Team Meeting: ELEN4010 Study Group Coordinator

26 June 2021

Time: 15:50 -18:00

Online: Microsoft Teams

Type of meeting:	Sprint Review and Retrospective
Note taker:	All

Attendees:	Tarryn Maggs, Taliya Weinstein, Yasser Karam, Nathan Jones, Basheq Tarifi

Minutes

Agenda item:	Sprint Review Session	**Presenter:**	All

Discussion:

15:50 - 16:00 Scrum Master Introduction: Yasser

Brief overview of the scope covered by the sprint as well as the user stories implemented.

16:00 - 16:10 Basheq

User Stories:

I completed the following user stories this sprint:

- Select an automatically suggested meeting location #136
- Have the option to fill in additional fields during setup #160
- Vote to invite a proposed new member #161

The following additional features were incorporated: the ability to view automatically recommended locations for face to face meetings based on the group and user. There are three recommendations made:

- The address of member of the group which would minimise the distance travelled by all members
  - I used the Google Maps Distance Matrix API to achieve which returns the distance between users addresses by home. If a user has no address, they are left out of the calculation.
- The address of the user creating the meeting
- The address of the University of the Witwatersrand, Johannesburg

In order to implement the above functionality, features allowing a user to add an address on registration were implemented. Additionally, improvements showing error messages in the login and registration pages were implemented. On the note of polls, they are now posted to the database once they are completed and poll history is shown on the 'Completed polls' tab. Furthermore, invite polls can be started

Additionally, some styling improvements were made throughout the app and minor fixes to previous stories were made.

Testing

Tests for registration, logging in and polling functionality (v4.0)

The following end to end functionality was tested using Cypress:

Registration and Logging in

- Registration of an invalid (existing) user fails
- Registration of a new user succeeds
- A newly registered user can log in
- An existing user can log in with the correct details
- Logging in with an incorrect username or password produces an appropriate message
- A user can log out
- A page requiring authentication cannot be accessed without being signed in

Polls

- A user can search for another user to invite into a group and start a poll
- Users can vote on the invite poll
- The invite poll's outcome is saved and action is taken, inviting the user only if there is a positive outcome
- A user can view group requests and start a poll to accept one
- Users can vote on the group requests poll
- The request poll's outcome is saved and action is taken, adding the user to the group if there is a positive outcome
- A user can view group members and start a poll to ban any of them
- Users can vote on the banning poll
- The banning poll's outcome is saved and action is taken, removing the user from the group if there is a positive outcome
- Creating a new custom poll
- Voting on a custom poll
- The custom poll's outcome is saved

Notes:

Major delays were encountered when testing, due to some unfamiliarity with cypress. Working with the Google matrix API was interesting and good to learn.

16:10 - 16:20 Yasser

User Stories:

I completed the following user stories this sprint:

- View which member performed the action #142
- View the nature of the action #141
- View the timestamp of the action #140

Features:

I included the ability to log several specific actions in groups into the `action_log` table in the database. Each of these actions are assigned to a specific user, time stamp, group, and are accompanied by a description that is used when displaying a group chat's `Activity Log`. The following are the actions that are used and listed in the `actions` table:

- POLL: This includes the start and end of polls for - invites, banning users, group requests, custom polls
- INVITE: On group creation, and when a poll outcome is a 'Yes' to invite a user to the group

- CREATED: On group creation
- MEETING: The Time and Place for a meeting are logged when a meeting is created
- SCREENING: The result of a screening is logged for every group a user is a member of
- MESSAGE: Messages sent on groups are logged
- ENTER: When a user enters a group chat
- LEAVE: When a user leaves a group chat

Improvements:

I limited the create-group functionality so that the users can only be a part of a maximum of 10 groups. I also changed the group creation so that the naming choice must be unique - thereby creating/ensuring unique groups.

Notes:

Logging actions was tough to do when people are going to make changes to the files I am inserting my functions in. Luckily the merging went smoothly.

Testing:

Action log testing (v4.0)

The following end to end functionality was tested using Cypress:

Action logging

- Create activities to be logged
- Activity log of a group shows activities

Each of these tests are broken down into finer details for each functionality.

Maintenance of test database

The test database is restored to its pretest state by clearing all the tables that have the group_id for `NewGroup1`.

Tests for group creation and joining (v4.0)

The following end to end functionality was tested using Cypress:

Group Creation

- The correct page is displayed to the user when entering the create-group page
- Users cannot input invalid group information for group creation
- User can create a new group with chosen members invited automatically

Each of these tests are broken down into finer details for each functionality.

