Team reflection week 20, sprint 6

KPIs, used to measure the effectiveness of the system (user app and admin page) are the following three:

- **Number of clicks**
  The number of clicks for a user to book a space in an office. The most important process to reduce has been the process to reserve a space in the users' home office. It has been optimized by saving the previous choice in terms of company and office in order to reduce numerous clicks. Furthermore, each component in the booking process has been overlooked to make sure unnecessary clicks are removed.
- **Time to booked**
  The time it takes for a user to book a space in an office is closely related to the first KPI, *Number of clicks*, described above. Although, time to book makes sure the user interface is easy to understand not only at a first glance when users are new to the app but also when the app is used regularly. Time has been measured regularly for the process to book a space, and it has been optimized by reducing the number of pages, without compromising on the understanding of the system in a user perspective. To fulfill the requirement of testing the app on users new to the system, it has been presented and tested by the client at the end of every sprint.
- **Number of queries to firebase**
  The number of queries the app and admin page have to make to the database when being used. Making sure the number of queries is low is fairly important in order to make sure the app runs smoothly. Further, if we were to launch this app commercially, keeping costs down by ensuring the number of database queries is low would definitely be of importance. The way Firebase is structured, it is very easy to make a lot of database queries. Therefore, the team felt it was imperative to design the database and the custom database API in a way that would minimize the number of queries to said database.

Design decisions and product structure

We have used a written definition of done which has included statements about code quality. More significant commits to our git repositories have been reviewed by someone different than the author, before being merged to the main branch. In actuality, we have some issues with poor null safety, repeating code, and poor structure.

In the future we would want a more precise code standard, and for it to be followed.

We are in a better position to write a proper code standard now that we have some experience with the languages and frameworks. In this project, we wrote the code standard in too general terms and should explore the language and its best practices before formulating our standards in the next project. The team could insist more firmly that all members should follow the code standards in the next project, and team members would probably follow the standards more willingly if the standards were better.

Application of Scrum

We have used scrum.org as a reference for how to use scrum, but have not spent that much time reading up on scrum.

There have been no guest lectures.

Lectures with Håkan and Jan-Phillip with associated slides have also been used. At the beginning of the course, we had a couple of lectures regarding scrum and the way of working scrum. Key things the “cake” principle of layers on layers that its possible to slice through. User stories and how to write them. Working with user stories relating to INVEST as listed underneath. Negotiation has been vital for our group and something that we have worked a lot with, especially in the last sprints when we used planning poker to discuss the user stories but also furthermore estimated together how much time each task was worth. We also made use of SMART criteria for the tasks. (See INVEST and SMART definition according to the lectures that we had)

I	Independent	User stories should not overlap and they should be formulated so they can be implemented in any order.
N	Negotiable	A user story should be an invitation for a conversation. It can be changed, augmented, and redacted; of course, always in dialog with the Product Owner!
V	Valuable	Each user story should deliver value, either to the Product Owner or to the Scrum Team.
E	Estimable	It must be possible to assign effort to each user story. A story that can not be estimated is not complete!
S	Small	A user story must be a manageable task. If its completion takes longer than 3 or 4 days, it must be broken down!
T	Testable	There must be clear, testable criteria to define when the story is done in the eyes of the Product Owner

The most important resource for our use of scrum during the course has been the feedback we got from out TA on the weekly tutoring sessions. The questions we got there have helped us make adjustments to our use of scrum.

In a future project, guest lectures could be nice and it could be nice to move the TA sessions in the beginning of the course so that they don’t land on national holidays.