

PROCESS REPORT

PRO202 – Agile Project 2024 Spring

Table of Contents

1. Introduction	4
1.1 The client	4
1.2 The Project.....	4
1.3 Problem Statement	4
1.4 Objective	5
1.5 Target Audience.....	5
2. The Team.....	5
2.1 Team members.....	5
2.2 Roles and Responsibilities.....	6
2.3 Radartest	7
2.4 Risk Plan	7
2.4.1 Sickness	8
2.4.2 Different knowledge about different programs	8
2.4.3 Privacy.....	8
2.4.4 Technical Issues.....	8
2.4.5 Unsatisfying solution	8
2.4.6 Quick fixes	8
2.4.7 No control on project status.....	9
2.4.8 No/little motivation	9
2.4.9 Ambitious solution	9
2.5 Contract	9
3. Development Process.....	10
3.1 Sprint planning and Execution.....	10
3.1.1 Sprint 1(Design sprint)	10
3.1.2 Sprint 2	14
3.1.3 Sprint 3	22
3.1.4 Sprint 4	23
4. Methodology	25
4.1 Scrum	25
4.1.1 Use of Scrum	25
4.1.2 Product backlog	25
4.1.3 Sprint planning.....	26
4.1.4 Task tracking and progress monitoring.....	26
4.1.5 Burndown chart	26
4.1.6 Daily meetings	26
4.1.7 Collaboration and communication.....	26
4.1.8 Retrospective and continuous improvements	27
4.2 Repository Management.....	27

4.2.1 GitHub Repository.....	27
4.2.2 Cloning.....	27
4.2.3 Branching strategy.....	28
4.2.4 Regular Commits	28
4.2.5 Pull Requests	28
4.2.6 Collaboration and code Review.....	28
4.2.7 Pair programming	29
4.2.8 Handling conflicts and challenges.....	29
4.2.9 Regular Commits	29
4.2.10 Pull Requests	29
4.2.11 Collaboration and code Review	29
4.2.12 Pair programming	30
5. Platforms and programs.....	30
5.1 Techniques and technologies used.....	30
5.1.1 Discord.....	30
5.1.2 Google Docs	30
5.1.3 GitHub/Desktop	30
5.1.4 Miro	30
5.1.5 Design Sprint 2.0	31
5.1.6 Figma	31
5.1.7 Pair programming	31
5.1.8 IntelliJ	31
5.1.9 Scrumwise.....	31
5.1.10 Atlas MongoDB Using the Atlas GUI for MongoDB to host a remote database for our data. ..	31
6. Conclusion of the Process.....	32
7. SOURCES	33

1. Introduction

1.1 The client

KPMG is a global leader in professional services, and is the client for this project. They have requested the development of a comprehensive digital onboarding application to facilitate a smooth and efficient transition for new hires. KPMG aims to ease the transition for new employees by providing an engaging and informative onboarding experience. This solution will ensure that employees feel welcomed and well informed.

1.2 The Project

KPMG is deeply committed to providing new hires with the best possible onboarding experience. To enhance the process further they want to digitalize the experience with KickStart. They have recognized a considerable gap from the time a new employee signs their contract, to their first workday. It is crucial for KPMG to maintain contact with new employees and provide them with necessary information in advance of their first workday. This is where our solution and KickStart come into play!

1.3 Problem Statement

KPMG is committed to ensuring that all new employees receive an exceptional onboarding experience. Recognizing the importance of seamless transition. KPMG aims to provide new hires with a good and comprehensive introduction to KPMG, even before their first official workday. For this to work well, KPMG seeks to modernize the onboarding process creating a more engaging and informative experience for new employees during the period from hire to the first day at work.

1.4 Objective

The objective for this project is to develop a comprehensive onboarding application tailored for presentation to KPMG. This application aims to streamline the onboarding process for new employees, seamlessly integrating a section with KPMG's culture and operations. We can do this by providing a structured, user friendly interface.

1.5 Target Audience

Our primary target audience for this onboarding application is new employees transitioning into their roles at KPMG. This application is designed to support these newcomers ensuring they have a smooth transition into KPMG. While we address the unique needs for new employees. We aim to provide a unique understanding of how things go about at KPMG.

2. The Team

2.1 Team members

- Magnus Mjåsund Hansen(E-business)
- Vetle Helling(E-business)
- Markus Østenstad Hagen (Programing)
- Tobias Fretheim(Programing)
- Elias Strømmen(Frontend and app development)
- Jonathan Lafjell Ed(Frontend and app development)
- Josef Nghia Trong Phan (Frontend and app development)
- Siyabend Göcüçü (Frontend and app development)

2.2 Roles and Responsibilities

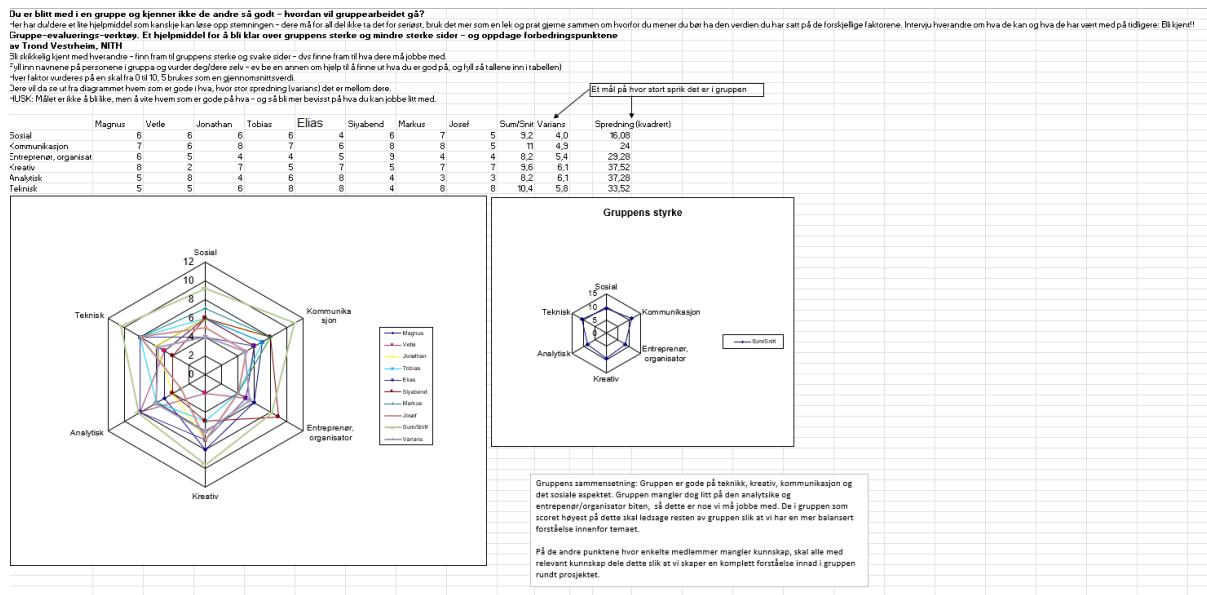
We are a group with different backgrounds based on academics. So according to this we have had different roles during this project. By doing this it helped us delegating tasks effectively and we could then leverage each member's strengths. But although roles were defined, we did take a cross-functional approach to the project. Where we often collaborated in areas where one might not be as confident because it was not their expertise, but we supported each other. This comes forward in the prototype process where every team member collaborated in Figma to get the prototype done for the interviews.

