

Team Reflection Course Week 6 (V20)

	A	B	A → B
Scope & Stakeholder	This week we are changing focus from functionality to design, by the recommendations of our stakeholder. The input we have received is that layout makes the navigation easier and the functionality more customer convenience. So this week we want to create customer value in the form of a nice looking application that is customer friendly.	Improve the design and the layouts so that navigation and functionality are equally customer friendly in order to create more customer value.	<u>Same as last week:</u> Complete the tasks in our scrum-board and perform testing of previous tasks.
Priority of features	As of now we are improving the design according to our stakeholders wishes. We are also connecting different parts of the application and tracking down bugs. We have slowed down implementing new features in order for us to have a working product in the end. We do not have 100 % fully functioning interfaces for our profile and message view as we had hoped.	Our aim for next week is to have a new design for the application so that parts of it can begin to be implemented next sprint. We also want to have all the basic interfaces/ui connected so that a user can fully navigate the app and it's functions.	We are going to try harder to get more tasks done and with a steady flow during the week instead of a huge dump each sunday.
Success criteria	This week we want to design the app and the layout so that customers find it appealing and make it easier to navigate.	We want to become better at the coding of design and have more convenient documentation of the design so that it's easier and faster to produce the app with functionality and sleek design.	<u>Same as last week:</u> 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. Also to have more meetings with the stakeholder so that we can 'test' our design and get input on how convenient and useful the app is with the

			improved design.
User stories	Like 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.	The same argumentation as last week: Since it's just a couple of sprints left, we want to really 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.	Same argumentation as last week: It gets increasingly more important each week to prioritize the tasks, since it now is only a couple of sprints left. 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	New meetings with our stakeholder have been held and we have been given valuable feedback since last sprint. This has been mostly design-wise and also that a user should be able to use basic functions of the app without logging in.	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.	Like last week: It is important to have continuous support and contact with our stakeholder to reach our aim.
KPI	The KPI:s have improved by a substantial amount since last week.	The project burndown can still improve for us to be able to reach our goal	We've put aside some less important user stories that's not essential for the application to work. We may also have to reevaluate our Epics since they may not include as many points as we first thought.

Social Contract and Effort

	A	B	$A \rightarrow B$
<i>The rules that define how you work together as a team, how it influenced your work, and how it evolved during the project</i>	The rules haven't changed since last week, we are continuously following the rules written in our social contract, which have been helpful for our collaboration.	We want to improve our social contract as much as we can and also see personal improvements from everyone within their roles.	We can get to B by working with the rules as we mentioned last week. By following the rules, and reminding each other about what's written in the social contract, we believe that there'll be improvements in our work during the project.
<i>The time you have spent on the course and how it relates to what you delivered</i>	This week the majority of the group have had a lot to do in the other courses, which have affected the amount of time that we as a group have spent on this week's sprint.	We want to spend more time coding and especially in estimating our tasks, in order to work as effectively as possible and reach our weekly goals.	<p>Since some of the group members have had a lot to do this week, we are all aware that it will somehow affect this week's KPI. But next week we'll all continue putting time and effort into the work as before.</p> <p>As mentioned last week, we get more data from the KPIs, which makes it easier to evaluate our velocity. By continuously monitoring our progress we'll see what improvements we can make. Hopefully, these improvements will result in a better workflow and a delivered product of high value.</p>

Design decisions and product structure

	A	B	$A \rightarrow B$
--	---	---	-------------------

<p><i>How your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value</i></p>	<p>This week we have begun with designing the layout of the app and some other features have been designed. The architecture of the app has also been changed to suit the design and make the layout more customer friendly.</p>	<p>Our ambition is to design a customer friendly application and layout so that we can create more customer value. For that we want the design to be customer friendly.</p>	<p>How to get here is by delegation of work on the designing and by having more meetings with our stakeholders so that we get the customer experience perspective on the use of our app. Also this is going to help us see how well the design of the app is appreciated by the stakeholders. By studying how other applications have their layout and what makes their layout successful is going to be a key activity for us in the coming weeks.</p>
<p><i>Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)</i></p>	<p><u>Same as last week:</u></p> <ul style="list-style-type: none"> - 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. <p>This week we added:</p> <ul style="list-style-type: none"> -The tasks have pdf:s of the layout so that everyone can see what the others have done so far and what the layout looks like. This helps us with documenting the design work. 	<p>We would like to find a web service or program to help us with the design of the application. We are looking for something like Miro board but for the design, with pdf:s and pictures.</p>	<p>Simplest way would be by testing the existing documentation platforms that we are using and maybe find them useful, but we are still looking for new platforms that can contribute to the team's deliveries and lean designing.</p>
<p><i>How you use and update your documentation throughout the sprints</i></p>	<p>The scrum board is somewhat better this week, but there are still a lot of user stories and tasks all over the board.</p>	<p>With the application becoming somewhat useful, there are a lot of tasks everywhere on the scrum board. But</p>	<p><u>Same as last week:</u></p> <p>Make sure to continue following the rules of the scrum board and perform the needed testing to remove the excess cards</p>

		this is normal, because some tasks are completely done, some are not yet done and some are new ones. We have good sense of the scrum board but we want it to be more sleek and to have less cards so that we can more easily estimate how much work we have until the app is completely finished.	from the testing column.
<i>How you ensure code quality and enforce coding standards</i>	Like last week: 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.	Like last week: To follow conventional code practices and document the code using comments and JavaDoc.	We should put more time and effort into documenting our code. Furthermore, the established rules should be followed. If we're unsure if they are properly followed, we can ask our teammates.

Application of Scrum

	A	B	A → B
<i>Roles</i>	<u>Same as last week:</u> We have tried to distribute the same amount of workload for each role.	<u>Same as last week:</u> We want each role to have the same amount of workload	<u>Same as last week:</u> During sprint reviews review the KPI and to bring up if any role caused any reason for the grade in the KPI. Reviewing during daily check-ups with role related activities.
<i>Agile practices</i>	We have few work sessions where everyone participates in a call to program	We want more work sessions	We try to make a conference call every time someone wants to work with the project free for every other team

			member to join.
<i>Sprint reviews</i>	We have done our first individual merge conflicts and reviewed one other person's code.	We want to be more comfortable with solving merge conflicts and reviewing other people's code.	We have made a new random selection of which code each team member will review so we can understand merge conflicts better.
<i>Best practices for learning and using new tools and technologies</i>	<u>Same as last week</u> Only a few team members know how to solve merge conflicts when merging to GitHub.	<u>Same as last week</u> Everyone should know how to solve merge conflicts.	We have made a new random selection of which code each team member will review so we can understand merge conflicts better.
<i>Relation to course material</i>	We are getting better in fixing the merge conflicts.	We still have to get more experienced when it comes to how to handle merge conflicts.	Everyone should be able to handle the complexity behind merge conflicts.