

Assignment T5: Second Iteration

- TeamShiba
 - Jianing Li (jl5543)
 - Yifei Wang (yw3229)
 - Eurey Noguchi (yn2377)
 - Chiqu Li (cl3895)

Github repository

<https://github.com/teamshiba/shiba>

User stories

Login

As a user, I can login to the app so that I can create or join a group and start voting.

My conditions of satisfaction are:

- I can successfully register and login with my gmail in the login page and I will be directed to my group lists.
- I can register and login using my email address

Case 1:

Sample input: One user at the sign up page and sign up with Gmail.

Sample output: This user would sign up successfully and would be successfully directed to the group list page.

Case 2:

Sample input: A user enters the wrong email or password

Sample output: The user is unable to login and is asked to try again

Creating a Group

As a user (organizer of a group), I want to create a new group where other users can join to vote.

My conditions of satisfaction are:

- I am able to create a new empty group
- I am able to name my group

Case 1:

Sample input: A user clicks on the create group button and inputs a name for the group.

Sample output: A new group is created and is displayed in the room list.

Case 2:

Invalid input: A user clicks on the create group button but tries to create a group without a name

Sample output: The user is unable to create the group and nothing is displayed on the room list.

View Room List:

As a user, I want to be able to see a list of groups that I am currently a part of or was a part of so that I can immediately access the voting page and the voting statistics.

My conditions of satisfaction are:

- I can find all of my joined groups as a list
- I can see groups that have already ended in a separate list

Case 1:

I am able to see rooms that I have created and I have joined in a list.

Sample input: The user clicks on the add group button to create a group

Sample output: The created group appears in the list in the Rooms tab

Case 2:

If the user has not joined nor created any group, then they should not be able to see any groups in the list.

Simple input: The user is new.

Sample output: The room list is empty and the user cannot see any groups

Invite friends/family to match

As an organizer of an event, I want to send a link to my group of friends/family so that they can join the group and vote for their preferences in my particular group.

My conditions of satisfaction are:

- I am able to generate a link for my specific group
- Other users can use this link to join my group
- When other attendees join my group, I should be able to see their names to identify who is currently in the group
- After other attendees have joined the group, they should be able to start swiping
- If the person who received the invitation link is not logged in or registered, they should be redirected to a login/registration page.

Case 1:

After a user successfully creates a group, they should be able to see an invitation link created that they can copy/paste to share to other people that should join the group. After another user receives the invitation link, they should be able to click on it and be redirected to and join the group that invited them. Once the user joins the group, they should be able to immediately start swiping to indicate their preferences on the restaurants.

Sample input: The organizer of the group creates a group

Expected output: There will be a unique link generated and displayed in the group profile page, which other users can use to join the group.

Case 2:

For each invitation link, no duplicate users should be able to join the same group. For example, if a user clicks on the link to join the group and successfully joins the group, then when they click on the same invitation link as the same user, they should not be able to join the group since they are already in the group.

Sample input: The same user access the same invitation link multiple times

Expected output: The user can only join the same group once and therefore, in the members list in the group profile page, we will only observe one instance of this user.

Case 3:

For a person that has not registered for the app, when they open the invitation link, they should be redirected to the login/registration page. After they finish registration and successfully logged in, they can access the link again to join the group.

Sample input: An unauthenticated user tries to join a group

Expected output: The user is unable to join the group and is redirected to the login page. After the user is logged and accesses the URL, they are able to join the group.

Voting

As a user, I want to express my preference by a simple action (like swiping, clicking) so that I can quickly vote for all options.

My conditions of satisfaction are:

- On the matching page (where information and images about an option is presented) on a mobile device, a user can swipe right to 'like' or left to 'dislike' it.

Test case 1:

When a user enters a match room through a link, he can start voting for the options already in the room one after another.

Sample input: one user clicks the link, enters the room, and swipes left. There're more than one items in the list.

Expected output: the matching page is displayed after he enters the room. After he swipes, the next item is displayed.

Test case 2:

When a user enters a match room where no items have been added to the list, he cannot vote.

Sample input: one user clicks the link and before that, the host didn't add any items to the list.

Expected output: the page is empty, with only the "add new item" button in the corner, and an image telling the user to click that button to add items.

Test case 3:

When a user enters a match room where no items have been added to the list, he cannot vote.

Invalid input: when a user enters a matching room, he swipes up.

Expected output: nothing happened.

Adding my own options

As the attendee of an event, I want to also add my own choice of restaurant to the list so that if I know a restaurant that is great but is not in the list or if I don't like any of the restaurants, I can add my own suggestions.