By doing this we have made sure that we have a versatile team environment. Another example was that not everyone in the group had worked with a React application before.

We have collaborated well together during this project. We have had morning meetings every day, as well as meetings during the day. Under this project we have used discord to communicate and to send files, ideas, problems etc. to each other. By doing this we have made sure that we are always up to date.

As previously stated we have used scrumwise and GitHub for the joint collaborative process.

2.3 Radartest



Before we started the project, we set up a chart picturing our different qualities. While this was primarily used as an icebreaker, it also helped us get to know each other's strengths and weaknesses.

Further on, we created a risk plan. We set up 9 risks we thought could potentially happen, and rated them by consequence and probability on a scale from 1 to 10, and 0 to 1 respectively. We had the following:

2.4 Risk Plan

Risikoplan	Gjeldende fra dato	21.05.2024			
Risiko	- Konsekvens (K)	- Sannsynlighet (S)	- Risikopeieng (R)	- Forebyggende tiltak	Tilbakavisning
Sykdom innad i gruppen	5,00	0,33	1,65	Tenk på egen helse	God kommunikasjon med gruppen og les det følgende spørre om hjelpe dels kunnskap
Forskjellig kunnskap ang. ulike programmer	1,00	0,70	2,80	Kartlegge kunnskap fra start	Kortlegge og løsning
Oppgaver ikke klart forstått	1,00	0,70	2,80	Oppgaver ved opp følge	Falsiske, reelle koda
Tekniske problemer	7,00	0,80	5,60	Kommunikasjon godt, praktisere faste følgeereiger, kommentarer	Konigere etter dialog
Ulfredstilende løsning	10,00	0,20	2,00	God dialog med kontaktperson, følge opp tilbakemeldinger, løse problemer følgepende	Skriv om evt. dierig koda
Lav kvalitet/quick fixes underveis	6,50	0,50	3,25	Skriv god koda fra start, følge felles oppsett, kommentarer i kode diskusjon	Skriv om evt. dierig koda
Ødelegging av teknisk status	1,00	0,70	0,70	Ødelegging av teknisk status på framtid	Ødelegging av teknisk status
Mangel på motivasjon	8,00	0,1	0,8	Team-meetings, oblig, oppmøte når det seg gøre, konstruktive tilbakemeldinger	Team-buildning / Ta en pause hvis vi slår for lenge fast
Før ambiens løsingen/ideer	7,00	0,50	3,50	Bygge en god løsning innenfor en realistisk tidsramme	Nedskaling, gjennomgang av hele prosjekter, hva kan vi ent klare oss uten?
TOTAL:	59,50	4,03	26,00		

2.4.1 Sickness

We saw sickness as a moderate consequence, of course depending on the severity. The probability of it happening rated 0,33, as you could in reality work from home if sick. The risk score this gave us was 1,65. In hindsight this risk actually realized itself, with one of the members being hospitalized for a few days, meaning they couldn't contribute.

2.4.2 Different knowledge about different programs

Seeing as we all came from different fields of study, we were unsure about the different coding languages, etc. We assessed the consequence of this as a 4 out of 10, and the probability as 0,7. We prevented this by mapping all our skills before we started, and we helped each other by teaching those who had lesser knowledge.

2.4.3 Privacy

This risk is not relevant due to this being only an MVP. In a fully developed solution we would add secure databases and encrypted sending and retrieving of sensitive data.

2.4.4 Technical Issues

We had some technical issues with setting up the database and the API's, but through good communication and troubleshooting we got it up and running. The consequence of technical issues could be quite substantial, and could mean we would have to change the solution we had planned.

2.4.5 Unsatisfying solution

An unsatisfying solution is what we deemed as the risk with the most consequence. But we used the feedback from our meetings with the Product Owner, and added or changed based on the input.

2.4.6 Quick fixes

Quick fixes were never a problem, because of the teamwork we had during the project.

2.4.7 No control on project status

With the help of Scrumwise we always had control over the project.

2.4.8 No/little motivation

We never felt that no or little motivation was a risk that came into play during the project.

We always had the chance to take a break if needed.

2.4.9 Ambitious solution

We thought the consequence of an ambitious solution could be a 7, with a 0,5 probability of happening. This did happen though, but we detected it early through our daily meetings, and managed to scale back our plan.

2.5 Contract

Our project contract outlines the goals, roles, procedures and expectations from the team members. It sets a goal for completion of the project and documentations of the project by June 14th 2024. We have a goal of achieving an 'A' grade and meeting the needs of KPMG.

The contract emphasizes respect and constructive feedback, we want to avoid big conflicts that will slow the project process down. This contract ensures that all of the team members are aligned and stay committed to the project's success.

UNDERSKRIFTER Sted : Discord, Dato 22.04.2024

MØH, ES, VH, SG, JLE,
JPTN, TPF, MMH

3. Development Process

3.1 Sprint planning and Execution

3.1.1 Sprint 1(Design sprint)

In the first sprint we gathered ideas for what we wanted to build in order to solve the given problem. We also started building our prototype in Figma during this sprint.

