chatmatch

Team Name:

blueSlip

Proposed Level of Achievement:

Apollo 11

Motivation

With the coronavirus having an impact on student's mental wellbeing, there is a need to create an ideal environment for students who seem to struggle studying alone. Problems faced by students include lack of productivity and motivation to study by themselves at home. The main question we would like to tackle is how do we mimic a school learning environment at home? Before the coronavirus, students in school can reach out to their fellow coursemates with relative ease, even though they may not know each other well beforehand. It becomes a particularly difficult task when their modes of interaction are left with the Zoom chat or Telegram groups.(which some may even argue that its distracting for other parties)

By having a way for students with compatible personalities and are taking the same modules to connect with each other, this serves to help students encourage each other to work harder and maximise productivity.

Aim

We hope that students will be more productive in their studies if they are able to find someone to study with.

User Stories

1. As a student who is revising for exams, I want to be able to focus on studying. Having someone else study with me helps me do that.

2. As a student studying, I would like to study with someone who is taking/has taken the same mods and has a personality that is compatible with me so that we are able to have a meaningful discussion.

3. As a student who has questions and problems with a module, I want to ask my peers first as I may be afraid to ask the prof.

4. As an administrator who wants to prevent abuse of the system, I want to be able to identify abusers, warn them and ban them if they continue to cause problems.

5. As an administrator, I would like to ensure that those using the app are actual students and prevent non-students from creating an account.

Tech Stack

- React Native

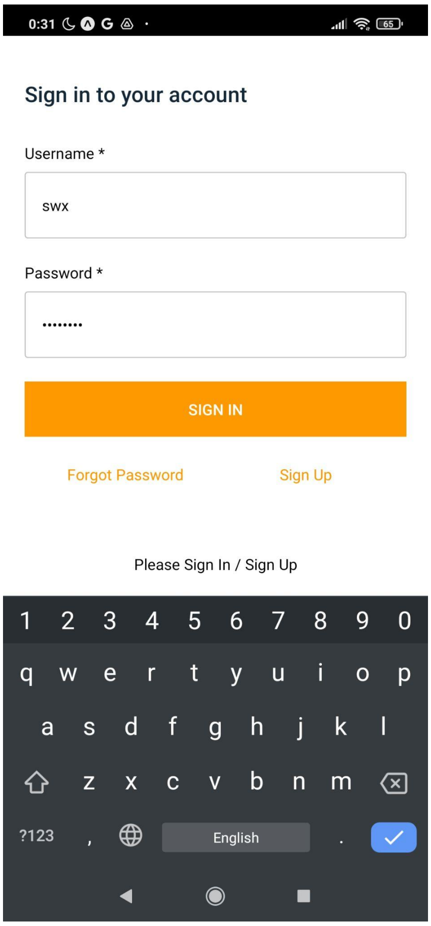
- AWS Amplify & other AWS services
  - Cognito (Authentication)
  - AppSync (GraphQL API service)
  - DynamoDB (NoSQL database)
  - DynamoDB Stream
  - Simple Email Service
  - Lambda

Features

User Authentication

A User authentication service that is built with Amazon Cognito that supports log in, log out and registration. Users will first register within the app by filling in their name, password, email address and phone number. An email verification code will then be sent to the user's email, for the user to input into the registration screen. Only verified users will have access to the app. This also allows users to log in and access their own profile and chat history on different devices.

On Amazon Cognito, the administrator has access to a dashboard that allows additional configurations, such as requiring mobile phone authentication instead or adjustments to password policies. This ensures full customization from the administrator depending on various use cases.



Sign in to your account

Username *

swx

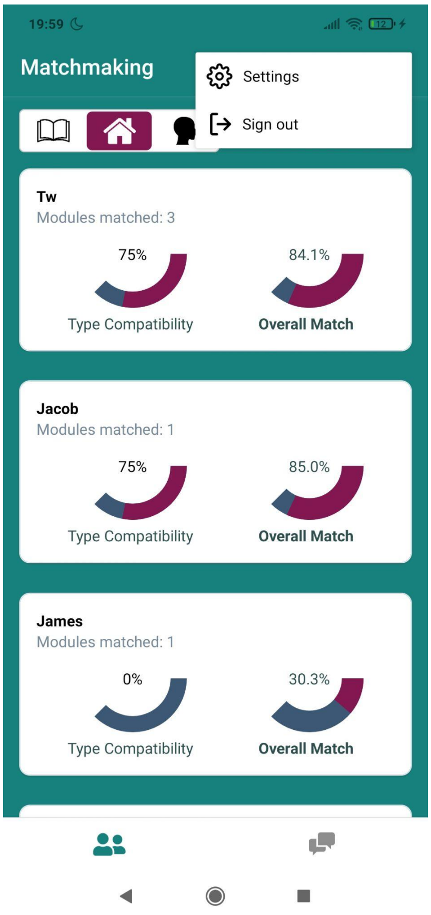
Password *

••••••••

SIGN IN

Forgot Password   Sign Up

Please Sign In / Sign Up



Matchmaking

Settings

Sign out

Tw
Modules matched: 3
75% Type Compatibility
84.1% Overall Match

Jacob
Modules matched: 1
75% Type Compatibility
85.0% Overall Match

James
Modules matched: 1
0% Type Compatibility
30.3% Overall Match

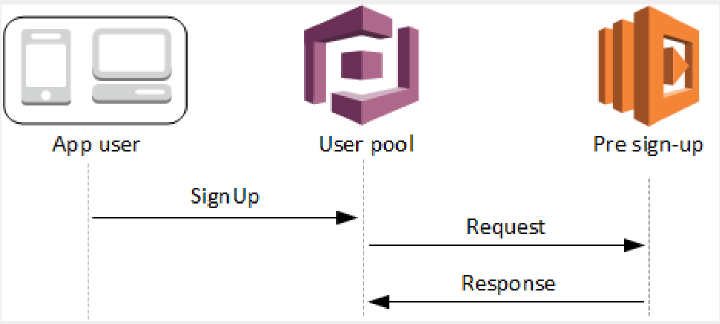
Implementation details

When the app first renders, a check is done to ensure the authenticated user has an entry in the DynamoDB User table. Else, a new entry will be added.

As an additional security step, the email domain (eg. xxxxxx@example.com) used for authentication can be **restricted to specific domains**. This is to ensure the safety of our target audience, students, who will be using the app to communicate with strangers.

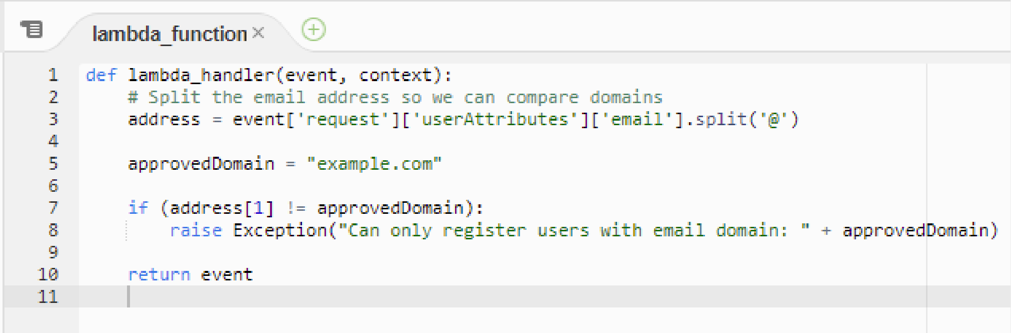
Implementation details

Before Cognito registers a new user, a pre-sign-up Lambda trigger will be used to validate the user's inputted email.



https://docs.aws.amazon.com/cognito/latest/developerguide/user-pool-lambda-pre-sign-up.html

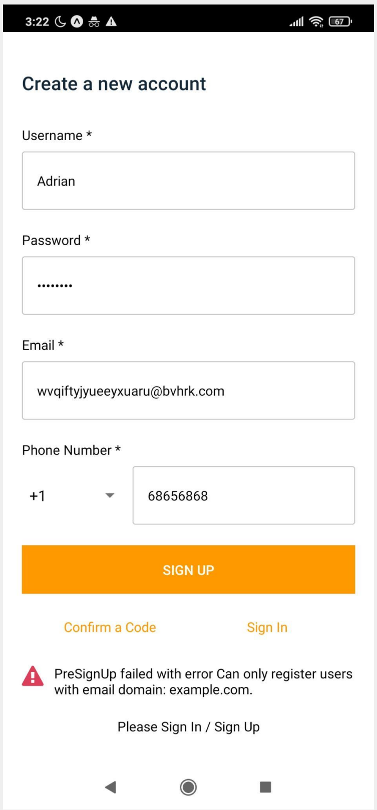
A short Python program is used for this custom validation.



```python
def lambda_handler(event, context):
    # Split the email address so we can compare domains
    address = event['request']['userAttributes']['email'].split('@')

    approvedDomain = "example.com"

    if (address[1] != approvedDomain):
        raise Exception("Can only register users with email domain: " + approvedDomain)

    return event

```

(Screenshot from Lambda console)

For example, we configure the Lambda trigger to only allow "example.com" email domain for registration. This screenshot shows a user registering with a different email domain and is denied registration.



