chat

Hi Yemima, you have a new private message from Sugiyem.

-