Miro process: We started the process with the design sprint 2.0. The first step of this process started when all members of the group wrote down how might we? (HKV/HMW)

Here is the HKV questions we voted for as the best options for our future of the project



We then moved forward with the long term goals. Long term goals refer to the overarching objective or outcomes that the team aims to achieve beyond the immediate scope of the sprint. Still, while the main focus of a design sprint is often on rapidly editing, prototyping, and testing solutions to a specific problem or challenge, it is essential to consider the broader horizon impact and implications of these efforts in the long term.

We then voted on which vision we thought would suit the problem best.

And we ended up with this: In two years the company is the most acknowledged in offers on onboarding solutions in Norway.

Om to år er bedriften den mest anerkjente innen tilbud av onboarding software for bedrifter i Norge



Then we had to decide on sprint questions. Sprint questions are important tools for the context of the design sprint to guide the team's focus, and make sure we are on the same page with our project goals. Sprint questions help us clarify the primary opportunities and challenges the team addresses during the sprint. By doing this we can get clear and actionable questions, the team can then maintain a laser sharp focus on the most critical aspects of the project, and drive towards a meaningful and practical solution.

Kan vi lage tjenesten både enkel og bred, men personlig og tilpasset for hver enkelt nyansatt?



Kan vi lage en tjeneste som fjerner ubesvarte spørsmål hos alle nyansatte?



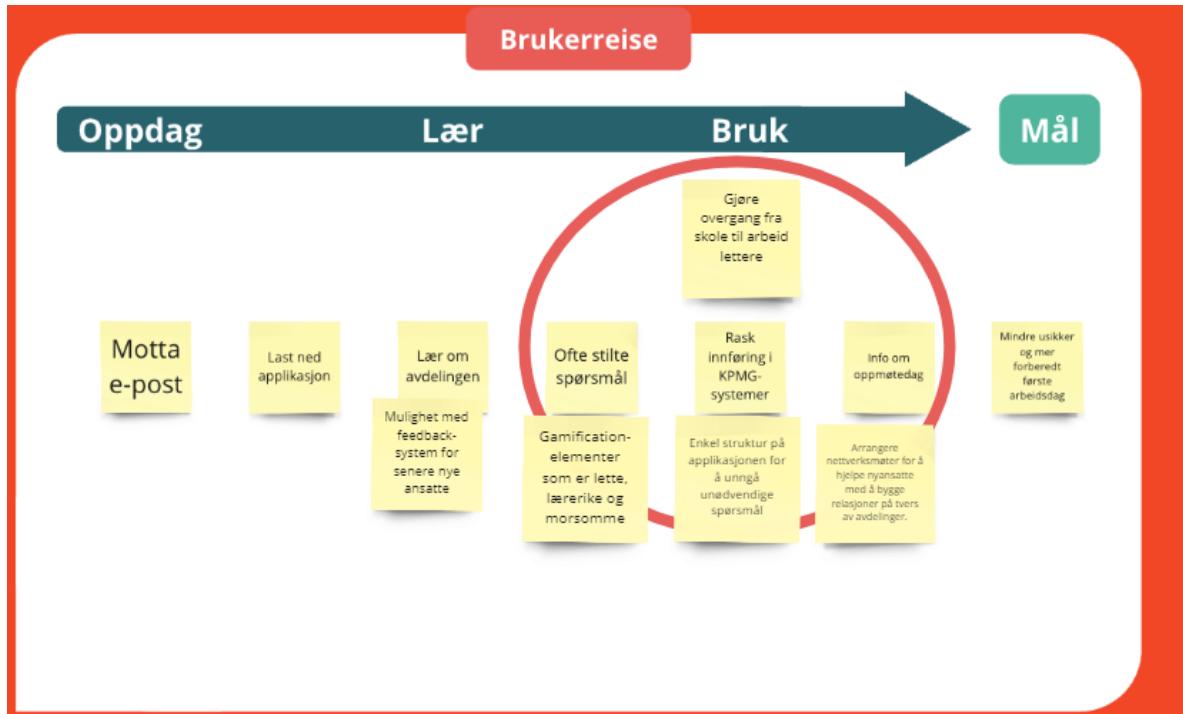
Kan vi lage en applikasjon som er lett for de ansatte å sette seg inn i?



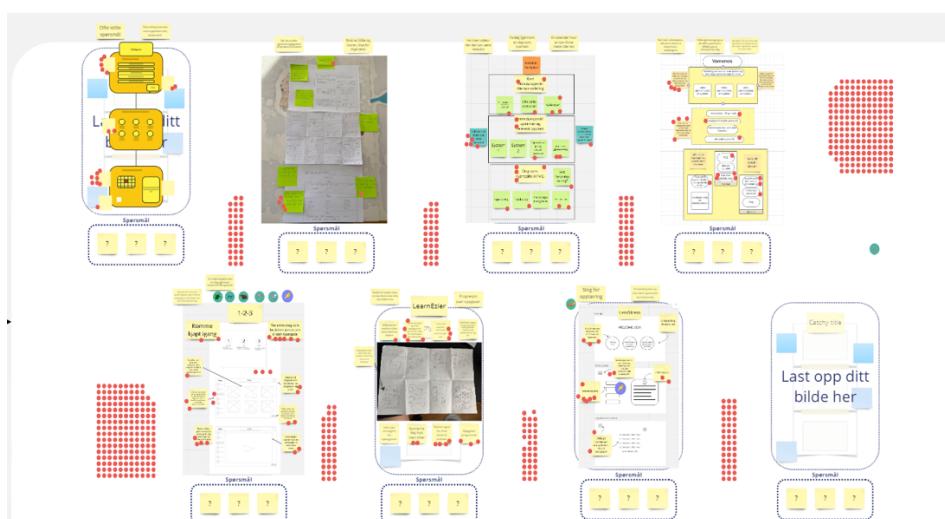
Above are the sprint questions we voted for, and what we felt was the most important for the project. This helped us find the way towards a prototype.

After we came to an agreement on the sprint questions, we went on to the map and focus area.