Create a new account

Username *

Adrian

Password *

•••••••••

Email *

wvqiftyjyueeyxuaru@bvhrk.com

Phone Number *

+1 68656868

PreSignUp failed with error Can only register users with email domain: example.com.

Please Sign In / Sign Up

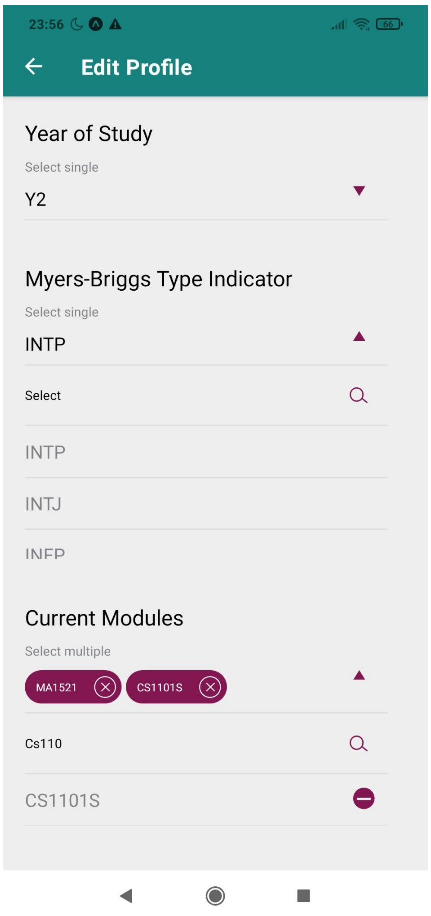
Note: This is currently not enabled.

Edit user profile

From the dropdown menus, users will indicate their **year of study** and **personality type** - from the Myers-Briggs Type Indicator (MBTI).

A multi-select dropdown menu showing a list of modules from NUSMods is available for users to select their **current modules**.

A short questionnaire, comprising of 4 statements, representing the 4 broad categories of user's **interests/hobbies**. As seen from the screenshot below, users will give a rating (Strongly Disagree - Disagree - Neutral - Agree - Strongly Agree) to indicate whether each statement best describes them.



23:56

Edit Profile

Year of Study

Select single

Y2

Myers-Briggs Type Indicator

Select single

INTP

Select

INTP

INTJ

INFP

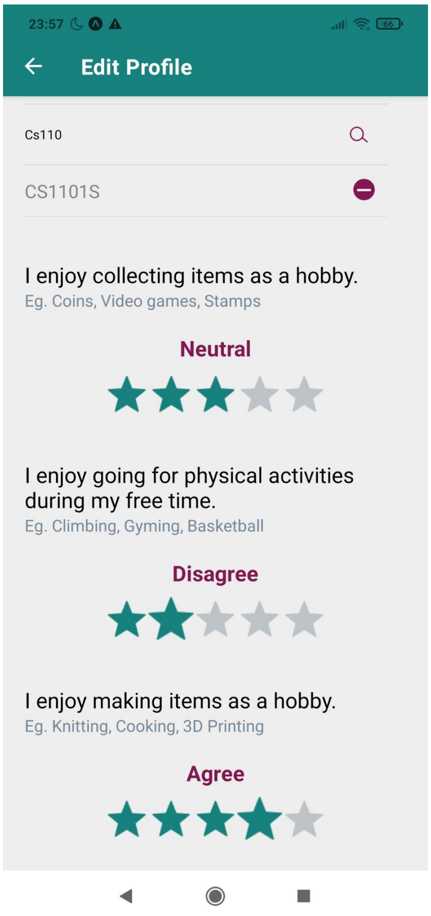
Current Modules

Select multiple

MA1521 ⊗ CS1101S ⊗

Cs110

CS1101S



23:57

Edit Profile

Cs110

CS1101S

I enjoy collecting items as a hobby.
Eg. Coins, Video games, Stamps

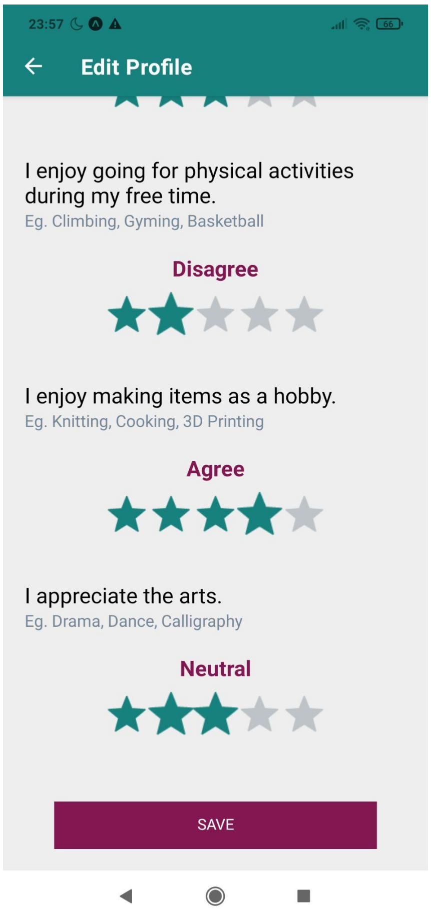
Neutral ★★★☆☆

I enjoy going for physical activities during my free time.
Eg. Climbing, Gyming, Basketball

Disagree ★★☆☆☆

I enjoy making items as a hobby.
Eg. Knitting, Cooking, 3D Printing

Agree ★★★★☆



23:57

Edit Profile

I enjoy going for physical activities during my free time.
Eg. Climbing, Gyming, Basketball

Disagree ★★☆☆☆

I enjoy making items as a hobby.
Eg. Knitting, Cooking, 3D Printing

Agree ★★★★☆

I appreciate the arts.
Eg. Drama, Dance, Calligraphy

Neutral ★★★☆☆

SAVE

Implementation details

An updateUser GraphQL mutation will update the DynamoDB User table to include the user's new details in his entry on the table.

