Team Meeting: ELEN4010 Study Group Coordinator

11 June 2021

Time: 9:50 -12:00

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:

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

Brief overview of the scope covered by the sprint as well as the user stories implemented. Early in the sprint, the team set up the Azure SQL database which was to be integrated in this sprint.

10:00 - 10:10 Basheq

Building on the previous sprint's functionality in integrating the database and adding more features.

This pull request implements two features. Firstly, the creation of accounts as well as logging in is now persistent via the Azure SQL database. The logged in user can view invites sent to them, and accept or reject the invites. Accepting an invite results in the user being added to the group on the database as well.

These two features cover three user stories:

- User details are saved when I log out
- View invites that have been sent to you
- Accept/Reject invites to be part of a study group

At the moment, tests fail due to the database interaction with Travis, which will need to be fixed next sprint.

10:10 - 10:20 Tali

I implemented the functionality to display all the groups from the database and then subsequently enable the user to search for a specific group and select the chosen group. At this stage the selection does nothing however, following integrating other user stories, selecting the group will give you the option of joining the group.

Additionally, through using Tarryn's base functionality for membership joining and display (implemented in Sprint 1) and her integration of the user membership functionality in the current sprint, I was able to build the feature of leaving a group from a user profile by deleting the user membership when a button is pressed. Due to time restrictions, no tests of functionality (with Jest or Cypress) were able to be implemented during this sprint since a testing database would need to be implemented prior to proper testing.

These functionalities completed the user story associated with the **"Search for a Study Group"** covering the product backlog items for search for all existing groups by name and the user story associated with **"Viewing your Existing Memberships"** covering the user story to leave a specific group.

Difficulties encountered during the sprint included my computer no longer been able to connect to the server which left me unable to correct my code on my laptop. This issue was solved by using Microsoft Teams to run my code off of Tarryn's laptop by taking control of her screen. Another difficulty encountered was that of spending a long time debugging the feature for the dropdown list population as it was not displaying correctly, only to realise 2 hours in that it was in fact displaying correctly but due to the text being in black on a black display background - it just could not be viewed.

10:20 - 10:30 Tarryn

Completed the following user stories:

- My Groups are saved when I log out
- View user details

I created a basic user profile which we can use to navigate between the user's current memberships. This iteration interacts with the azure database of our application. The user specific details and memberships that are displayed are linked to a particular user-id. Since integration of the user login feature was implemented after my pull request, I will potentially need to change the accessing variable name from 'id' to 'user_id'.

When the profile page loads, the user's profile details can be seen. The user details are loaded with the page loading. The process is as described before.

The membership button is pressed to load the user specific memberships into the table. This is done by accessing the memberships table with the specific user id and then finding all the corresponding entries, an inner join is then placed on the group_id of the memberships table and the group_id of the groups table. The table will include a row for each of the memberships. Each row will contain the group's ID and name, as well as the date the user joined the group, the number of users(members) that belong to the group and the number of members that are currently online (in the group's chat).

The navigation to the groups has not been implemented, and the database currently does not have columns for the following: the number of users(members) that belong to the group and the number of members that are currently online (in the group's chat). The leave group button functionality was implemented within Taliya's code.

Future improvements and considerations: Increase the speed at which the profile page loads as it is currently rather slow. Finish the change in integration with the user login from Basheq's code.

10:30 - 10:40 Nathan

Building on the functionality of the last sprint, I implemented the required functionality of making the group chats persistent across different user sessions. Now, messages sent in the group chat are added to the server's database as well as being sent to the other users in the chat. Whenever a user enters the group chat, the previous messages sent to the database are retrieved and displayed in the chat area, which can then be viewed by the user.

The primary difficulty was with testing this added functionality. When we were setting up the database, we did not account for the fact that our tests would all pollute and interact with the same set of data, resulting in flakey tests. Moreover, Travis CI does not have access to the database meaning that an error is thrown when certain pages attempt to retrieve data. As a temporary workaround, these tests were skipped until a solution can be found in the next sprint

This functionality covered the following user stories

- My chats are saved when I log out #53
- My chats are linked to my user account #56

However, it should be noted that currently any valid username can be chosen. This needs to be integrated with the user login functionality of a future sprint. Future improvements could include the following:

- Displaying a visual loading indicator while the chat history is being received so the user can see what is occurring
- Separating messages by day in the chat so the history can be tracked more easily
- Having different coloured messages associated with each user to better identify the flow of a conversation.

10:40 - 10:50 Yasser

This sprint saw the integration of the last sprint with the Azure SQL database. The user can now search for usernames to invite to the group that they would like to create. The usernames drop down list is updated by the ‘users’ table in the database.

The table showing the groups that exist and the added groups, has been adjusted to show the attributes from the ‘groups’ table. The ‘members’ column is not yet linked to the ‘memberships’ table in the database. The groups that the user creates, enter a new record in the ‘groups’ table, and the page then refreshes the table view of all existing groups to show the created group.

Unfortunately, due to time constraints, the invitations to members selected from the dropdown menu does not update the ‘invites’ table yet. The implementation of this is linked to the update of the ‘members’ column on the web page, and so the functionality related to joining groups - invitations on creation, and manually joining are not implemented yet, but can be done in the next sprint mimicking the integration with the database already completed.

The integration of existing functionality with the database took longer than expected, and produced more developer stories than anticipated. This should be considered in the next sprint.

User stories completed: #52 (#54 for next sprint)

Developer stories: #70, #71, and #78

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

Most user stories were completed, aside from “Send an invite to the selected user when creating a group”.

This was due to the workload and other commitments to university work. The testing will be improved upon in the next week. A testing database will be created for the purpose of testing using Jest and Cypress.

The release was made a week after the previous release, and within an hour of the previous week’s release time. Although the whole team was involved in the release, I (as the scrum master) approved and walked through it.

Action items	**Person responsible**	**Deadline**
✔ Sprint review upload	Basheq Tarifi	11 June 2021
✔ Sprint planning	Team	11 June 2021

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

11:05-11:10 Break

11:10-11:40 Discussion

Further integration will be required for the next sprint. Given the constraints of university, the team did well to achieve the functionality described.

11:40-12:00 Sprint Velocity: Basheq

In the second sprint, 10 user stories were completed, which amounted to 12 user story points. Therefore, using the formula below (see this article), the average sprint velocity after Sprint v2.0 is calculated to be:

$$Avg\ Sprint\ Velocity\ =\ \frac{story\ points}{number\ of\ sprints} = \frac{18+12}{2} = 15\ pts/sprint$$

As discussed in the retrospective, this was due to commitments to other university work. However, most user stories were implemented.