This is an important section because it's where we determine the most crucial focus areas. The focus area is a specific segment of the map where the sprint will concentrate. By doing this we are narrowing down the focus, so the team can tackle the most critical path of the user journey. This ensures that efforts are concentrated where we can make the most difference.



After day one was done, the group made sketches of solutions individually. Starting day two the members of the group individually placed red dots on what they thought were the best solutions and ideas. By doing this we ensure that the sprint focuses on solutions with the highest potential to address the identified problems.



We then had dots to place on the sketches and modules, we all in silence placed dots on what we liked the most. And from there we took the sketch with the most dots and made it our template for the prototype.

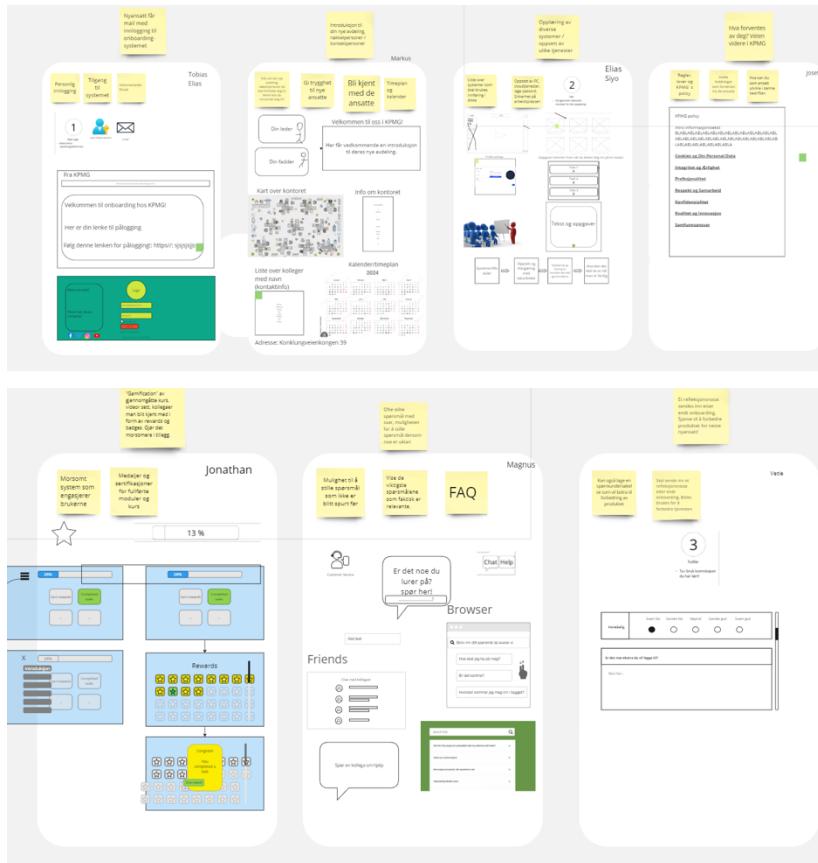
Lighting critique was the next phase. This is a rapid form of structured feedback process used to evaluate and refine the ideas for the prototype. The goal here is to gather feedback from team members thus allowing for adjustments and improvements. At the same time we went through and discussed the feedback we had orally and not just what we wrote down.



The user test flow is essential for validating how users interact with the application, ensuring it meets usability and functionality standards. This flow outlines the steps users take from receiving login credentials to navigating the application and performing key tasks. The goal is to identify the most intuitive route for users and gather feedback for improvements.



Then we were heading into the last and final part of the storyboard. The storyboard is a visual storytelling technique used to outline the user's journey or experience with a product or service. It involves creating a series of sketches to depict key interactions, decisions. It helps us as a team to visualize and communicate our ideas, identify potential pain points, and align on the desired user experience before prototyping.



We had a small retrospective after the sprint to discuss what was done and some plans for the future sprint. The first week was a full and intensive design sprint where we worked in miro as you can see.

3.1.2 Sprint 2

For the second sprint of this project, we had a morning meeting on Discord, where we planned the week ahead. Every morning at 9 or 10 am, we agreed to meet online to have a morning meeting. Not everyone could make the meeting everyday due to either illness, work, or other reasons. But the majority could be a part of those meetings, for the majority of the days. Most of the days, everyone participated in the meetings.

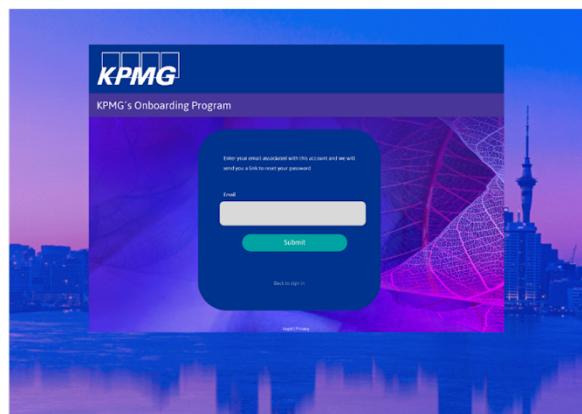
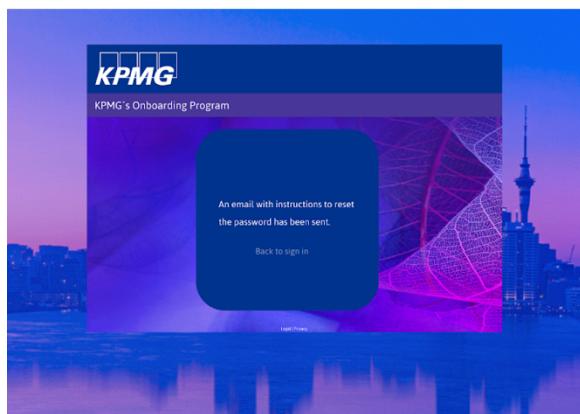
We started the week by adding the tasks to the task board in Scrumwise, and figured out the points for the different tasks, based on the difficulty/time-consumption. We then started to divide the tasks within the team. Some of us took upon adjusting the prototype in Figma, based on the results of the user tests that were conducted on the Friday of the first week of the project.

A couple of us started the setup of the React application, which we agreed as a group in the morning meeting, were the language we were going to use for this project. And the rest of us started to write on the report, which we thought was a smart thing to do, in order to properly include everything we did during the whole project.