Matchmaking process

The Myers–Briggs Type Indicator (MBTI) categorises each individual to the 4 categories of traits forming 16 possible permutations:

1) Introversion or Extraversion - how one tend to interact with others
2) Sensing or Intuition - how one perceives new information
3) Thinking or Feeling - how one solves problems and make decisions
4) Judging or Perceiving - how one views the the world around them

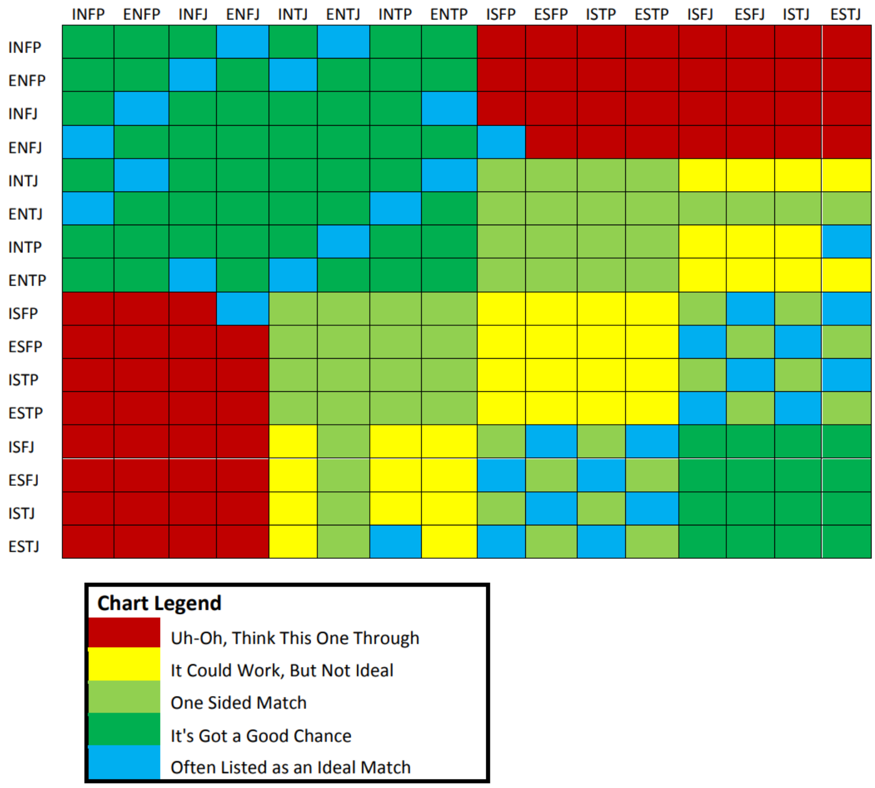
A letter is taken from each category to form the test result, eg. “INFP”, which will represent the personality type of the individual. MBTI is commonly used as a way to see whether 2 partners will suit each other in a relationship or a tool for companies to use for hiring. No single type can match anyone with 100% accuracy, but each one of us probably has a dominant type. Here, we will be using a simplified compatibility chart for our purposes.

For the matchmaking process, **3 match criteria** are used to determine how compatible a user is with another person.

Personality Match

- 2 user’s personality type compatibility, according to the MBTI compatibility chart[1]

Simplified Myers Briggs Type Compatibility Chart



	INFP	ENFP	INFJ	ENFJ	INTJ	ENTJ	INTP	ENTP	ISFP	ESFP	ISTP	ESTP	ISFJ	ESFJ	ISTJ	ESTJ
INFP	🟩	🟩	🟩	🟦	🟩	🟦	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ENFP	🟩	🟩	🟦	🟩	🟦	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
INFJ	🟩	🟦	🟩	🟩	🟩	🟩	🟩	🟦	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ENFJ	🟦	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟦	🟥	🟥	🟥	🟥	🟥	🟥	🟥
INTJ	🟩	🟦	🟩	🟩	🟩	🟩	🟩	🟦	🟢	🟢	🟢	🟢	🟨	🟨	🟨	🟨
ENTJ	🟦	🟩	🟩	🟩	🟩	🟩	🟦	🟩	🟢	🟢	🟢	🟢	🟢	🟢	🟢	🟢
INTP	🟩	🟩	🟩	🟩	🟩	🟦	🟩	🟩	🟢	🟢	🟢	🟢	🟨	🟨	🟨	🟦
ENTP	🟩	🟩	🟦	🟩	🟦	🟩	🟩	🟩	🟢	🟢	🟢	🟢	🟨	🟨	🟨	🟨
ISFP	🟥	🟥	🟥	🟦	🟢	🟢	🟢	🟢	🟨	🟨	🟨	🟨	🟢	🟦	🟢	🟦
ESFP	🟥	🟥	🟥	🟥	🟢	🟢	🟢	🟢	🟨	🟨	🟨	🟨	🟦	🟢	🟦	🟢
ISTP	🟥	🟥	🟥	🟥	🟢	🟢	🟢	🟢	🟨	🟨	🟨	🟨	🟢	🟦	🟢	🟦
ESTP	🟥	🟥	🟥	🟥	🟢	🟢	🟢	🟢	🟨	🟨	🟨	🟨	🟦	🟢	🟦	🟢
ISFJ	🟥	🟥	🟥	🟥	🟨	🟢	🟨	🟨	🟢	🟦	🟢	🟦	🟩	🟩	🟩	🟩
ESFJ	🟥	🟥	🟥	🟥	🟨	🟢	🟨	🟨	🟦	🟢	🟦	🟢	🟩	🟩	🟩	🟩
ISTJ	🟥	🟥	🟥	🟥	🟨	🟢	🟨	🟨	🟢	🟦	🟢	🟦	🟩	🟩	🟩	🟩
ESTJ	🟥	🟥	🟥	🟥	🟨	🟢	🟦	🟨	🟦	🟢	🟦	🟢	🟩	🟩	🟩	🟩

Chart Legend	
🟥	Uh-Oh, Think This One Through
🟨	It Could Work, But Not Ideal
🟢	One Sided Match
🟩	It's Got a Good Chance
🟦	Often Listed as an Ideal Match

[1] https://www.dreamsaroundtheworld.com/mbti-compatibility-guide/

Modules Match

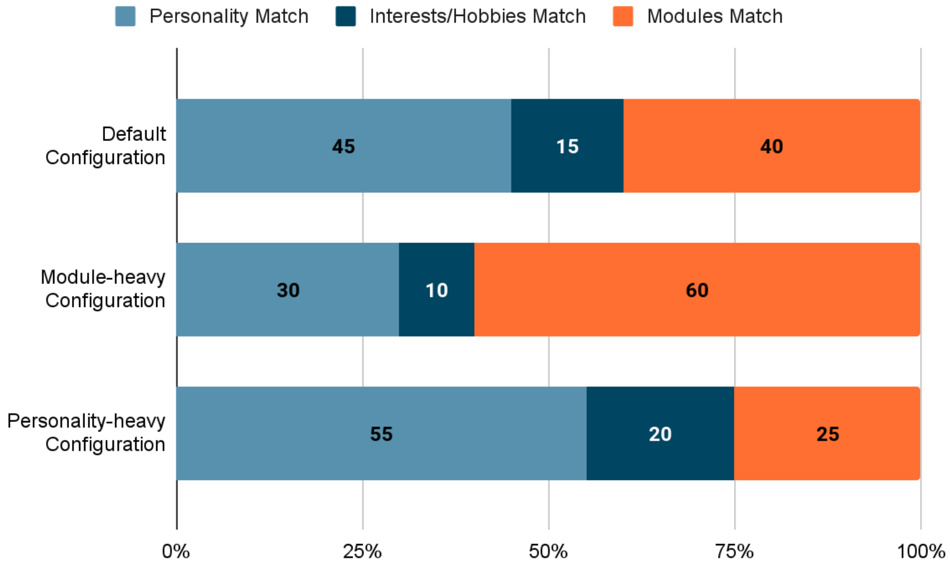
- Percentage of overlapping modules between the 2 users, determined by the user with the lower number of mods

Interests/Hobbies Match

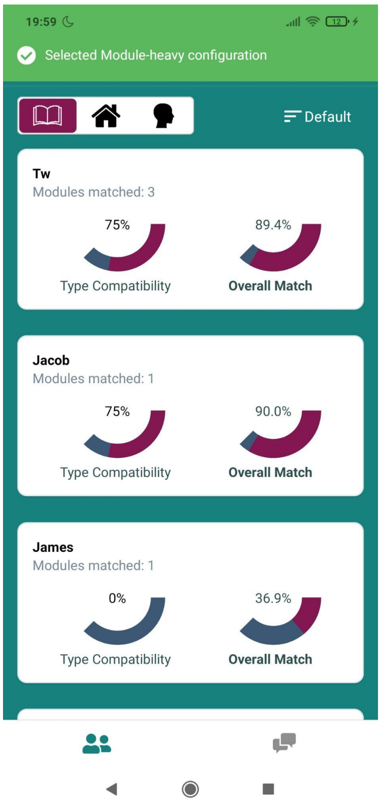
- Similarity percentage out of the 4 broad categories of hobbies (Collecting, Making, Physical activities, Arts)

These 3 criteria will be summed to reflect the overall compatibility between the 2 users.

