# DAEDALUS

Course code: BOP3000

Members:

Emre Olgun

Håkon Larsen

Roy Alexander Daae

Øystein Furuli Wangen

Glenn Andre Parsons Halvorsen

Supervisor:

Boban Vesin

# Summary

This report is written and made by the Daedalus group in conjunction with our bachelor thesis in IT & Information Systems at the University of South-east Norway, spring of 2020. It describes the different processes and work done to create a web application that can bring together students in teams and find businesses with assignments for them to do.

After a meeting with USN Digital, we learned that there we were open to use whatever technology or programming requirements from their side. As the group had all worked with .NET applications before, after a short discussion we chose to use it for our application in conjunctions with React, as some of our members had experience with that as well. Everyone started out with learning and building competence with .NET and React before the project got started and in the planning phase in the beginning.

The main tools used during the development are:

- GitHub (Version control)
- Microsoft Visual Studio and Azure database (Software)
- .NET, SQL and TypeScript (Programming)

# Table of Contents

**Summary** 1

**Figure list** 4

**Table list** 5

**Preface** 6

**1.0 Introduction** 7

1.1 Group members 7

1.2 Our Client 8

1.3 Why this assignment? 8

1.4 What will we create? 8

**2.0 Analysis** 9

2.1 Stakeholders 9

2.2 Project Goals 10

2.2.1 Main Goal 10

2.2.2 Goals & Milestones 11

2.2.3 Initial Goals 11

2.2.4 Deliverables 12

2.2.5 Tasks 12

2.2.6 Costs 12

2.2.7 Deadlines 13

2.3 Client Needs Assessment 14

2.3.1 Users 14

2.3.2 Use case 15

2.3.3 User stories 16

2.3.4 User Scenarios 17

2.3.5 Functional Requirements 22

2.3.6 Non-functional requirements 23

2.4 System Architecture 25

2.4.1 Main Component Diagram 25

2.4.3 ER-Diagram 26

2.4.4 ER-Diagram Explanation 27

2.5 Risk Analysis 27

2.6 Target Audience 29

2.6.1 Personas 30

2.7 Privacy & GDPR 32

**3.0 Design** 33
3.1 User Groups 33
3.2 Sequence Diagram 33
3.2.1 Student login & join team 33
3.2.2 Business create assignment 34
3.3 Quality Control 34
3.3.1 “In-House” Testing 34
3.3.2 User Tests 35
3.4 Chosen Technology 35
3.4.1 ReactJS 35
3.4.2 .NET Core 35
3.4.3 TypeScript 36
3.4.4 MySQL 36
3.4.5 Azure 36
3.4.6 Axios 36
3.4.7 MobX 37
3.4.8 Visual Studio 37
3.4.9 GitHub 37
3.5 Safety Measures 38
3.6 Wireframes 39
3.7 Communication & Tools 45
3.7.1 Discord 45
3.7.2 Google Docs 45
3.7.3 Facebook Messenger 45

**4.0 Project Execution** 46
4.1 Methodology 46
4.2 Project Management 46
4.2.1 Management Tool 47

**5.0 Reflecting** 48
5.1 Workflow 48
5.2 Experiences 49
5.3 Future Development 50

**6.0 Results** 51

**7.0 References** 53

# Figure list

Figure 1: Main Component Diagram……...…..26
Figure 2: ER-Diagram……....27
Figure 3: Student Sequence Diagram……....34
Figure 4: Business Sequence Diagram……..35
Figure 5: Wireframe Frontpage……..40
Figure 6: Wireframe Student Profile……..41
Figure 7: Wireframe Business Profile……....41
Figure 8: Wireframe My Assignment…….....42
Figure 9: Wireframe Create Assignment……...42
Figure 10: Wireframe Request……...43
Figure 11: Wireframe Team Request…….....43
Figure 12: Wireframe Find Team…….....44
Figure 13: Wireframe Team Accept Requests……...44
Figure 14: Wireframe Team Page…….…45
Figure 15: Trello…….…...48

# Table list

Table 1: Stakeholders…….....10
Table 2: Initial Goals…….....12
Table 3: Deadlines……..14
Table 4: Users……....15
Table 5: Use Cases…….16
Table 6: User Stories…….....17
Table 7: User Scenarios…….18
Table 8: Functional Requirements…….23
Table 9: Non-functional Requirements……..24
Table 10: ER-Diagram Explanation……...28
Table 11: Risk Analysis…….....28
Table 12: Personas…….……31
Table 13: Safety Measures…….39
Table 14: Future Development……...51

# Preface

This report is written for bachelor thesis spring 2020 (BOP3000), at the University of South-East Norway under the IT & Information Systems course.