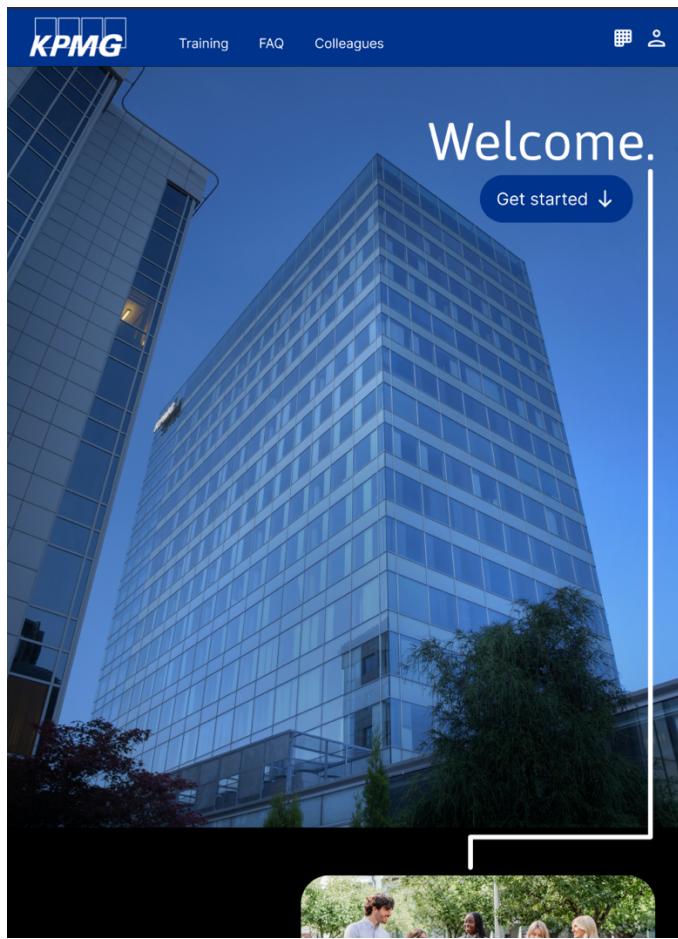
The prototype was adjusted and perfected in Figma, before we started to code in IntelliJ, using React JavaScript. Look at the screenshots from the adjusted prototype in Figma at the bottom of this section.

Come Friday, and the end of the second sprint. Now we had a good amount of code completed, and a good start to the report, which was closely followed, day by day, in order to not forget important things that we might have forgotten if we were to write the whole report in the bitter end.

3.1.2.1. Login



3.1.2.2 Landing page



The landing page features a large, modern glass skyscraper in the background. In the upper right corner, the word "Welcome." is displayed in a large, white, sans-serif font. Below it is a blue button with the text "Get started ↓".

Get to know your new co-workers at our teambuilding events.



A photograph showing five people in professional attire walking together outdoors on a paved area with greenery in the background.

Get your relevant training by our competent mentors before the start-up.



A photograph of a person from behind, wearing a black hoodie, looking at a laptop screen. The laptop screen shows another person in a video call, and the person is holding a pen and paper.



A photograph of a city skyline at night, with illuminated skyscrapers reflected in the water in the foreground.



A close-up photograph of a person's hands writing on a piece of paper with a pen.

Get to know your new colleagues



Legal Privacy Accessibility Sitemap Help Glossary Manage Choices



3.1.2.3 Calendar

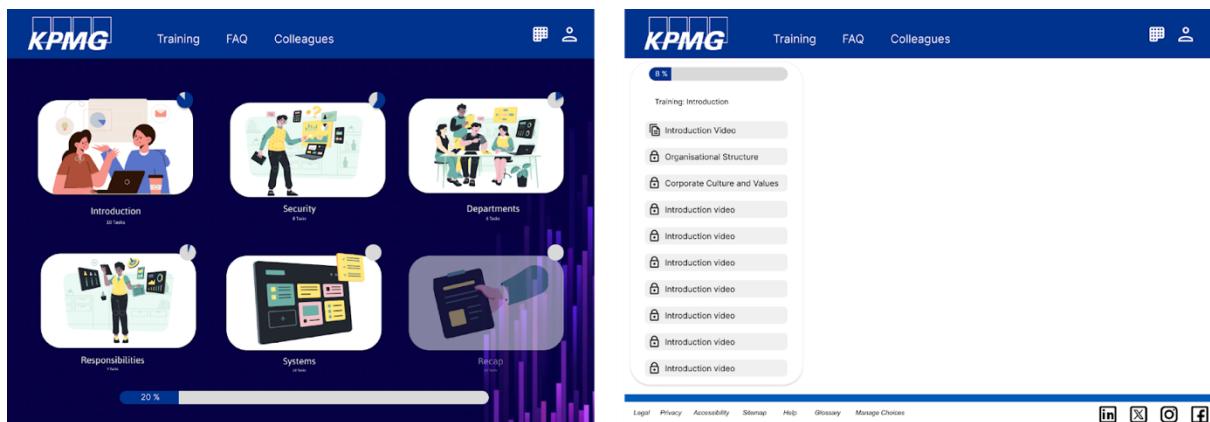
A screenshot of the KPMG calendar interface for March 2024. The calendar is dark-themed with white text. It shows the days of the week and the dates from 1 to 31. A red circle highlights the 30th. The left sidebar shows 'Events' and 'Meetings' sections.

A screenshot of the KPMG calendar interface for March 2024, showing a more detailed view with specific event blocks. The 24th is highlighted in red and labeled 'MEETING'. The 25th has a block labeled 'TEAMSMEETING'. The 26th has a block labeled 'TEAMSMEETING'. The 27th has a block labeled 'TEAMSMEETING'. The 28th has a block labeled 'TEAMSMEETING'. The 29th has a block labeled 'TEAMSMEETING'. The 30th has a block labeled 'TEAMSMEETING'. The left sidebar shows 'Events' and 'Meetings' sections.