Weightage sets for the 3 configurations:



Screenshot shows a user selecting the **Module-heavy configuration** and the overall match associated with other users



19:59

Selected Module-heavy configuration

Default

Tw
Modules matched: 3
75% Type Compatibility
89.4% **Overall Match**

Jacob
Modules matched: 1
75% Type Compatibility
90.0% **Overall Match**

James
Modules matched: 1
0% Type Compatibility
36.9% **Overall Match**

Example - **Default** configuration

Student A is currently taking CS1101S, MA2001 and has the personality type of "INTP". Rated "Strongly Agree" (5), "Disagree" (2), "Agree" (4), "Disagree" (2) to the 4 questions respectively.

Student B is currently taking CS1101S, MA1521, GEA1000 and has the personality type of "ENFP". Rated "Neutral" (3), "Disagree" (2), "Strongly Agree" (5), "Agree" (4) to the 4 questions respectively.

Interests/Hobbies Match = (16 - (|5 - 3| + |2 - 2| + |4 - 5| + |2 - 4|)) / 16

= 11/16

Overall Match = Personality Match * weight + Modules Match * weight

+ Interests/Hobbies Match * weight

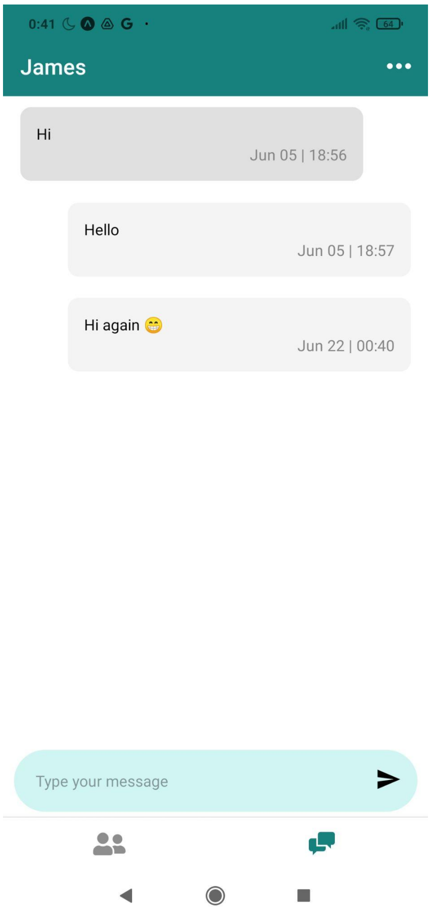
= 75/100 * 45 + 1/2 * 40 + 11/16 * 15

= 64.1% (rounded to 1dp)

Chat Screen

A **real-time chat interface** that allows users to interact with each other after the matchmaking.

- Incoming/outgoing messages are received instantly by both parties in the same chatroom
- Supports text messages and emojis
- Time stamp (local time) indicated at the bottom right corner of each chat bubble

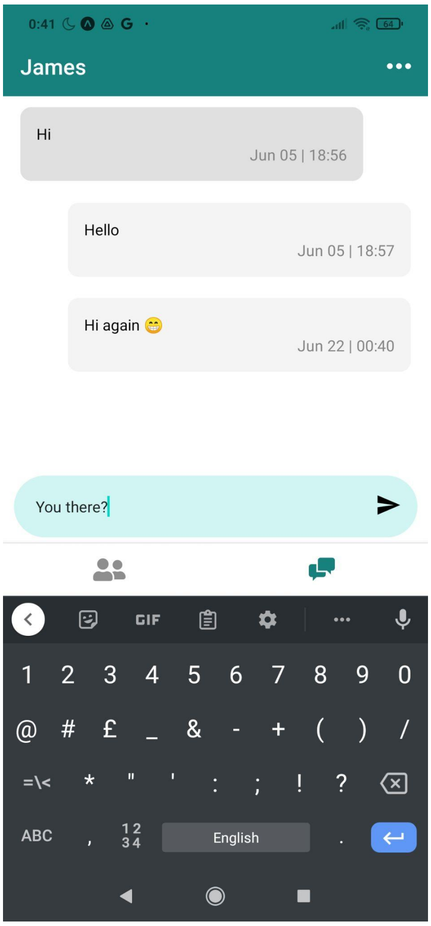


Hi
Jun 05 | 18:56

Hello
Jun 05 | 18:57

Hi again 😁
Jun 22 | 00:40

Type your message



Hi
Jun 05 | 18:56

Hello
Jun 05 | 18:57

Hi again 😁
Jun 22 | 00:40

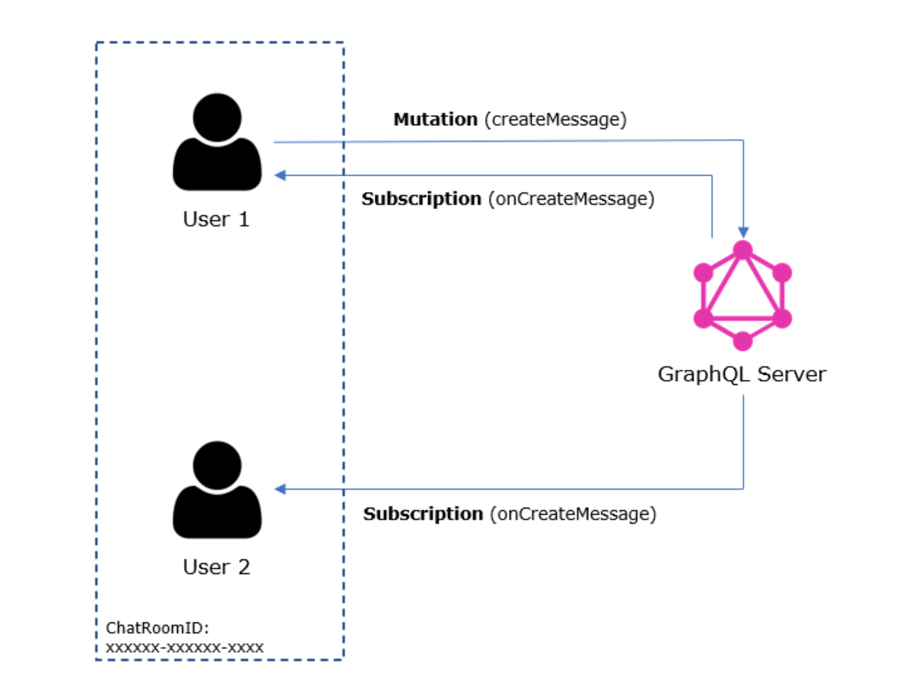
You there?

Implementation details

After a user chooses the person to chat with from the Matchmaking screen, a new chat room will be created with these 2 users if an existing one is not found. Besides adding a new entry to the DynamoDB ChatRoom table, the 2 users will also be added to the ChatRoomUser table, as required for a many-to-many relationship (multiple users are associated with a chat room and multiple chat rooms are associated with a user). If there is an existing chat room with the 2 users, the user will be redirected to that chat room which contains the old chat history.

A custom GraphQL query will then fetch past messages based on the chatRoomID from the ChatRoom table and render them on the screen according to the createdAt time. Once the user types out a message and presses send, a createMessage GraphQL mutation will create a new entry in the Message Table containing the message body and associate it with the correct chatRoomID and userID.

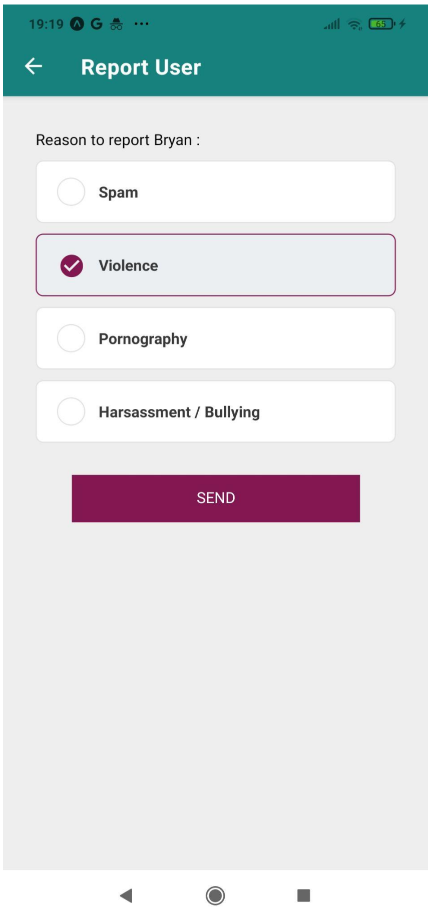
Users in the same chat room will be subscribed to the GraphQL subscription according to the chatRoomID. Both users will receive new incoming/outgoing messages on the screen instantly upon a createMessage mutation.