Join group

- The correct page is displayed to the user when entering the Join-group page
- User can search for groups
- User cannot join groups they are a member of
- User can join groups that they are not members of

Maintenance of test database

The test database is restored to its pretest state by clearing all the tables that have the group_id for `NewGroup1`.

Note

Big problems faced with setting the database to its pre-test state. The need for many cy.wait() clauses to slow cypress down such that certain calls can be made to the database was a tough task. With the advice of Basheq, I managed to get the tests for create group, join group and action log to work with minimal changes. There were also problems faced and solved to do with cypress erasing its cookies and losing the user session between ‘describe()’ blocks.

16:20 - 16:30 Tali

User Stories:

I completed the following user stories this sprint:

- Search for groups by subject #134
- Choose from recommended groups based on your activity #133
- Be able to view other member's ratings #157

Features:

I included the ‘tag’ functionality. This applies to newly created groups where you can set a predefined tag value, searching for a group with a specific tag value and automatically recommending other groups to the user based on the current tags of the users groups (recommends all groups with the same tags).

Improvements:

The user is now able to view the ratings of their peers during the rating process, by placing their rating next to their user name in the selection list of users.

The covid-screening form is now accessible through the group chat window. I found it challenging to know where to start debugging as there were many aspects combing together in the tests - the actual javascript files, the html elements needing to be labeled for the cypress testing and then the test database interfacing. This was particularly difficult when there were things which needed to be altered in the database which the test needed to perform as what I was finding to happen was that only some of the calls would update the database. Eventually, it was determined that the database calls merely required time to

Notes:

The key difficulty faced this week was interfacing the test database with the user tests required to be performed. This was due to the test database taking its time to load up which was longer than what Cypress allowed for testing resulting in either failed tests, inappropriate test database updates or both. A solution to this was to implement waiting in the Cypress tests so that the database changes could be reflected. Additionally, it was required to increase the time-out time of Cypress so that it accounted for the increased wait time.

Testing:

Rating Testing

The following tests were added for the existing group rating functionality:

1. Ratings page displays correctly
2. User's rating is correctly displayed on profile
3. Rating works as expected for a null rated user
4. Rating works as expected for a user with a previous rating
5. The submit button generates an alert when selected

Covid-Screening

Tests to ensure that the form works and displays correctly was implemented.

16:30 - 16:40 Nathan

User Stories:

I completed the following user stories this sprint:

- Display a chronological list of activities performed within a group #139

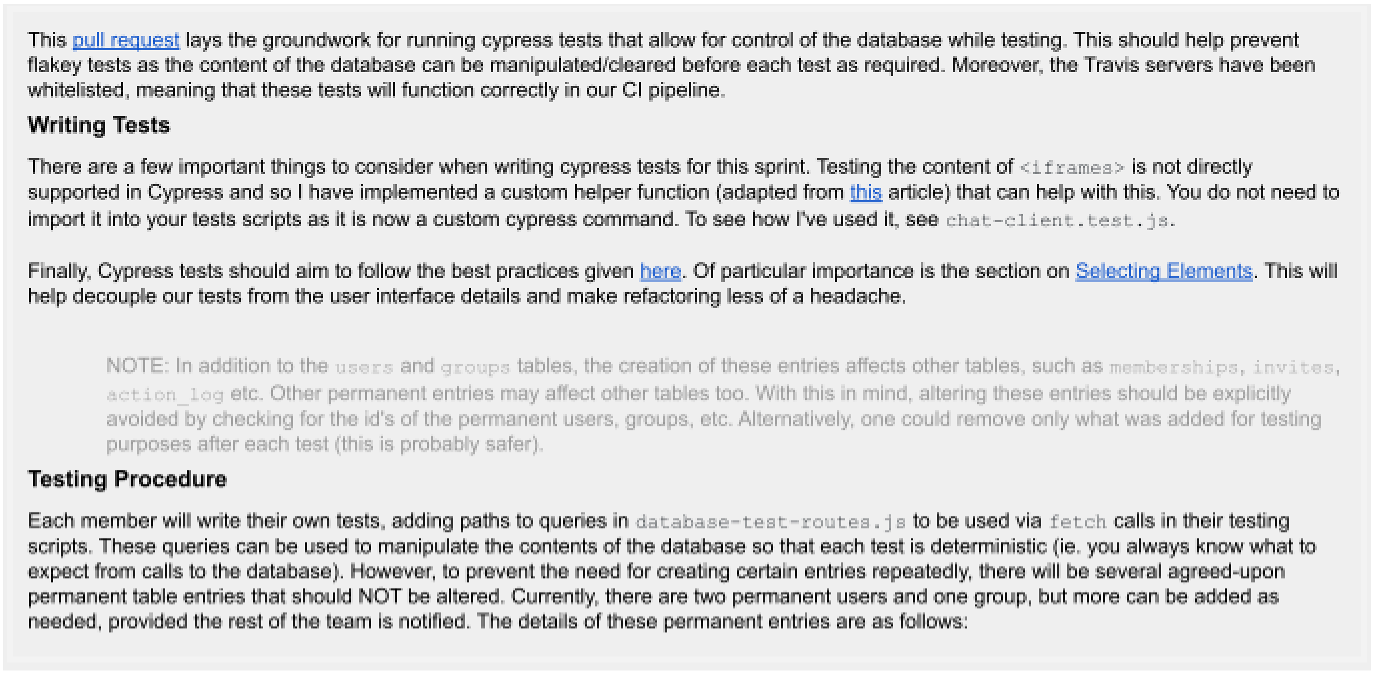
Features:

I included the ability to view the group activity log with all actions from the group. This is accessed through the 'View Log' tab on the group chat homepage. Furthermore, I included the ability to filter the options which means that the user can filter by: action, timestamp, member and description. Over and above the functionality implemented by my assigned stories, I implemented the basic architecture changes required to enable members to write tests that interact with the test database - both locally and on Travis CI.

Notes:

No major problems were encountered while implementing the user stories. A future improvement would be to be able to search within a range of timestamps, rather than just the ability to sort in ascending or descending order.

Notes to Team Regarding Testing:



This pull request lays the groundwork for running cypress tests that allow for control of the database while testing. This should help prevent flakey tests as the content of the database can be manipulated/cleared before each test as required. Moreover, the Travis servers have been whitelisted, meaning that these tests will function correctly in our CI pipeline.

Writing Tests

There are a few important things to consider when writing cypress tests for this sprint. Testing the content of `<iframes>` is not directly supported in Cypress and so I have implemented a custom helper function (adapted from this article) that can help with this. You do not need to import it into your tests scripts as it is now a custom cypress command. To see how I've used it, see `chat-client.test.js`.

Finally, Cypress tests should aim to follow the best practices given here. Of particular importance is the section on Selecting Elements. This will help decouple our tests from the user interface details and make refactoring less of a headache.

> NOTE: In addition to the `users` and `groups` tables, the creation of these entries affects other tables, such as `memberships`, `invites`, `action_log` etc. Other permanent entries may affect other tables too. With this in mind, altering these entries should be explicitly avoided by checking for the id's of the permanent users, groups, etc. Alternatively, one could remove only what was added for testing purposes after each test (this is probably safer).

Testing Procedure

Each member will write their own tests, adding paths to queries in `database-test-routes.js` to be used via `fetch` calls in their testing scripts. These queries can be used to manipulate the contents of the database so that each test is deterministic (ie. you always know what to expect from calls to the database). However, to prevent the need for creating certain entries repeatedly, there will be several agreed-upon permanent table entries that should NOT be altered. Currently, there are two permanent users and one group, but more can be added as needed, provided the rest of the team is notified. The details of these permanent entries are as follows:

Testing:

Chat Tests (v4.0)

The following tests were added for the existing group chat functionality:

1. Chat page displays correctly
2. Messages can be sent to the chat and remain persistent
3. Empty messages cannot be sent
4. Website and internet links can be sent and are automatically converted to clickable anchor elements
5. Links above 40 characters are truncated
6. Messages are grouped by date using appropriately named date dividers
7. Multiple users can interact by sending messages and viewing active chat participants

Meeting Tests (v4.0)

The following testing categories were covered for the functionality associated with adding meetings:

1. Dynamically generated form elements load correctly
2. User can successfully create an online meeting

3. User can successfully create a face-to-face meeting
4. Automatic face-to-face recommendations function correctly

16:40 - 16:50 Tarryn

User Stories:

I completed the following user stories this sprint:

- Automatically send a message on the group chat when a face-to-face meeting is scheduled #193
- Automatically send a message on the group chat when an online meeting is scheduled #135
- Automatically send a message on the group chat when a group member reaches their given safe destination #138
- Track group members for a period after the meeting #137