Consensus

As an attendee of an event, I want to know if my group has a consensus (aka. a match) so that I can see which restaurant the group wants to go to.

My conditions of satisfaction are:

- If all members of that group have 'liked' the same restaurants in the list, when anyone clicks the link and enters the room again, the agreed restaurants would be displayed.

Case 1:

When all members of a group agree on the same restaurants, the system should present the matching result to every member.

Sample input: 3 users in the group swipe right for the same restaurant. All attendees stay on that page after they completed their own voting process. All of them have voted for the same restaurants (let's say "Shake Shack").

Expected output: the same page shows that matching results after the matching ends on every members' device. The matching results are the intro of "Shake Shack".

Case 2:

When members of a group cannot agree on their preferences, the screen should say there are no matches.

Sample input: 3 users in the group swipe right for some restaurant. All attendees stay on that page after they completed their own voting process. They voted for the different restaurants (let's say user 1 for "Shake Shack", user 2 for "five guys", user 3 swiped left for both of them).

Expected output: the same page shows that there were no matches. The matching results may be seen in the statistics page.

Statistics table

As an attendee of an event, I want to have a table showing how many people chose each option so that I can know about other people's choices and select an option manually when no exact match could be made in the end.

My conditions of satisfaction are:

- A statistics on how many people chose each option will pop up to every member of the group if the group has finished matching.
- The statistic could also be shown when the user clicks the "Statistics" button during matching

Case 1:

Sample Input: All group members finished swiping all the restaurants.

Sample Output: The user will be prompted to click the statistics button to see the results. Restaurants that everyone agrees will be marked out.

Case 2:

Sample Input: During a match with multiple users, one of the users click the "Statistics" button

Sample Output: The user will see the correct information of how many people choose each option correctly.

Case 3:

Sample Input: No user has voted for anything, one of the users click the "Statistics" button

Sample Output: The statistic graph will show all items with counts of 0.

View Consensus in Completed Room:

As a user, I want to be able to see which items reached a consensus in a completed room so that I can know where this group should go after the voting has ended.

My conditions of satisfaction are:

- In a completed room, I am able to see an item that has reached a consensus by members in the group
- I am able to swipe the item card left or right to show the next matched item

Case 1:

Sample Input: All group members finished swiping and the organizer marks the group as completed. In the swiping process, all group members liked two of the items.

Sample Output: In the completed room, we see the item cards for the two items. Initially we see one card but when we swipe either left or right, we can see the second matched card.

Case 2:

Sample Input: All group members finished swiping and the organizer marks the group as completed. In the swiping process, all group members disliked all the items.

Sample Output: In the completed room, no item cards will be displayed as there was no consensus reached for any item.

Edit Profile:

As a user, I can go to my user profile page and see my information.

My conditions of satisfaction are:

- When I am in the app, I can go to my user page and update my user name.

Case 1:

When a user signs in to the app, he can go to my profile and update my information. Once he updates the info, his info will permanently change.

Sample input: One user goes to the personal file page, updates the user name.

Sample output: The name of the user changed in the user profile page.

Case 2:

When a user doesn't login, he cannot access his profile.

Invalid input: A user enters the URL for profile edit in browser without login.

Sample output: The user cannot access the profile.

Logout:

As a user, I can logout from the app so that the group info is no longer accessible.

My conditions of satisfaction are:

- I can click the logout button on my profile page and the app is no longer accessible until I logged in again.

Case1:

Sample input: One user at the user profile page and click logout button.

Sample output: This user would sign out successfully and be directed to the sign up page. He can't access shiba content until logged in.

There are no invalid inputs in this case since either user presses the logout button or not.

Test plan

The automated test suite

Front-end: separated.

Backend: <https://github.com/teamshiba/shiba/tree/main/service/tests>

Scripts to run the test: <https://github.com/teamshiba/shiba/blob/main/Makefile>

Web App (react.js)

Voting for an item

Page for users to swipe/click a button to indicate their preference on an item. Users can swipe right, left, up, down, or click the like or dislike button.

Equivalence partitions & boundary conditions

1. Direction of the card swipe to indicate preference
 - Card is swiped right
 - Test case: result screen (swipe right)
 - Card is swiped left
 - Test case: result screen (swipe left)
 - Card is swiped up or down
 - Test case: result screen (swipe up / down)
2. Clicking button to indicate preference
 - Like button is clicked
 - Test case: result screen vote by click (like)
 - Dislike button is clicked
 - Test case: result screen vote by click (dislike)

Number of items in a group

There can be 0, less than 0, or greater than 0 number of items in the group. The boundary conditions would be -1, 0, 1.