Real-time chat using GraphQL subscription

Report User

A system to **report inappropriate behaviour**. Since our target audience is students, it is especially important to ensure a safe & conducive environment for students to freely converse with others. From the chat screen, users are able to cite the reason and send a report to the administrator. The administrator will receive the report as an email, as seen from the screenshot below. This report includes the reported user's id and reason for the report, to facilitate further actions from the administrator.



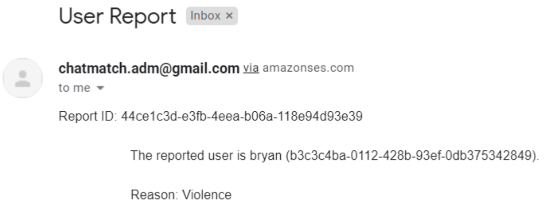
Reason to report Bryan :

○ Spam

● Violence

○ Pornography

○ Harsassment / Bullying


User Report Inbox ×

chatmatch.adm@gmail.com via amazonses.com
to me

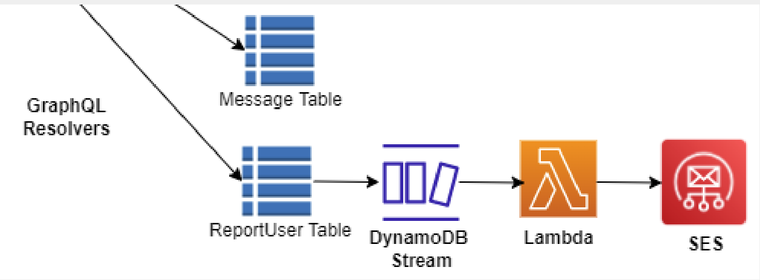
Report ID: 44ce1c3d-e3fb-4eea-b06a-118e94d93e39

The reported user is bryan (b3c3c4ba-0112-428b-93ef-0db375342849).

Reason: Violence

Implementation details

After the user sends a report, a createReportUser mutation will add an entry to the DynamoDB ReportUser Table. DynamoDB Stream will reflect that a new entry has been added, which in turns, triggers the Lambda function to send an email to the administrator through Simple Email Service (SES). The email address is set as an environment variable in the Lambda trigger.



Software Testing

To evaluate the correctness of our application, a series of tests would be done. Tests range from checking if individual components work as intended (**Unit Testing**) to checking if different components work together as a group (**Integration Testing**). **End-to-End Testing** is also conducted to assess that all components in the application works from start to finish. These tests would help us to identify bugs or potential bugs, as well as performance issues in our application.

Unit Testing

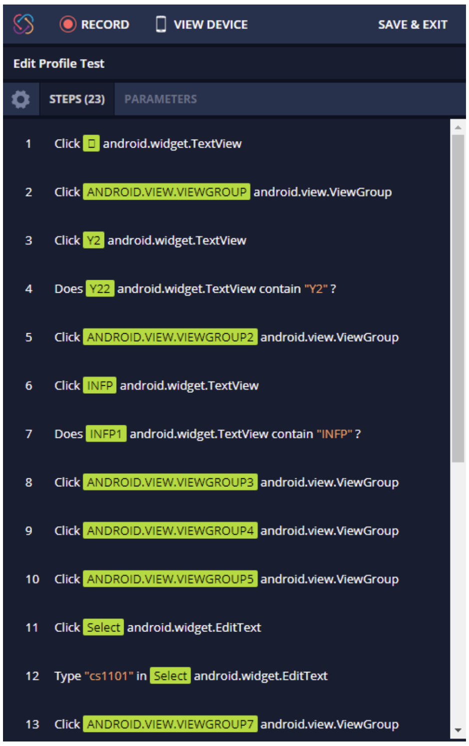
Unit tests are conducted to check whether the individual components behave as expected. For example, a check is done to verify the correctness of the matching algorithm for the Module-heavy configuration according to the weightage (specified from Features section).

Task	**Steps**	**Status**	**Note**
Registration	1. Click sign up 2. Enter username, password, email, phone no. 3. Click sign up 4. Enter email verification code 5. Click confirm	**Pass**	- Verification email received in user's inbox
Login & Log out	1. Enter username and password 2. Click sign in 3. Click 3 dot icon at top right corner 4. Click sign out	**Pass**	
Switch from Default configuration to Module-heavy configuration	1. Click on Module-heavy icon 2. Check if selector has Module-heavy configuration as active	**Pass**	
Switch from Default configuration to Personality-heavy configuration	1. Click on Personality-heavy icon 2. Check if selector has Personality-heavy configuration as active	**Pass**	
Calculation of overall match for Module-heavy, Default, Personality configuration	1. Click on Module-heavy / Default / Personality icon 2. Check if overall match is correct based on formula	**Pass**	
Filter by Name, Overall Match, Personality Type	1. Click on Filter by icon 2. Select 'Name' / 'Overall' / 'Type' option 3. Check all user cards are sorted correctly accordingly	**Pass**	

Integration Testing

Integration tests are conducted to check whether different components work correctly together. For example, a check to see whether all authenticated users from the database are displayed in the Matchmaking screen. In order to speed up our testing process and improve accuracy in tests, TestProject is used for automated testing. TestProject provides an interactive UI to record/code the test cases, before running and verifying the results.

The screenshot here shows the recorded actions for a sample test case that involves checking if the Edit Profile screen captures the user input correctly.



Edit Profile Test

1 Click □ android.widget.TextView

2 Click ANDROID.VIEW.VIEWGROUP android.view.ViewGroup

3 Click Y2 android.widget.TextView

4 Does Y22 android.widget.TextView contain "Y2" ?

5 Click ANDROID.VIEW.VIEWGROUP2 android.view.ViewGroup

6 Click INFP android.widget.TextView

7 Does INFP1 android.widget.TextView contain "INFP" ?

8 Click ANDROID.VIEW.VIEWGROUP3 android.view.ViewGroup

9 Click ANDROID.VIEW.VIEWGROUP4 android.view.ViewGroup

10 Click ANDROID.VIEW.VIEWGROUP5 android.view.ViewGroup

11 Click Select android.widget.EditText

12 Type "cs1101" in Select android.widget.EditText

13 Click ANDROID.VIEW.VIEWGROUP7 android.view.ViewGroup

Task	Steps	Status	Note
Edit and save user profile	1. Click on the edit profile icon 2. Update user's year of study, MBTI, current modules & interests/hobbies 3. Click save 4. Click on the edit profile icon again 5. Check if user's updated year of study, MBTI, current modules & interests/hobbies are correctly fetched & reflected	**Pass**	
All registered users are displayed in Matchmaking screen	1. Find number of users in Matchmaking screen 2. Check if it equates to total no. of registered users - 1 (excluding myself)	**Pass**	

Chat message history displayed	1. Click on user that has sent messages before 2. Verify that all past messages are fetched & displayed	**Pass**	
Send message to user	1. Click on first user from Matching screen 2. Type and send message from input box 3. Check if the message appears as a bubble almost instantly.	**Pass**	
Receive message from other user	1. Click on user that is about to send message 2. Wait and check if message appears on screen without refreshing	**Pass**	

End-to-End Testing

We will be using the following scenario for our evaluation.

> **Scenario 1** - User would like to chat with another user that he/she has the highest overall match with, using the personality-heavy configuration. Afterwhich, the user reports for harassment behaviour.