Features:

I included the following features:

- A notification is sent to the respective group chat in the form of a message when a meeting is created.
    - The chat message when broadcasted will have the group name as the sender. However, since the group does not have a user_id, the creator's user_id needed to be used so that the message could be saved to the database. Which means that the message is later seen as being sent by the creator of the meeting.
- I included a meeting attendance page with a non-persistent chat.
    - The user has the ability to view the location of other members in the meeting (unless those members opt to stop sharing their location - in which case only their most recent location will be sent).
    - A user may click on the name of any user which will result in the opening of another tab which will navigate towards the selected user's location.
    - A message is automatically sent into the attendance chat when a user leaves or joins the meeting.
    - A message is automatically sent when a user has either reached their destination or has opted to stop sharing their location. 'Reaching their destination' in this iteration implies that the user has been in the same place for over 2 consecutive server location requests - this equates to approximately 4minutes in real-time.
    - The user may also press a button to share their location again.

Improvements:

The following improvements were made to already existing functionality:

- Meetings can no longer be created if they have a date that is set in the past.
- The user can only view the *upcoming* meetings, this includes meetings in the past 24hours.
- The view-meetings page includes a 'static' timer with each of the meeting entries which indicates the time remaining until the meeting starts, the clock displays the time remaining from when the meetings were loaded.

Notes:

Future Improvements:

- Change the location sharing to an 'opt', so that the user location is not shared IMMEDIATELY as the user enters the chat. This also means that the user will be able to access the meeting chat without needing to share their location.
- improve the location accuracy.
- Include an active clock which indicates the time until the meeting starts.
- change the time remaining in the view meetings page to also actively adjust

Problems:

- The Google maps script does not always load before the rest of the script for the first time when entering the page.
- The user NEEDS to allow location sharing for the feature to work.
- Travis is unable to allow location sharing.

Testing:

Profile tests:

The following tests were added for the view profile functionality:

1. The profile page displays correctly
2. The user's username, first name, last name and rating is displayed.

My Groups tests:

The following tests were added for the view my groups functionality:

1. The groups table loads correctly for the given user (checks the Scotland group for Archie).
2. The group name contains the correct link to redirect the user to the group chat page.

Attend Meetings Tests:

The following tests were added for the attend meetings functionality:

1. The attend meetings page is displayed correctly
2. Messages are automatically sent into the chat when a user joins.
3. The user can send messages into the chat
4. Multiple users can join the meeting and their location is displayed on the map with their respective 'label'
5. The user's location is updated on the map and clicking on a username will open up a window with directions to that user's current address.
6. Messages are automatically sent into the chat when a user reaches their destination or stops sharing their location.

PLEASE NOTE:

Travis does not account for the pop-up that occurs when the browser requests permission to share location. This means that when the tests are first run, since the call to join the socket occurs in the loop after the location has been accessed - the socket server never receives the request to join. Therefore the member information from the chat is never received, and the user never joins. This impacts all the tests involving a user/member. These tests have been skipped for deployment purposes, as they work locally where you can allow location sharing. The video and screenshots below illustrate the working tests.

View meeting tests

The following testing categories were covered for the functionality associated with viewing meetings:

1. User can navigate between meeting types
2. User cannot view face-to-face meetings if they have not passed covid-screening

16:50 - 17:00 Scrum Master Conclusion: Yasser

Action items	Person responsible	Deadline
✔ Sprint review upload	Yasser Karam	26 June 2021
✔ Sprint planning	Team	26 June 2021

Agenda item:	Sprint Retrospective Session	**Presenter:**	All

17:05-17:10 Break

17:10-17:40 Discussion

Given the time constraints imposed due to university commitments, the team did well to achieve the functionality described.

17:40-18:00 Sprint Velocity: Yasser

In the second sprint, 18 user stories were completed, which amounted to 35 user story points. Therefore, using the formula below (see this article), the average sprint velocity after Sprint v4.0 is calculated to be:

$$Avg\ Sprint\ Velocity = \frac{story\ points}{number\ of\ sprints} = \frac{18+12+32+35}{4} = 24.25\ pts/sprint$$

Scrum Master Post Review and Release Conclusion:

This sprint saw extremely great work ethic from every member of the team. The testing threw many errors and issues but with the cooperation of everyone, the team overcame them. Some tests ended up failing exclusively because of Travis and so they are skipped in this release.

This is the last formal sprint, but there will be another release including some final artefacts and testing that does not add any new functionality.