Equivalence partitions & boundary conditions

- There are more than 0 items in the group (1 item)
 - Test case: renders (in Voting directory)

- There are 0 items in the group
 - Test case: voting screen (with no items)
- There are less than 0 items in the group
 - Not handled since it is not possible to mock negative numbers of items

State of the group

The group can be completed or not completed. In addition, the group may have more than 0 matches or 0 matches. The boundary condition would be 1, 0, -1 matches.

Equivalence partitions & boundary conditions

- Group is completed with no match
 - Test case: renders completed (no match)
- Group is completed with match
 - Test case: renders completed (with a match)
- Group is completed with multiple matches
 - Test case: renders completed (with multiple matches)
- Group is not completed with no items (newly created group)
 - Test case: voting screen (fresh group)
- Group is not completed with no items (swiped through all items)
 - Test case: No more items to swipe
- Group is not completed with items
 - Test case: renders (under Voting directory)

Sign Out

A logged in user can sign out

Equivalence partitions & boundary conditions

- User is logged in
 - Test case: sign out
- User is not logged in
 - User is unable to access this page

Statistics

There may be 0 or more items in a group, which we should display the statistics for. The boundary condition would therefore be -1, 0, 1 items.

Equivalence partitions & boundary conditions

- There are more than 0 items
 - Test case: renders (in Statistics directory)

- There are 0 items
- There are less than 0 items
 - Not handled since it is not possible to mock negative numbers of items

Group member list

A group can have several members. The boundary condition would be -1, 0, 1 members in a group.

Equivalence partitions & boundary conditions

- There are more than 0 members (1 member)
 - Test case: renders (in RoomProfile directory)
- There is 0 or less members
 - Not handled since it is not possible to have a group with less than or equal to 0 members

Member management

The organizer of the group is able to appoint another member as the organizer or delete other members from the group.

Equivalence partitions & boundary conditions

- An organizer can manage other member's role and delete other members
 - Test case: open menu for organizer (but not himself)
- A non-organizer cannot manage other members
 - Test case: don't open menu for non-organizer

Room list

A room can be in different states: active or history.

Equivalence partitions & boundary conditions

- Active room list
 - Test case: renders active rooms
- History room list
 - Test case: renders history rooms
- Creating a group
 - Test case: modal creates a group

Invitation

When a user enters the invitation page, they can either accept or decline the invitation.

Equivalence partitions & boundary conditions

- Accept invitation
 - Test case: accept invitation
- Decline invitation
 - Test case: decline invitation

Log In

A user is able to login using their account or if they are already logged in, they can directly access the app.

Equivalence partitions & boundary conditions

- User has not logged in so needs to login to use the application
 - Test case: renders (in Authentication directory)
- User has already logged in so will be redirected to the home screen
 - Test case: redirects if logged in

Adding items to the group

When a user wants to add an item to the group, they can either choose from the recommendation list or search for it. The boundary condition here would be on the length of the search string. This means that the boundary would be -1, 0, 1.

Equivalence partitions & boundary conditions

- Search field is empty so should show recommended items
 - Test case: renders recommendation
- Search field contains more than 0 characters so should show searched items
 - Test case: renders search result
- Search field contains less than 0 characters
 - Not handled since it is not possible to type less than 0 characters

Service (flask.py)

Create group API

The API endpoint handler used to create a new matching room.

Equivalence partitions & boundary conditions

- The request body contains a 'roomName' of any string value. (valid input)
 - Test case: `"test_routes TestRoom test_create_group_pass"`
- The request body doesn't contain an attribute named 'roomName'.
 - Test case: `"test_routes TestRoom test_create_group_fail"`
- The request body is empty.
 - Test case: `"test_routes TestRoom test_create_group_fail"`

Join group API

Equivalence partitions & boundary conditions

- The request URL contains a path parameters of existent 'groupid'. (valid input)
 - Test case: *"test_routes TestRoom test_create_group_pass"*
- The request URL does not contain a path parameter.
 - Test case: *"test_routes TestRoom test_create_group_fail"*
- The request URL contains a path parameters of inexistent 'groupid'.
 - Test case: *"test_routes TestRoom test_create_group_fail"*
- The user is not logged in'.
 - Test case: *"test_routes TestRoom test_create_group_fail"*

API: Get a list of rooms

Equivalence partitions & boundary conditions

- The query parameters include a "state" attribute, and the user is logged in. (valid input)
 - Test case: *"test_routes TestRoom test_get_group_list_pass"*
- The user is not logged in.
 - Test case: *"test_routes TestRoom test_get_group_list_fail"*