Steps	**Status**	**Note**
Scenario 1 1. Sign in with username & password 2. Click on Filter by icon 3. Select 'Overall' option 4. Click on personality-heavy configuration icon 5. Click on the first user card 6. Type and send a message 7. Click on the 3-dots icon on the top right 8. Click on Report User 9. Select Harassment/Bullying option 10. Click Send 11. Click on the left icon of bottom bar to return to the Matchmaking Screen 12. Click on the 3-dots icon & Sign out	**Pass**	- Email stating reported user's ID and reason received in administrator's email

Qualitative Evaluation

Cognitive Walkthrough

To evaluate the User Experience(UX) design of our app, a cognitive walkthrough is done to determine the learnability of our app from a new user's point-of-view.

We will be using 2 scenarios for our evaluation.

Scenario 1 - User would like to update his/her user profile.

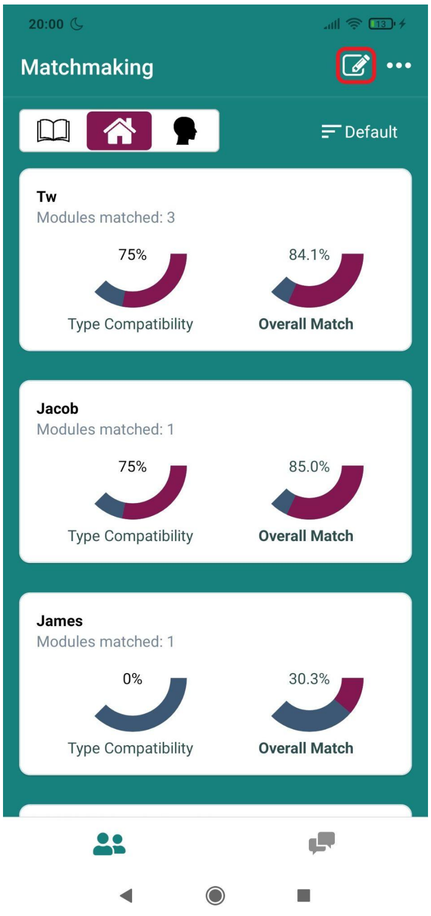
Scenario 2 - User would like to chat with another user that he/she has the highest overall match with, using the module-heavy configuration.

For each scenario, we will be answering 4 questions[2].

Scenario 1

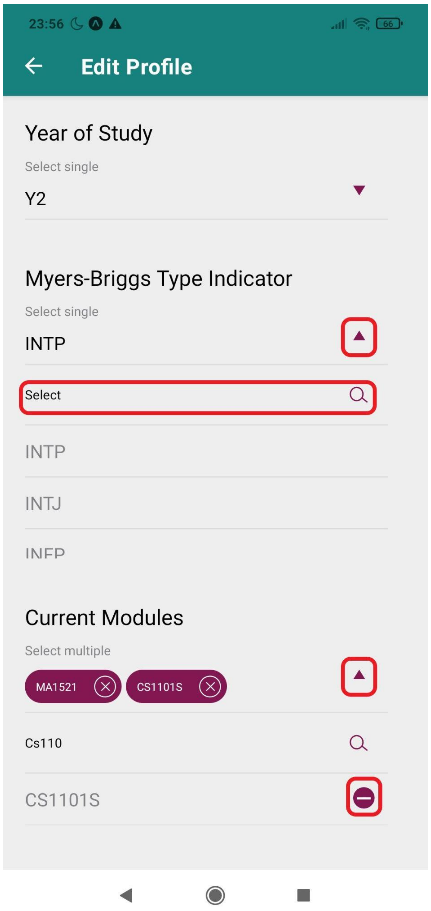
Action 1

Click on the Edit Profile icon.



[2] https://www.nngroup.com/articles/cognitive-walkthroughs/

Action 2 - Enter Year of Study, Myers-Briggs Type Indicator(MBTI), Current Modules and answer 4 questions on hobbies/interests.



23:56

Edit Profile

Year of Study

Select single

Y2

Myers-Briggs Type Indicator

Select single

INTP

Select

INTP

INTJ

INFP

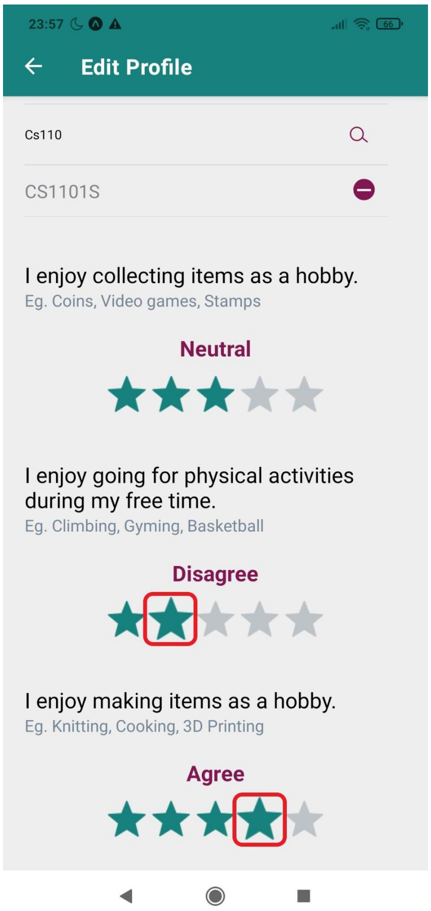
Current Modules

Select multiple

MA1521 CS1101S

Cs110

CS1101S



23:57

Edit Profile

Cs110

CS1101S

I enjoy collecting items as a hobby.
Eg. Coins, Video games, Stamps

Neutral

★★★☆☆

I enjoy going for physical activities during my free time.
Eg. Climbing, Gyming, Basketball

Disagree

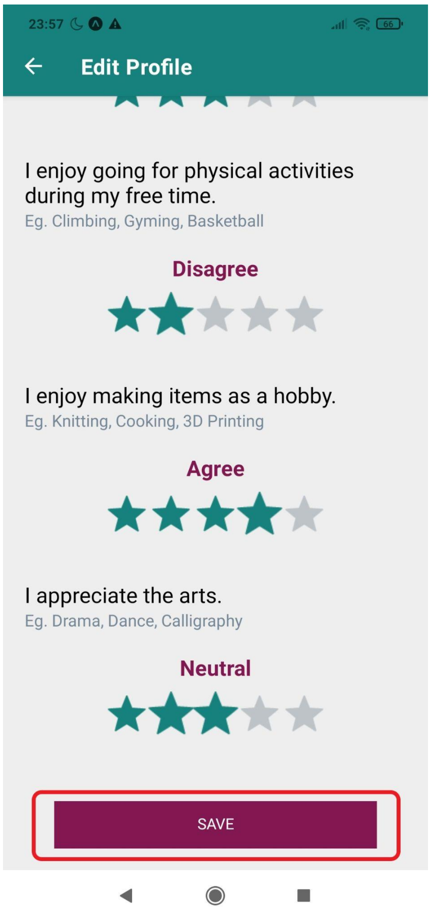
★★☆☆☆

I enjoy making items as a hobby.
Eg. Knitting, Cooking, 3D Printing

Agree

★★★★☆

Action 3
Click on the 'SAVE' button.