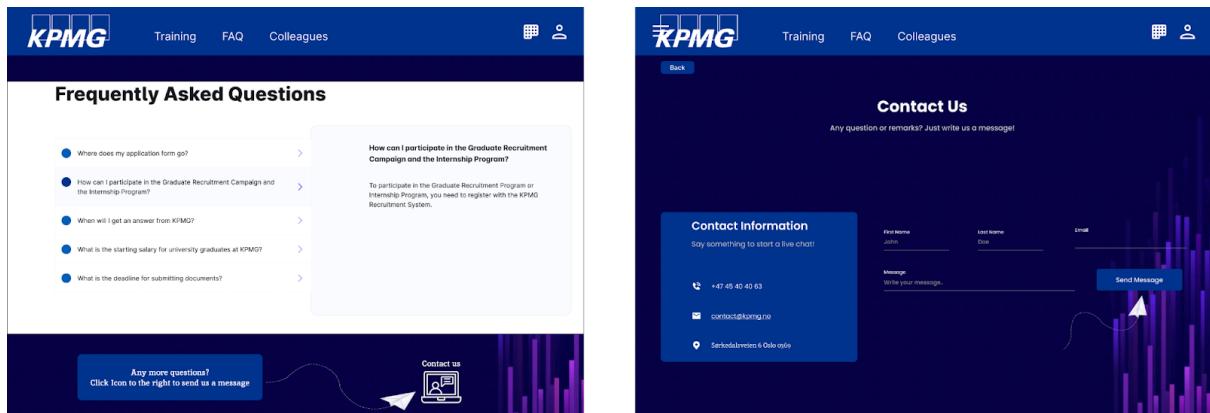
3.1.2.4 Profile

A screenshot of the KPMG profile interface for a user named John Doe. The profile includes a placeholder profile picture, the name 'John Doe', and the email 'johndoe@kpmg.no'. Below this, there are input fields for 'Full Name' (with 'Your First Name' placeholder) and 'Nick Name' (with a placeholder '...'). There are dropdown menus for 'Gender' and 'Job Title'. A section for 'My email Address' shows an entry for 'example@kpmg.no' with a timestamp '1 month ago'. A button '+Add Email Address' is at the bottom. The bottom navigation bar includes links for Legal, Privacy, Accessibility, Sitemap, Help, Glossary, and Manage Choices, along with social media icons.

3.1.2.5 Training page



3.1.2.6 FAQ (Frequently Asked Questions)



3.1.2.7 Colleagues

The screenshots illustrate the KPMG colleague search feature. The first two panels show search results for 'Your team' and 'Your department' respectively, displaying a list of colleagues with their names and department. The third panel shows a detailed profile for 'Ola Normann' with fields for full name, nick name, gender, job title, and contact information.

3.1.2.8 Feedback

The screenshots illustrate the KPMG feedback survey. The left panel shows a survey with four questions and a progress bar at 55%. The right panel shows a 'Welcome to KPMG!' page with a reflection note field and a 'Send' button.

3.1.2.9 About Us



In KPMG, our values define our actions and the responsibility we have towards our customers and society. Guided by the Global Code of Conduct, all KPMG employees are committed to conduct themselves to the highest standards both professionally and personally. We know that trust is earned and that's why we do what is right, always! Our people, in addition to the integrity that defines them, operate in accordance with international procedures and standards, necessary for all KPMG professionals.

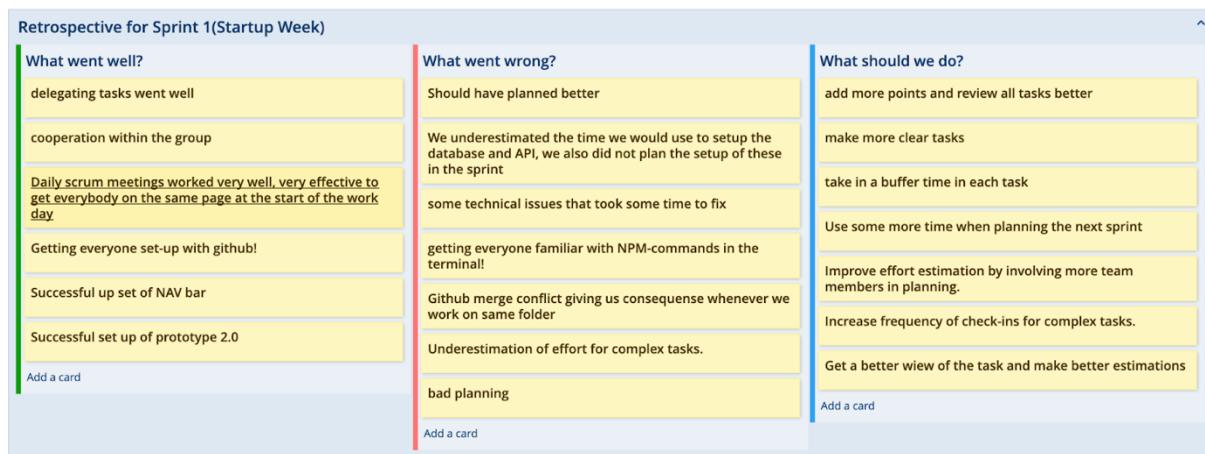
 

Global Code of Conduct Information security and privacy management policy Anti-bribery and corruption

[Legal](#) [Privacy](#) [Accessibility](#) [Sitemap](#) [Help](#) [Glossary](#) [Manage Choices](#)

[!\[\]\(9ce535ae3f041d7c0eb1883b286082dc_img.jpg\)](#) [!\[\]\(c2a4ef8aba0fe183743185a538adffc6_img.jpg\)](#) [!\[\]\(692b05d24fc6c4ac356cdd4368041ef8_img.jpg\)](#) [!\[\]\(6e2fcb89970d41a52ce6628b942539cc_img.jpg\)](#)

3.1.2.10 Sprint Retrospective results



Retrospective for Sprint 1(Startup Week)

What went well?	What went wrong?	What should we do?
delegating tasks went well	Should have planned better	add more points and review all tasks better
cooperation within the group	We underestimated the time we would use to setup the database and API, we also did not plan the setup of these in the sprint	make more clear tasks
<u>Daily scrum meetings worked very well, very effective to get everybody on the same page at the start of the work day</u>	some technical issues that took some time to fix	take in a buffer time in each task
Getting everyone set-up with github!	getting everyone familiar with NPM-commands in the terminal!	Use some more time when planning the next sprint
Successful up set of NAV bar	Github merge conflict giving us consequence whenever we work on same folder	Improve effort estimation by involving more team members in planning.
Successful set up of prototype 2.0	Underestimation of effort for complex tasks.	Increase frequency of check-ins for complex tasks.
Add a card	bad planning	Get a better view of the task and make better estimations
	Add a card	Add a card

3.1.3 Sprint 3

For the third sprint, the plan and our focus was on completing the application so we could show the PO(Product Owner) at KPMG an MVP that was ready. The team worked structured and hard throughout the week to ensure we had all the essential functions and features, and that they were being integrated correctly and tested.