API: Get the profile

Equivalence partitions & boundary conditions

- The query parameters include a "gid" attribute and there exists a corresponding matching group. (valid input)
 - Test case: *"test_routes TestRoom test_get_group_profile_pass"*
- The query parameters do include a "gid" attribute but there doesn't exist a corresponding matching group.
 - Test case: *"test_routes TestRoom test_get_group_profile_fail"*
- The query parameters does not include a "gid" attribute but there doesn't exist a corresponding matching group.
 - Test case: *"test_routes TestRoom test_get_group_profile_fail"*

API: Get items of one group

Equivalence partitions & boundary conditions

- The query parameters include a "gid" (target group ID) attribute, and the current user is a member of the target group. (valid input)

- Test case: *"test_routes TestItem test_get_item_list_pass"*
- The query parameters include a "gid" (target group ID) attribute, but the current user is not a member of the target group.
 - Test case: *"test_routes TestItem test_get_item_list_fail"*
- The query parameters do come with a "gid" or "group_id" (target group ID) attribute.
 - Test case: *"test_routes TestItem test_get_item_list_fail"*

API: Remove a user from a matching group

Equivalence partitions & boundary conditions

- The request URL contains a path parameters of existent 'groupid' and current user is a member of that'. (valid input)
 - Test case: *"test_routes TestItem test_remove_user_pass"*
- The request URL contains a path parameters of existent 'groupid' but current user is not a member of that.
 - Test case: *"test_routes TestItem test_remove_user_fail"*
- The request URL does not contain a path parameter.
 - Test case: *"test_routes TestItem test_remove_user_fail"*

API: Add an item to one group

Equivalence partitions & boundary conditions

- The request body contains a 'groupid' and an 'item' object. (valid input)
 - Test case: *"test_routes TestItem test_add_item_pass"*
- The request body doesn't contain an attribute named "groupid".
 - Test case: *"test_routes TestItem test_add_item_fail"*
- The request body doesn't contain an attribute named "item".
 - Test case: *"test_routes TestItem test_add_item_fail"*

API: Put a vote

Equivalence partitions & boundary conditions

- The request body contains 'groupid', 'item_id', 'type', 'auth_uid'. (valid input)
 - Test case: *"test_routes TestVoting test_put_a_vote_pass"*
- The request body doesn't contain an attribute named "groupid".
 - Test case: *"test_routes TestVoting test_put_a_vote_fail_gid"*
- The request body doesn't contain an attribute named "auth_uid".
 - Test case: *"test_routes TestVoting test_put_a_vote_fail_valid_user"*
- The request body doesn't contain an attribute named "item_id".
 - Test case: *"test_routes TestVoting test_put_a_vote_fail_item_id"*
- The request body doesn't contain an attribute named "type".
 - Test case: *"test_routes TestVoting test_put_a_vote_fail_v_type"*

- The request body contains invalid "type".
 - Test case: *"test_routes TestVoting test_put_a_vote_fail_v_type_num"*
- Target item not in the group list.
 - Test case: *"test_routes TestVoting test_put_a_vote_fail_item_list"*
- User not in the group members.
 - Test case: *"test_routes TestVoting test_put_a_vote_fail_members"*

Subroutine: request data from Yelp

Tested method:

`utils.external.yelp_search_biz`

Equivalence partitions

- Either a location name, e.g. "New York City", or valid targeted coordinates, i.e. longitude & latitude, are passed in. (valid input)
 - Test case: *"test_yelp_search_biz_pass"*
- Neither a location name or a pair of coordinates are passed in.
 - Test case: *"test_yelp_search_biz_fail"*
- Invalid location coordinates are passed in.
 - Test case: *"test_yelp_search_biz_fail"*

Boundary analysis

- Boundaries for valid location coordinates are [latitude=90, longitude=180], [latitude=-90, longitude=180], [latitude=-90, longitude=-180] [latitude=90, longitude=-180].
- Any input values within the above range (defined by for corner values) are valid. Though, only a small subset would come with a response of non-empty results from Yelp.

Branch coverage report

Reports for front end part:

<https://github.com/teamshiba/shiba/tree/main/reports/frontend-coverage>

Reports for back end part:

<https://github.com/teamshiba/shiba/blob/main/reports/backend-coverage>

Continuous integration

CI configuration: <https://github.com/teamshiba/shiba/tree/main/.github/workflows>

CI reports: <https://github.com/teamshiba/shiba/actions>