23:57

Edit Profile

I enjoy going for physical activities during my free time.
Eg. Climbing, Gyming, Basketball

Disagree

★★☆☆☆

I enjoy making items as a hobby.
Eg. Knitting, Cooking, 3D Printing

Agree

★★★★☆

I appreciate the arts.
Eg. Drama, Dance, Calligraphy

Neutral

★★★☆☆

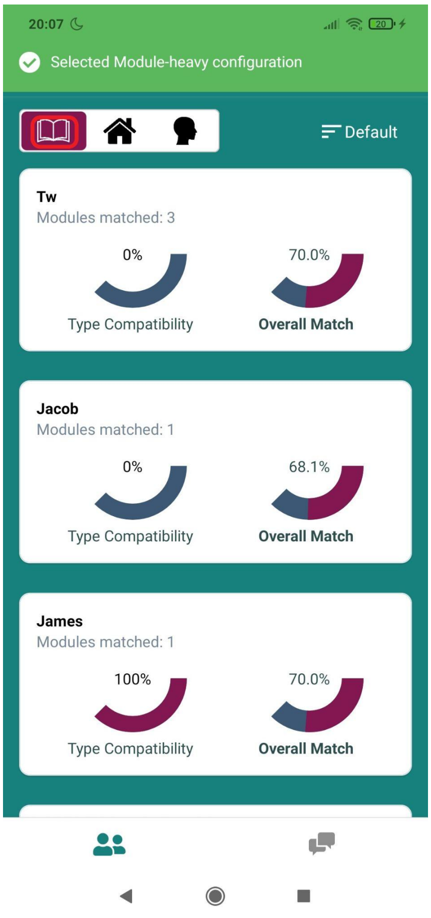
SAVE

Scenario 1		
Analysis Question	**Action**	**Findings**
1. Will users try to achieve the right result?	1	**Yes:** A user would want to edit their profile in order to reflect a better match.
	2	**Yes:** A user would want to complete the profile details and questions in order to update their profile.
	3	**Yes:** A user would want to save the profile details and questions in order to save their updated profile.
2. Will users notice that the correct action is available?	1	**Yes:** An icon showing a pencil and paper is found on the right side of the top bar.
	2	**Yes:** The screen has a scrollable interface, which shows the scroll bar after the user swipes their screen. Each question is associated with a selection box or a rating selection right below.
	3	**Yes:** A large 'SAVE' button at the bottom of the screen.
3. Will users associate the correct action with the result they're trying to achieve?	1	**No:** The icon signifies to the user that it is used to edit something but a new user may not know it is meant to edit your profile.
	2	**Yes:** The downward pointing arrow for the selection boxes signifies to the user that there is a list to select your options from, in order to answer the questions. The default rating is shown for each rating question, users would change the rating to reflect their choices.
	3	**Yes:** The word written in the button is explicit in telling the user its purpose - to save the edits.
4. After the action is performed, will users see that progress is made toward the goal?	1	**Yes:** After clicking, the user will be directed to a loading screen then a screen with an 'Edit Profile' header.
	2	**Yes:** User's selection for each question is reflected immediately in the selection box or shown as a word to describe the rating chosen.
	3	**Yes:** The user will return to the matching screen (original screen) after clicking.

Scenario 2

Action 1

Click on the Module-heavy icon to switch configuration



20:07

Selected Module-heavy configuration

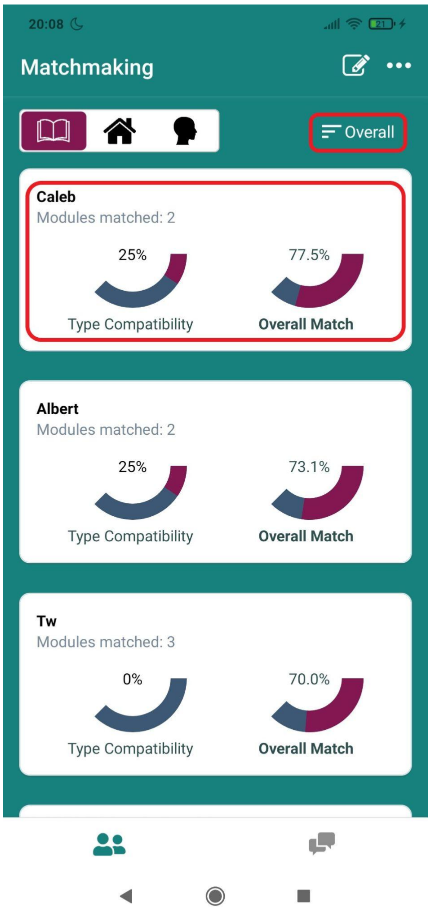
Tw
Modules matched: 3
0% Type Compatibility
70.0% Overall Match

Jacob
Modules matched: 1
0% Type Compatibility
68.1% Overall Match

James
Modules matched: 1
100% Type Compatibility
70.0% Overall Match

Action 2

Click on 'Overall' from the filter. Select the first match which has the highest overall match percentage.



20:08

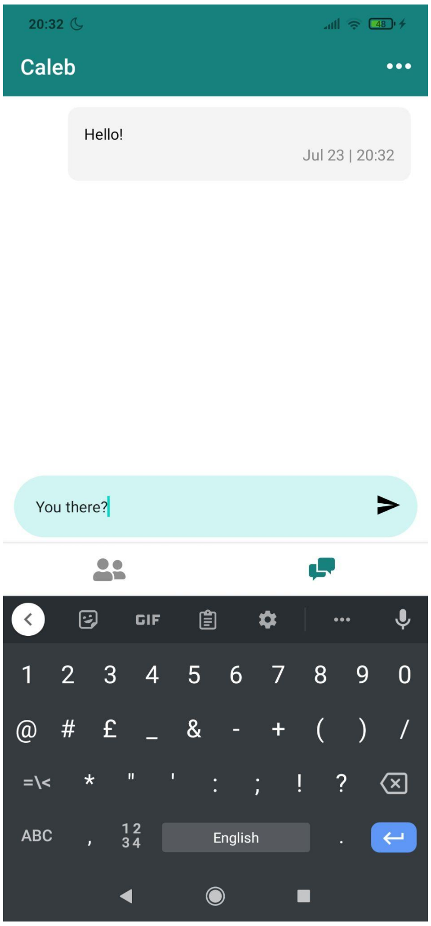
Matchmaking

Caleb
Modules matched: 2
25% Type Compatibility
77.5% Overall Match

Albert
Modules matched: 2
25% Type Compatibility
73.1% Overall Match

Tw
Modules matched: 3
0% Type Compatibility
70.0% Overall Match

Action 3
Type and send a message to the chosen match.



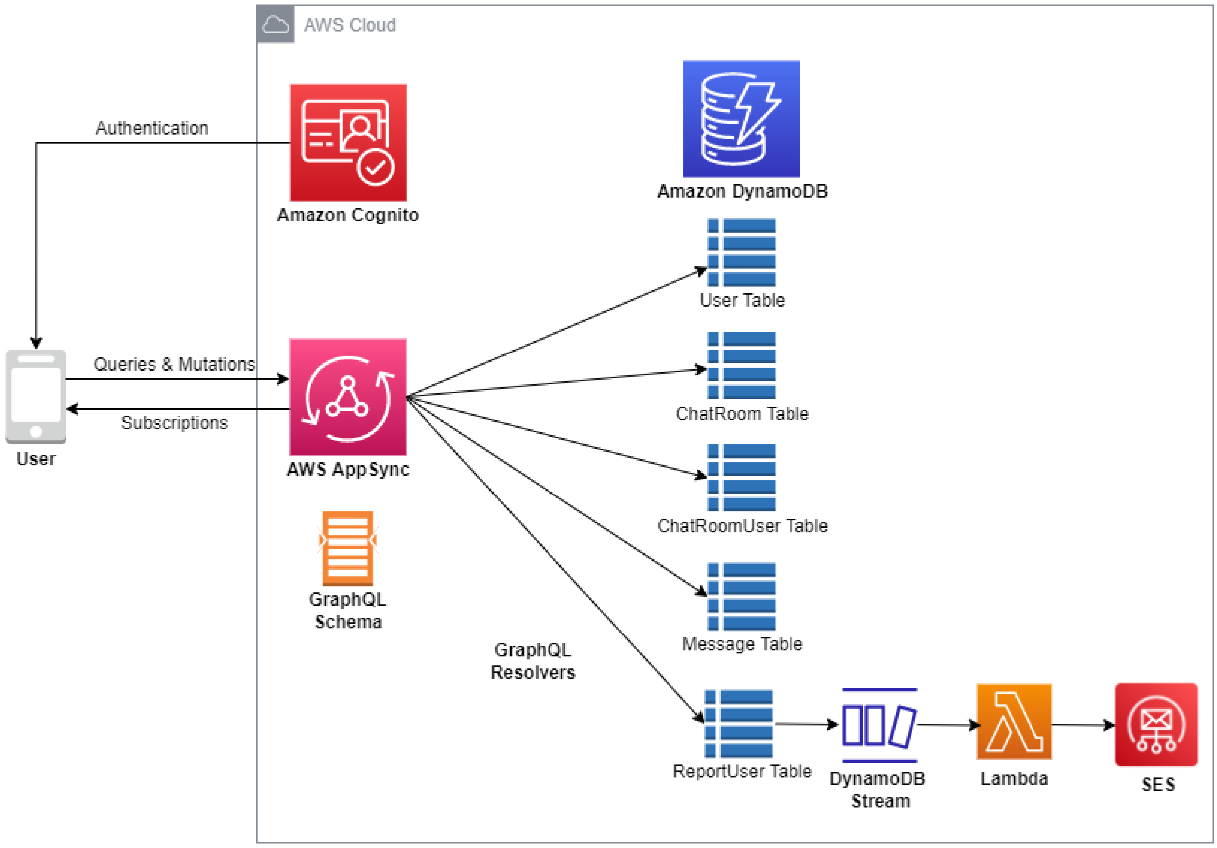
Hello!
Jul 23 | 20:32

You there?

