Team Reflection Course Week 5 (V19)

	A	B	A → B
Scope & Stakehol-der	-We have different ways of displaying created activities. -We have a login function and almost fixed the logout function. -Users can navigate through the basic structure of the app but some of it still remains to be done. -Users can create activities and view a basic description of it.	Develop a mobile Android application where people can plan and create leisure time activities and also join in on other people's activities.	Complete the tasks in our scrum-board and perform testing of previous tasks.
Priority of features	As of right now we are implementing the Facebook and Google login features so that the user can have alternative ways of logging in on our app. We are also working on the two interfaces for 'finding activities' which will enable the user to alternate between two interfaces and search for an activity.	Our aim for next week is to have functioning interfaces for finding activities and also functioning interfaces for profile and messages. The reason behind this is because our stakeholder told us that these are some of the essential parts of the app and therefore our focus has been on that.	Because the interfaces we are working on right now are quite important and somewhat complicated to do, we have sessions where we code together by screen sharing and meetings over teams.
Success criteria	This week we want to achieve a basic functionality of the interfaces for finding activities and also interfaces for profile.	We want to better use the team meetings for group programing because programming together is better for developing code faster.	How we can get here is by working on tasks together in the same branches on Github and also having meetings where we code together by screen sharing.
User stories	The same as last week. Most of the user stories have been written. If we get some new ideas from in-house or from our stakeholder, we add this as a new user story.	Since it's just a few sprints left, we want to further prioritize the user stories according to the value they bring to the application. They should of course also be able to be completed in time.	Each week, we get better to estimate the time and effort of each task. To be able to prioritize the tasks, we want to evaluate how important each user story is for the application. To further understand this, we get help from our stakeholder.

Acceptance tests	We have booked new meetings with our stakeholder to be able to complete our acceptance tests.	Our aim is the same as last week, to have recurring meetings with our stakeholder. This is made because we want to make sure that the work we do provides value for our future customers.	It is important to have continuous support and contact with our stakeholder to reach our aim.
KPI	Like last week, we are evaluating the KPI:s each week. This week we have looked at the KPI:s for all of the past sprints and concluded that they've been fluctuating a little bit.	We want the KPI:s to improve each week.	To achieve this, we're trying to, for example, give points to tasks more correctly and evaluate for past tasks if they needed higher/lower scores. We've also put aside some less important user stories that's not essential for the application to work. The aim of this is that we feel less stressed and can put more effort into making better implementation of the other tasks.

Social Contract and Effort

	A	B	A → B
The rules that define how you work together as a team, how it influenced your work, and how it evolved during the project	The rules haven't changed since week 3. We keep following the written social contract and continuously try to improve it every week. The social contract has been helpful for our collaboration, since everyone has their own role and responsibility. We can see how the social contract has a positive influence and how it evolves by looking at how the team members have become more comfortable with their role and the rules	We want to improve our social contract as much as we can and also see personal improvements from everyone within their roles.	By having certain rules and roles during the project, we believe that the group will be able to work more efficiently. As mentioned last week, we will handle conflicts in early stages and adapt our rules and roles to minimize any future conflicts. The social contract will be molded into one that truly works for us and reflects our group dynamic. We will continuously remind each other about the rules in order to retain the responsibilities that we have. Hopefully, we'll see some improvements we want during our work with the project.
The time you have spent on the course and how it relates to what you delivered	We have individual as well as cooperative coding by sharing screens. We also have regular meetings each week. We are each trying to gather at least 10p in each sprint. Currently we are not reaching our 70p weekly goal.	As a group we have to spend more time actually coding and maybe defining tasks even better in order for us to be more effective in completing tasks and reaching our weekly goals.	We are starting to get more data from the KPIs and it gets easier to evaluate our velocity. We are also monitoring the individual workload for each group member. We will continue to monitor our progress and maybe start limiting extra features to be able to deliver a product in the end without killing ourselves in the process.

Design decisions and product structure

	A	B	A → B
How your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value	This week we have improved the security of the Firestore by implementing security rules that prevent nefarious users from getting access to private user information.	Make design decisions that support customer value.	Continue to improve the security of the Firestore database to provide the expected security for users. Furthermore we should continue to choose api:s that support customer value.
Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)	<u>Same as last week:</u> - We use Google Drive for e.g. KPI, team reflections, and other text documents that everyone in the group needs access to. - Diagrams.net for UML. - Miro for our mockup. - Github. - Spreadsheets. - Trello for scrum-board.	We want to begin using testing documentation, case documentation.	By reading online how to conduct these and become experts on the field so that we are confident before using them.
How you use and update your documentation throughout the sprints	The structure of the scrum board has improved since last week but it's still a bit messy since a lot of stories are in the testing stage.	Have a well structured and easy to follow scrum board.	Make sure to continue following the rules of the scrum board and perform the needed testing to remove the excess cards from the testing column.
How you ensure code quality and enforce coding standards	<u>Same as last week:</u> By having established rules that everyone's follow (such as MVVM, delegation of functions and other). Currently only some of the classes and methods have the appropriate documentation.	To follow conventional code practices and document the code using comments and JavaDoc.	We can get here by everyone following the conventional rules and if someone does not follow the rules to mention that to them so that they are illuminated for next time. Furthermore we will have to focus more on the documentation of the code.

Application of Scrum

	A	B	A → B
Roles	We have tried to distribute the same amount of workload for each role.	We want each role to have the same amount of workload	During sprint reviews review the KPI and to bring up if any role caused any reason for the grade in the KPI.
Agile practices	We have daily check-ups in which everyone can participate in discussion.	We want our discussions to be more clear for everyone so that everyone can participate in them.	We want to start using screen share more often when discussing programming issues.
Sprint reviews	Only a few team members solve merge conflicts when merging branches in GitHub so we can spend more time during sprint reviews for discussion of the sprint.	We want to distribute the workload of reviewing the code so everyone can participate in solving conflicts when merging branches.	Everyone is responsible for merging their own branch and developing a system so at least one other team member also reviews your code.
Best practices for learning and using new tools and technologies	Only a few team members know how to solve merge conflicts when merging to GitHub.	Everyone should know how to solve merge conflicts.	Learning by doing. For the coming sprint review each team member is responsible for solving their own merge conflicts from their branch.
Relation to course material	We have not read anything about solving merge conflicts.	We want to read and share articles and videos regarding solving merge conflicts	We want everyone to have established knowledge about solving merge conflicts.