The Daedalus group was formed at the end of 2019 as all the members had worked together previously with good results. After looking through the different assignments available, the choice fell on the “Oppgave dating” that was given out by the USN Digital, as most members thought it would be interesting to work with.

The Daedalus group has throughout the semester developed a web application to help students find other students to work with, as well as let groups of students find work assignments posted by various organizations and businesses.
The application will allow businesses to post assignments they want done by students, to then have students apply to take on these assignments. It will have a message system to help them get in contact with each other to set up further cooperation.

We have been in dialogue with USN Digital through Henning Tollefsen and feel like we have gotten good help from their part with what we need. We also got good information on what the client wanted to see in the application and what needed to be changed.

We want to thank USN Digital for giving us the assignment and our contact Henning Tollefsen, as well as our supervisor from USN, Boban Vesin.

# 1.0 Introduction

## 1.1 Group members

**Emre Olgun**

Email: emre.olgun@email.com

LinkedIn: https://www.linkedin.com/in/emre-olgun-8668b41a6/

- 23 years, IT & Information Systems.

**Håkon R. Larsen**

Email: haakon.larsen@outlook.com

LinkedIn: https://www.linkedin.com/in/h%C3%A5kon-larsen-289ba3178/

- 23 years, IT & Information Systems.

**Roy Alexander Daae**

Email: Micro_atx@hotmail.com

LinkedIn: https://www.linkedin.com/in/roy-alexander-daae/

- 30 years, IT & Information Systems.

**Øystein Furuli Wangen**

Email: oystein.f.wangen@gmail.com

LinkedIn: https://www.linkedin.com/in/%C3%B8ystein-furuli-wangen-bb5430161/

- 29 years, IT & Information Systems.

**Glenn Andre Parsons Halvorsen**

Email: glennandreph@gmail.com

LinkedIn: https://www.linkedin.com/in/glennandreph/

- 27 years, IT & Information Systems.

## 1.2 Our Client

USN Digital is a part of the university where the main mission is connecting the students from all the different campuses together in one digital campus. The task we have taken will help with connecting students together and also with businesses that got assignments they want done. An example of this is can be bachelor students looking for a thesis and IT Business X having one that they post on the platform.

## 1.3 Why this assignment?

We thought the concept was interesting, while also seeing the potential to challenge ourselves. Most of the group have not used React prior to this and since React is used a lot out in the workfield, we thought this assignment would be a good fit.
It also allows us to create a platform that can help other students to find teams to work in and potentially even the subject for the thesis.

## 1.4 What will we create?

The application we are creating is a platform for students to find other students to work with, as well as a place for businesses to post work they want done by students, to allow students to browse and apply for the work. The goal is to connect the two sides to allow students to gain work experience in their field and to allow businesses to get projects done and to find new potential employees should they be happy with the work done.

# 2.0 Analysis

## 2.1 Stakeholders

The stakeholders which are the most relevant for us, are the project owners, employees, investors, shareholders, companies, users of the application and competitors. We have both internal and external stakeholders.

| |
|---|
| ● The shareholders might directly impact the company's decisions. Certain decisions might have a bigger impact on the company that can lead to losing or gaining shareholders that influences the decision making of the company. Want neat features that take little time to market, lower costs and higher profitability, application on par with competitors. |
| ● The project owners want a successful application with the least amount of resources wasted. Depending on the interest of investors and shareholders, along with employees and users. They usually have the final say on decisions unless a shareholder owns more of the company/project. |
| ● Investors are looking for a platform that is usable and accessible. See a future in the application which in return can make them profit or help them promote their company. Can influence decision making. An unpopular decision can lead to investors losing interest. |
| ● Developers are in charge of developing and maintaining the application. Collecting information and feedback from users. Updating the database. They influence features implemented and the time and cost of the development process. |

| |
|---|
| ● Employees are administrators hired to maintain the website. They want a fair salary. Are in charge of managing users, companies, reports, reporting bugs to developers, user support etc. |
| ● Users of the website are interested in viewing and using the application, find assignments of a certain category, and join teams. They want usability, security, accessibility and depend on the stability of the application. Influenced by certain features which can lead to loss and gain of users and potential companies. |
| ● Companies want to find new employees or teams to complete an assignment. They want to single out the skilled from the bad. Want a smooth experience and easy-to-use application. Impacted by the performance and features of the application along with the users skill level. |
| ● Competitors affected by the success of our application. The more quality users and companies we have, as well as the better performance and usability will lead to more users and companies picking our application over the competitor. Resulting in loss of users for the competitor. |