Scenario 2		
Analysis Question	**Action**	**Findings**
1. Will users try to achieve the right result?	1	**Yes:** A user would want to switch to different configurations depending on their preferences.
	2	**Yes:** A user would want to match with a compatible user.
	3	**Yes:** A user would want to chat with the selected match.
2. Will users notice that the correct action is available?	1	**Yes:** A book icon is found in the selector, above the list of other users.
	2	**Yes:** Filter by icon is found on the right side of the configuration bar. The 'Overall Match' description is bolded and the associated percentage wheel is thicker than the other wheel beside it.
	3	**Yes:** An input box with a description of 'Type your message' is found towards the bottom of the screen. A send button is located towards the right of the input box.

3. Will users associate the correct action with the result they're trying to achieve?	1	**No:** The icon may not be clear in signifying to a new user that the selector is meant for users to change configurations. However, a flash message will appear after the user chooses any of the selections. This will direct the user to make a correct action afterwards.
	2	**Yes:** An individual card (containing details like Type Compatibility and Overall Match) is associated with each match. After filter by 'Overall' is selected, the first match shows the highest Overall Match.
	3	**Yes:** The location and description of the input box will signify to the users that messages are typed and sent here.
4. After the action is performed, will users see that progress is made toward the goal?	1	**Yes:** The selector will switch to the option clicked. A flash message will appear temporarily at the top of the screen, stating the configuration selected.
	2	**Yes:** The user will be redirected to a chatroom screen with the match selected.
	3	**Yes:** After the send button is pressed, the user's message will instantly appear as the latest message bubble in the conversation.

Architecture



Security Best Practices

Cognito

Encryption at rest & in transit

User authentication details are encrypted at rest. Requests to Cognito are made over TLS for encryption in transit. Key management is managed by Cognito.

DynamoDB

Use of IAM Roles to authenticate access

IAM Roles are created to gain temporary access to each of the DynamoDB tables by AppSync.

Encryption at rest & in transit

Data, which includes messages and user profile information, stored in the DynamoDB tables are encrypted at rest. Communications to and from DynamoDB are made over TLS for encryption in transit. Key management is managed by DynamoDB.

AppSync

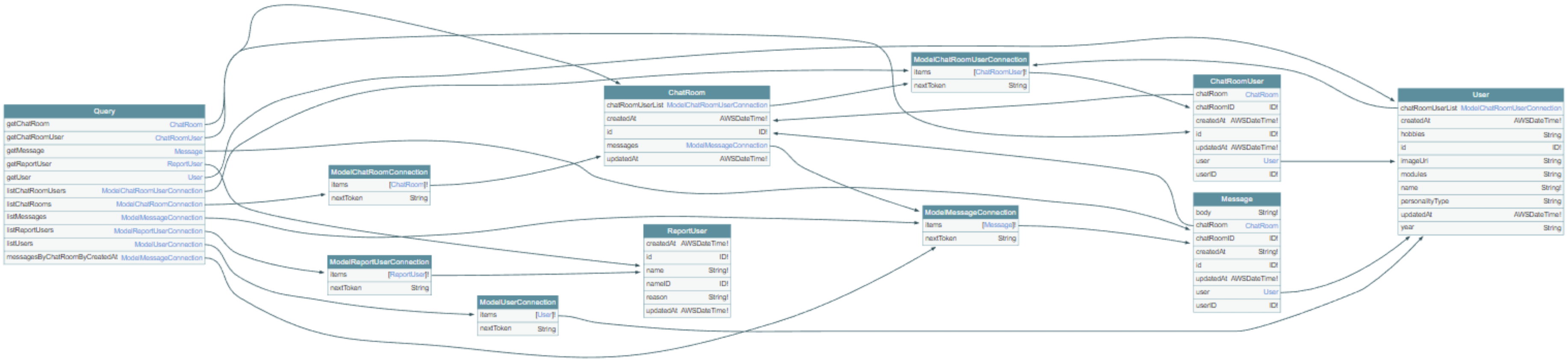
Encryption in transit

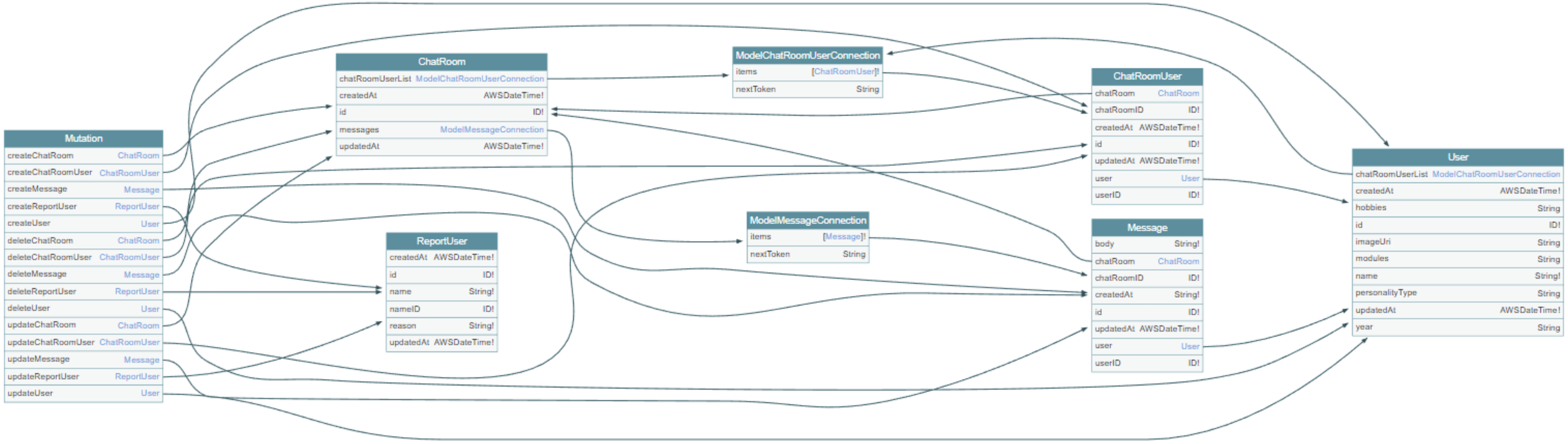
Communication between AppSync and DynamoDB tables are made over TLS for encryption in transit.

User Authentication

Cognito User Pool Tokens are used to restrict calls to GraphQL schema by authorised users, through the use of `@auth(rules: [{ allow: private }])` transform.

Schema Diagram





User Feedback

After gathering the feedback from users who tested the application, some additional features and enhancements were highlighted.

Feedback	**Status**	**Note**
Not satisfied with the current weightage of the 3 components in calculating Overall Match	**Implemented**	Added 2 new configurations (Personality-heavy & Module-heavy), aside from Default configuration. Overweighting Personality type match & Modules match respectively
Difficulty in searching for highest overall match, as the list of users are in random order	**Implemented**	Added a function to filter by Overall Match, Name and Personality Type Match
Form chat groups with multiple users	**Future Progress**	
Block other users	**Future Progress**	
Ability to send file attachments in chat	**Future Progress**	