Software review session nr 3:

Date: 02/06/2023

Meeting Start time: 08:40am

Meeting purpose: review sprint 3

Members joined: Reuben Ellis, Meg Hammond, Mannan Johari, Maria Daras

Meeting Platform: WhatsApp Audio call

Agenda:

08:42- discuss and reflect what we did in sprint 3. The goal for the week was to complete the remaining functionality. Also, to include testing. Deploy the website and to setup the remote MongoDB database.

08:44 - Tasks completed this week from the scrum board: Decided to rather have a table with scheduled bookings which can be seen by the lecturers and the students. The lecturers are able to cancel a booking, set their available times and see only bookings that affect them. Students are able to see all bookings for specific lecturers and are able to cancel bookings that were made by them. They are also able to schedule a new booking or join a booking already scheduled by another student.

08:47 – Maria says the issue for her this week was the tests. A way this could have been avoided is by starting sooner on the tests and having more people working on that part. She struggled with the understanding of the mock databases and how to connect the jest library to the MongoDB database. She spent a lot of time researching. The time constraints were quite harsh. Because of this she was only able to make unit tests for the user file which is used for the login and registering part of the web app. As well as the error handling of invalid registrations and logins. She would have liked to do more testing, but the time was an issue. She spent around 16 hours working on the tests this week.

08:51 – Maria says that we actually handled the week quite well, we achieved a lot, and we used our time well. We understood how much time we should put into this week to accomplish our result.

08:53 – Mannan: He completed the remaining of the user interface and it went quite smoothly. He spent about 3 hours doing that. Deploying the azure website resulted in many errors as we had missing info in the .json file. Some paths had to be changed, The app.js file also had to be updated as well as the port speed had a syntax issue. The azure deployment took the whole day.

08:56 – Meg says she implemented the consultation booking functionality allowing you to book a consultation. This included creating a booking page and all the logic for validating a consultation. She also added the maximum number of consultations per day for a lecturer. This needed further refinement of the consultation booking logic. For the consultation database she had to add the attendees and titles to the database. She also created the database that holds the action log. She also implemented it to catch all the actions/activities such as logging in registering, lecturer updating their availability, student making a successful booking and a student unsuccessfully making a booking. Meg also added the functionality for a student to join a consultation. She also made some fixes which were overlooked during the development of the stories, as well as added the lecturer to be able to see other attendees on the lecturer dashboard. She spent a total of 23 hours coding this week, including some research and understanding datatypes in Javascript and how datatypes are reflected in a MongoDB database. She also needed to understand how to access and modify data in a database.

09:08 – As far as error handling goes there wasn't a need to implement a lot of error catching on Megs part. She specifically set up the html pages so that the input types specify the datatypes. So for example if you are to specify the email address then it automatically checks that an @ sign is present. Likewise with the numbers it will only allow you to enter a number and it is automatically range checked so it is within a certain range. This is why there is no error handling in the lecturer availability page. The input types within the html page perform the error handling and error checking needed. It required quite a lot of research to realise this and how it works.

09:13 – I, Reuben, onboarded the MongoDB database onto the atlas server which is a remote server. This allows us all to use one database instead of all having separate local databases. This database then required changing the app.js to connect to this server using a password and user ID. I also did the part so that students can see the available times of the lecturer, a lecturer can see all the consultations for themselves. The lecturer is able to cancel a consultation. Students can see all consultations for a specific lecturer and are able to cancel bookings made by them. I implemented the action log page which shows all the activities on the home page.

09:20 – Realistically if this web app is implemented it would be a safety concern for people to have access to the action log. This should only be accessible to some admin staff. This admin staff should be some or other IT professional who knows how to handle this information. Even having this action log without permission from the user will have to be asked for. This will be something like using cookies. If this information is leaked it could be sold and so on. This may be very problematic. The action log is purely there for demonstration purposes and that actions can be viewed. If this was a viable product it would need to be implemented differently as discussed before.

09:25 – Some issues and reflection of our project. The student can access the lecturer dashboard by using the URL to the dashboard after they have signed in as a student. This is an issue with the ensure Authenticated part of the code. This causes some issues with our web security. A further improvement that could be made on Meg's part is that when a lecturer sets the maximum number of consultations they will have in a day. She chose 50 rather arbitrarily. She should have added some additional checking to make sure that the maximum number of consultations can fit into the longest day. A maximum number of consultations should be set to each day of the week rather than having the dame amount each day. It would also make sense for lecturers to be able to set multiple available times within the same day. Potentially we could also have added a reset password functionality. This would be a really good feature to implement, and is also very necessary for actual usage of the web app. Another feature to add could have been back buttons to the login pages as this will help usability. When registering they should enter both their name and surname to be displayed. Following on form the name situation it would also be beneficial for when the lecturer sees who is attending the lecturer can see the students by name and not by email. Sometimes when a lot of people are connected to the database or if the internet connection is rather slow, the connection to the database times out. There is no actual page that shows when there is a server error and this should be added. It will be ideal to rather host our website on something other than azure or a paid version of azure. A security certificate should be obtained and the error of the "dangerous site ahead" should be sorted out with google. The website can also maybe a have a list of instructions to reduce confusion. It would be more beneficial to have error messages come up as pop ups or alerts, otherwise you have to look around to actually find the error messages. Another option is to add the error messages in bright red font.

09:45 - The project review:

The website is basically done, backend we could have had more unit tests. And should have implemented the tests after we make a new implementation. Communication was a bit poor throughout the project and teamwork should have been better. The group should have come together a bit more to talk about changes and ideas. From Maria and Mannan's side they feel that they were left out a bit. Tasks could have been delegated a bit differently. Between when the brief was handed out and when we should have started our first sprint to have a 4 week sprint, there was almost no time to plan out user stories. Creating the user stories was the first thing we should have done. This would have allowed us to better split the tasks. Developer sized stories should be very thin slices which also includes some frontend and backend. But most of the stories were actually large backend stories. There were also some user stories we didn't mention, like a lecturer should be able to log in. But these were added in documentation. There were also many timetable clashes which didn't allow us to meet as often s we actually needed. The this is this is not a job for us, and this project actually requires professional development plans, we have 5 other subjects that require just as much time making it very hard for some of these meetings to be attended or for us to actually keep up with the sprint velocity required.

Meeting ended – 10:02