We started off the week with a detailed planning session, where we outlined objectives and tasks for the week and added them to our backlog. We took use of daily stand-up meetings to track progress, address issues and keep up with each other.

By the middle of the week, we had a very robust and almost complete MVP. We worked together to finish up the pages that were missing and optimize, debugged and rewrote some of the code. Everybody was effective and good at taking use of the task board scrumwise when we worked on the backlog.

On friday, we had a presentation of the MVP for the PO at KPMG. The presentation went very well and she seemed to like our solution well. She praised the applications design and functionality, telling us that it met a lot of their needs and expectations. The PO provided some feedback on what to work on, like a better description of the route to the office which we later added.

In summary, the third spirit week was very successful, and a lot of work got done. The positive feedback from the PO gave us a lot more confidence and motivation going into the final sprintweek.

3.1.3.1 Sprint Retrospective results

The screenshot shows a digital retrospective tool with three main columns:

- What went well?** (Left Column):
 - More detailed sprint planning
 - Smoother work experience
 - Teamwork
 - Clearer tasks
 - Great task delegation
 - Completion of pages
 - Daily scrum meetings
 - The switch to MongoDB suited our needs better!!
- What went wrong?** (Middle Column):
 - some merge conflicts
 - some cache issues
 - had to switch from MySQL to MongoDB
 - Minor bugs
- What should we do?** (Right Column):
 - make better use of story points
 - remember to update task board more frequently
 - add a time buffer for unplanned urgent tasks

At the bottom of each column, there is a "Add a card" button.

3.1.4 Sprint 4

In the fourth and final sprint, we shifted our main focus over to the report writing and preparing for the final deliveries, we only had some small issues left to fix at the application, so while some stayed on the code to complete it, the rest shifted all their focus over to the reports.

We began the week with our usual sprint planning and adding off tasks to the sprint backlog from the product backlog during our daily scrum.

With most of the coding behind us, most of the members continued on the report writing and held some meetings to summarize the work done and delegate different areas to focus on the report writing. A few members finished up some small bugs and details on the code before also joining in on the continued report writing.

We recorded the presentation video on Tuesday, which showed off the applications design and features. We kept the presentation simple with everyone introducing themselves, their study program and what they had worked on in the application. After the introductions, one member of the group held the presentation and went slowly through it as he explained the choices of design and features.

Thursday was the final day of the sprint and the day we wrapped up the report writing, we all sat down and reflected upon the work we had put in the last month. We all ended up pretty happy with the solution.

In summary for the fourth and final sprint week, the transition from the main focus on coding to report writing and finalizing all the deliveries was as smooth as we could have wished for. All the groundwork from the earlier sprints made the final week very much less stressful for us.

3.1.4.1 Sprint Retrospective results

Retrospective for Sprint 3		
What went well?	What went wrong?	What should we do?
Completed the code	Merge issues	Add a card
Good teamwork	Bugs	
Even more efficient meetings		
Less reliability on each other		
Effective report writing		
Fixed some crucial bugs at the end		
Add a card		

4. Methodology

4.1 Scrum

Scrumwise is an agile management tool made and designed to facilitate the implementation of scrum. This provides a good visual and interactive platform for teams to manage sprints, see/track progress and collaborate effectively.

4.1.1 Use of Scrum

The use of Scrum has proven to be highly valuable for us, enabling us to efficiently plan our weekly activities, assign tasks, and adhere to deadlines. This framework allows us to monitor our progress and ensure timely completion of tasks.

Before diving into our backlogs, we hold a planning meeting to discuss and prioritize the most critical tasks. During these meetings, we determine what needs to be done and whether tasks need to be divided to achieve our goals more effectively.

Each day begins with a morning stand-up meeting where we review our progress and address any challenges. This daily sync helps us maintain an overview of our progress and identify if anyone needs assistance.

Post-meeting, we organize into different Discord voice channels based on our focus areas. Team members working on the client-side (frontend) collaborate in one channel, while those working on the server-side (backend) gather in another. This separation ensures focused and efficient collaboration tailored to our respective tasks.

4.1.2 Product backlog

The product backlog was used and maintained in Scrumwise, where we listed all user stories, tasks and bug fixes. Each item was based and prioritized on its value and importance

4.1.3 Sprint planning

During the sprint planning meetings, we started off by discussing which user stories to select from the product backlog to include in the sprint backlog. Scrumwise was an easy application to use because it allowed us to easily drag and drop items into the sprint backlog, providing us with a good visual of the selected tasks.

4.1.4 Task tracking and progress monitoring

Scrumwise came with a good visual Kanban board where tasks were visually represented like: To do, In progress and done. This helped us as a team to understand the status of each task and made it a lot easier for us to see which tasks were being worked on and it made it so we did not have to constantly check in on each other to avoid working on the same task.

4.1.5 Burndown chart

This helped us by showing a chart over the team's progress over time. This helped with monitoring the remaining tasks and time, which also allowed us to make adjustments. Even though it is not visually perfect, we can see the improvement from the first sprint we had.

4.1.6 Daily meetings

In every morning meeting we had. Scrumwise was a good tool for us to see and review the progress, discuss any problems we might have and then plan the day.

4.1.7 Collaboration and communication

Communication: Under the project we had a discord chat where we could communicate with each other. Here we mostly talked during the project process. And we could keep every member up to date, even if they were not in the meeting, by posting an update in one of the text chats.

4.1.8 Retrospective and continuous improvements

In our retrospectives, we will use Scrumwise to document what went well and what didn't. Throughout this project, we adhered to scrum principles, conducting one-week sprints excluding weekends.

