Team Meeting:
ELEN4010 Study Group Coordinator

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:

09:50 - 10:00 Scrum Master Introduction: Tarryn

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

10:00 - 10:10 Basheq

I completed the following user stories this sprint:

- Vote to accept students that applied for group membership #85
- Vote to terminate an existing membership #86
- Vote on custom polls made by users #129

This pull request implements features related to polls and voting

- Viewing group requests and starting a poll to accept group members
- Viewing group members and starting a poll to ban a member
- Viewing all active polls and voting on polls
- Votes are viewed in real time and graphically displayed with Chart.js
- Only one vote per user
- Upon completion of a group request or banning poll, the appropriate action will be taken with regards to group members

Due to time constraints and university commitments, all tests were not conducted. However code was written to be modular so tests will be written in the next sprint. The test database is also now ready and will be used for unit tests and e2e tests in the next sprint. Testing is essential and more focus should be placed on it in the next sprint. An extra user story was completed compared to what I was assigned, since it was very similar domains.

10:10 - 10:20 Yasser

Completed the following user stories:

- Send an invite to the selected user when creating a group #54
- Apply to an existing study group to start a vote on your acceptance #81
- Add new users to an existing group #94
- Have the option to invite members during group creation #131

This sprint added functionality to the creating and joining of groups. When a user selects people to invite to the group that they are creating, invites are sent to those usernames. When a user wants to join an existing group, a group request is made to that group. These are both using the Azure database for persistence.

There is a drop down menu that shows all the users that are not in a particular group. This is for later integration in a group chats page.

10:20 - 10:30 Tali

I created the functionality of being able to take a Covid Screening, based on the physical [Wits form for Covid screening](#) , which then assesses whether the user is eligible for face-to-face meetings. In the choose-group.html file, there is a new button to complete the Covid screening. Clicking on this link redirects you to the form submission for Covid screening with the option to go be redirected to the official Wits Covid Screening page.

This new code addition implements the functionality of being able to rate a current member in the group you are currently in. Once a user is within the chat page, they are able to click on the rate members button. They are then redirected to a separate page where the current members in the group are listed. From here, the user is able to select a user within the group and provide them with a rating from 0-5 (selected as an option in a radio list). For a future release, the current user should not be able to rate themselves. To update the rating, the selected member's current rating is fetched from the database, multiplied by the number of ratings they have received and then the new ranking is added. The average rating after accounting for the increased number of ratings is then posted back to the database.

No testing was able to be implemented during the sprint. Difficulties encountered were the fact that my computer stopped being able to load the localhost which meant that my development was delayed this week as I had to get my laptop sorted out first. The problem turned out to be a Windows update that blocked the local host from running. Additional difficulties were figuring out how to capture the user name for queries in the database. Once, this was figured out, the rest of the sprint was relatively smooth.

10:30 - 10:40 Nathan

The functionality required for interactive locations has been implemented and showcased. Users can type in an address or set of coordinates in a form field. This input dynamically updates a map using the [Google Maps Embed API](#). The user can use this map to confirm that the correct address has been found. Once satisfied, they may submit the form. For demonstration purposes, this adds a [Google Maps link](#) to an adjacent list that when clicked will open a new window displaying the directions to the chosen location. It is assumed that these two features will be integrated with the creation and display of meeting events.

Users can now send clickable links in the chat that can be accessed by other members. Users simply paste the link into the chatbox and send - the link will be automatically converted to an anchor link using the [Anchorme.js](#) package. This works for a range of links including HTTP, domain names and emails. In addition to this, messages are now grouped by the date sent, making it easier for the user to distinguish adjacent messages sent at similar times on different days.

10:40 - 10:50 Tarryn

Completed the following user stories:
- [Add face-to-face meeting events to the group calender with date and time](#)　　#89
- [Add online meeting events to the group calendar with date and time](#)　　#87
- [Add a relevant link to the meeting](#)　　#83

Included Create and View meeting functionality, the meetings displayed are specific to the group in which it was accessed.

In creating a meeting:
- The user can choose to create an online or face-to-face meeting
- The link creation and interactive location selection from "Interactive Group Creation and Viewing v3.0" is integrated with the creation of face-to-face meeting events.

In viewing a meeting:
- The user can choose to display the online or face-to-face meetings
- The link creation from "Interactive Group Creation and Viewing v3.0" is integrated with the displaying of face-to-face meeting events.

2

- Integrated with Tali's covid screening so that in order to view the face-to-face meetings the user must have passed the covid-screening procedure.

Updated the following from the previous sprint:
- The view user profile and view memberships functionality (previously accessible through the 'memberships' page on the temporary dashboard) has been imported into the dashboard page (choose-group), so that the user may click a button to load their details and memberships to the same page.
- Links have been added to the memberships table which allows the user to navigate to the respective chat page upon selecting the group name.

Future Improvements:
- include a time-line for the displaying of face to face locations (ie can only see the location if the user has passed a covid screening test within a certain timeframe relating to the start of the meeting)
- include the creator's name/username instead of their id

Possible Future improvements:
- Query related:
  - change the query to filter the results based on whether 'online' ?
  - add/alter query to retrieve face to face meetings only if the user has passed covid screening

10:50 - 11:00 Scrum Master Conclusion: Tarryn

All user stories were completed, with two extra stories included ("Vote on custom polls made by members" and "Have the option to invite members during group creation"). The user story: "Be rated based on the activities on the app" was moved to another epic with a different description that better describes what we wanted to achieve: "Give a member an overall rating out of 5".

Testing will be further examined using jest and cypress in the next sprint.

Action items	Person responsible	Deadline
✔ Sprint review upload	Tarryn Maggs	18 June 2021
✔ Sprint planning	Team	18 June 2021

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

11:05-11:10 Break

11:10-11:40 Discussion

Further testing with the integration will be required for the next sprint. Given the time constraints imposed due to university commitments, the team did well to achieve the functionality described.

11:40-12:00 Sprint Velocity: Tarryn

In the second sprint, 15 user stories were completed, which amounted to 32 user story points. Therefore, using the formula below (see this article), the average sprint velocity after Sprint v3.0 is calculated to be:

$$Avg\ Sprint\ Velocity\ =\ \frac{story\ points}{number\ of\ sprints} = \frac{18+12+32}{3} = 20.67\ pts/sprint$$

Scrum Master Post Review and Release Conclusion: Tarryn

The Sprint itself was concluded prior to this review, however the release was not made until integration of the different aspects of the program had occurred. In order to continue with testing in the next sprint, the integration was imperative as well as setting up the test database. This is so that the testing can continue more seamlessly in the next sprint.

As a result of the extra work involved in the integration as well as the workload imposed on the team members due to university commitments, the release was delayed by less than 24hours of the previous release date.

As this sprint's scrum master, I would like to commend the team for their outstanding dedication, commitment and exemplary work. Furthermore Nathan and Basheq went out of their way to make sure that each page followed the same theme so that a 'flow' for the application was achieved.