Week6

The following lists our questions and TA's responses

For the database question, we must import the json file from the activity bank to the database and be able to query them. It doesn't matter whether we do the import only once or we import it every time the server starts, but it should be down with some functional code on the server-side(e.g. some endpoints to parse the json), it is not OK to only use GUI to do the import.

For the initialization question, Registering scene as parameters is fine if it works properly, however, if the start method gets bigger and bigger and becomes too large, we'd better break that into parts (I don't quite understand Duuk's advice here, but I think the basic idea is doing some encapsulation instead of using the constructor),

For the testing question, we don't be expected to test the controllers, but there are still lots of things in the backend which needs to be tested (common classes/endpoints etc)

For the advice of adding service, it is nice to have services, because it is kind of the backbone of the backend, generally, clients ask for information through the endpoint, the endpoint then uses services to handle the request and send data to the client. (I remember later we discuss something such as adding services in the common folder so that the frontend can also use it. It is beyond my understanding so I just didn't record it, please tell me if this part needs to be improved)

The following lists Duuk's suggestions to keep us on track

1. We should improve our issues, each issue should have a detailed definition of what to do (issue description), and it and be listed as subtasks with checkboxes(check it when it was done). Issues should be independent of each other, and the workload should be set properly so that one person can handle that in one week. Every issue should be assigned to somebody, and the person that takes the issue makes a separate branch of it and merge it to main when it is done.
2. We have to pick some issues as the current sprint for every week
3. We should improve our Javadoc and test for the project.
4. If the branch in the merge request has some problem, we should fix everything within a single merge request rather than create a new branch for that or actually fix it in another merge request.
5. We can plan ahead what is needed to do then start implementing, it's better to divide the tasks as frontend/backend distribution so that we can work in parallel, and we switch by week so that everyone is familiar with both sides.

For further possible technique questions, we can first reach TA, then go mattermost/stackoverflow for solutions.