Our first sprint, while ultimately successful, was somewhat chaotic due to inadequate planning and our inexperience with scrum. However, by the second sprint, we had improved significantly. We established a more strategic plan, better task delegation, and meticulously updated the backlog with even the smallest changes before starting. This clarity in our tasks greatly enhanced our efficiency.

As the project progressed, our understanding of scrum deepened, and we found it increasingly effective. Overall, we all found scrum to be highly beneficial for our workflow

When you look at the burndown charts we were above the line at almost all times but we felt we had good control and knew that we were going to reach the finish line in time. We always had morning meetings to check in on progress so that we all knew where we were standing. And during the day we also had little check ins.

4.2 Repository Management

4.2.1 GitHub Repository

The team used GitHub as the remote repository hosting service. This allowed for centralized storage of the project code and facilitated collaboration amongst team members.

4.2.2 Cloning

Team members cloned the GitHub repository to their local machines to work on the project.

4.2.3 Branching strategy

During the span of the project we decided on having just one Main branch. While this approach is not a good practice in general, due to potential merge conflicts and other factors, it was a conscious decision based on the general group composition.

Some members of the group had little to no experience with git and GitHub and the idea of having multiple branches appeared confusing during a little tutorial we did together. By having one branch and aiming to streamline the workflow, as well as reducing the learning curve, we ensured that every team member could efficiently collaborate on the project.

This meant we could focus on building the application together, while increasing the team's knowledge of version control and resolving the conflicts that appeared together as a group.

4.2.4 Regular Commits

Team members made regular commits with descriptive messages to document their progress and changes. This practice made it easier to track changes and revert to previous versions if necessary.

4.2.5 Pull Requests

Changes to the project were made by pulling requests. We merge branches. We have also written a code standard that we follow. This is crucial for a good consistency, collaboration, quality, documentation and maintenance.

4.2.6 Collaboration and code Review

Code review: Pull requests allowing for team members to comment on code, and maybe catch bugs early. This process enhanced code quality and shared knowledge amongst the team.

4.2.7 Pair programming

During multiple stages of the project, the group opted to use pair-programming during initial setup of different services or when facing bugs. An example is when we started to set up the backend-side of the project, because this was a very complex problem that affects the whole application further on. Thus allowing for a faster setup and higher-quality code.

4.2.8 Handling conflicts and challenges

We as a team had very good communication throughout the span of the project. There were no big conflicts, but some discussions occurred and we see this as a positive thing! Through discussions, the best solutions appeared. Our daily meetings and general cooperation made for a fun time with good conversations about the solution, as well as life in general!

4.2.9 Regular Commits

Team members made regular commits with descriptive messages to document their progress and changes. This practice made it easier to track changes and revert to previous versions if necessary.

4.2.10 Pull Requests

Changes to the project were made by pulling requests. We have also written a code standard that we follow. This is crucial for a good consistency, collaboration, quality, documentation and maintenance.

4.2.11 Collaboration and code Review

Code review: Pull requests allowing for team members to comment on code, and maybe catch bugs early. This process enhanced code quality and shared knowledge amongst the team.

4.2.12 Pair programming

During multiple stages of the project, the group opted to use pair-programming during initial setup of different services or when facing bugs. An example is when we started to set up the backend-side of the project, because this was a very complex problem that affects the whole application further on. Thus allowing for a faster setup and higher-quality code.

5. Platforms and programs

5.1 Techniques and technologies used

5.1.1 Discord

Used for setting up daily scrum-meetings and general communication within the group.

5.1.2 Google Docs

Used to easily write together and keep track of ideas and documentation throughout the exam.

5.1.3 GitHub/Desktop

We created a github repository early on for co-development. This has worked very well, despite a few merge-conflicts, but resolving these gave everyone valuable experience!

5.1.4 Miro

A digital design platform used to work in teams/groups for the sprint design weeks. This was used for a better understanding of the project and learning throughout this exam.

5.1.5 Design Sprint 2.0

The usage of design sprint was done through a time limited five-phase process (Empathize, Define, Ideate, Prototype and Test.) that uses design thinking to, intended to minimize the risks with launching a new product.

5.1.6 Figma

We used Figma to create and visualize our ideas from the design sprints done in Miro. With the use of Figma we could progress better to a better prototype and see what we wanted to improve on.

5.1.7 Pair programming

Used to help each other when facing bugs, and get a broader knowledge of the application as a whole.

5.1.8 IntelliJ

Project is coded in the JetBrains IntelliJ IDE.

5.1.9 Scrumwise

Used for helping us working agile with scrum and plan and work on our sprints.

5.1.10 Atlas MongoDB

Using the Atlas GUI for MongoDB to host a remote database for our data.

6. Conclusion of the Process

During this project we have successfully developed an efficient and user-friendly application for onboarding services for KPMG. This application addresses the needs and enhances the onboarding experience for employees.

Our team focused on making a solution that is functional while also intuitive, thus making the onboarding process smoother. By following a structured development process in agile, we were able to implement features that cater specifically to KPMG

The completion of this project marks a milestone in improving KPMG's onboarding services. And we are confident through feedback that an application like this will benefit KPMG and new employees.

7. SOURCES

The familiar,(2020, November 26). (Sprint questions). Retrieved 14:20, May 30. 2024, from
<https://www.thefamiliar.tech/blog/2020-11-26-sprint-questions-the-design-sprint-unpacked>

The familiar,(2020, December 3. Hutchings Sam). (Mapping the design sprint). Retrieved 15:04, May 30. 2024, from

<https://www.thefamiliar.tech/blog/2020-12-03-mapping-the-design-sprint-unpacked>

Design Sprint

<https://design-sprint.com/understand/>

Wikipedia contributors. (2024, May 6). Scrum (software development). In *Wikipedia, The Free Encyclopedia*. Retrieved 09:13, May 31, 2024, from

[https://en.wikipedia.org/w/index.php?title=Scrum_\(software_development\)&oldid=1222507636](https://en.wikipedia.org/w/index.php?title=Scrum_(software_development)&oldid=1222507636)

OpenAI. (2024). ChatGPT (June 12. 2024 Version) [Big language modell].

<https://chatgpt.com/>