*(Table 1: Stakeholders)*

## 2.2 Project Goals

### 2.2.1 Main Goal

Our main goal is to implement all the core functionality and features required to make the web application usable with a simple user friendly interface and good

performance. This will hopefully result in delivering a pleasant user experience, which is also one of our main goals.

### 2.2.2 Goals & Milestones

Completing features within reasonable time depending on difficulty. We have a short timeline to complete the application and report, so it's important that we follow the timeline and reach our milestones in time to make sure we are able to deliver an application that is as optimal as possible.

### 2.2.3 Initial Goals

| |
|---|
| Simple UI |
| CRUD React |
| CRUD ASP.NET Core |
| Message System |
| Login/sign up with authentication |
| Filter/Search Assignments/Teams/Students |
| Team System |
| Assignment System |

*(Table 2: Initial Goals)*

### 2.2.4 Deliverables

An easy to use and secure web application where different users have access to different functions. The goal is to have moderators to keep the application functional and not require developer support. The end goal for the application is that it's usable and can be implemented to USN Digital for them to use for their students and partners. In that case the application will either result in us further developing it for USN Digital or they will take over the development. A project report also needs to be submitted.

### 2.2.5 Tasks

To assign tasks we are using Trello.com to organize which tasks need to be done and can assign members that are working on them. This helps us to know which members are working on a certain feature at that moment, so that we don’t do something twice or mess up for each other. We also have weekly meetings to discuss what we have done, what needs to be done and who should do it. GitHub also makes sure that there are no conflicts and that everyone has the latest version of the files available.

### 2.2.6 Costs

The main costs for this application will be operation and maintenance, as well as future developments. The operation costs include hosting the database and domain, as well as keeping a few employees to run the application. The maintenance costs will also have keeping employees employed to work on the application to further develop and fix it as needed if any problems were to be discovered.

### 2.2.7 Deadlines

These are the requirements that are planned to be done by the 21th of May, but changed to the 24th with the updated delivery times:

| Functions/Tasks | Requirements |
|---|---|
| Database | ● ER-Diagram<br>● Connect DB to .NET and React<br>● Create Tables |
| Authentication | ● Login<br>● Logout<br>● Sign up<br>● Security Features |
| Assignments | ● CRUD<br>● Filter/Search<br>● Send Requests<br>● Accept/Decline Requests |
| Team | ● CRUD<br>● My Profile<br>● Send requests<br>● Accept/decline Requests<br>● Overview of members & Assignments |
| Students | ● CRUD<br>● My Profile<br>● My Teams<br>● My Assignments<br>● Inbox |
| Businesses | ● CRUD<br>● My Assignments<br>● Create Assignments<br>● My Profile<br>● Inbox |
| Category | ● CRUD |
| UI | ● Web Application |
| UX | ● Responsive<br>● Useable on Mobile<br>● Easy to use design |

| | |
|---|---|
| Project Report | |

*(Table 3: Deadlines)*

## 2.3 Client Needs Assessment

### 2.3.1 Users

| User | Functions |
|---|---|
| Student | ● Register<br>● Search for teams & assignments<br>● Apply for teams<br>● Create/join/leave/delete team<br>● Chat with users |
| Team | ● Create/edit/delete ad - looking for student(s)<br>● Add members, remove members<br>● View applications, decline & accept<br>● View assignments, send application<br>● Update latest status |
| Business | ● Register<br>● Create, edit & delete assignment<br>● View applications, decline & accept<br>● Chat with with users |

*(Table 4: Users)*

### 2.3.2 Use case

| User - Send application |
| --- |
| 1. Users go to overview for teams & assignments.<br>2. Selects the team they are interested in.<br>3. Click on “Join” and await confirmation. |

| Business - Accept application |
| --- |
| 1. A team member goes to the discover page and clicks on an active assignment.<br>2. Selects chosen team and applies for assignment.<br>3. Application comes up with an option to accept or decline. The business then accepts and the team is added to the assignment. |

| Business - Create assignment |
| --- |
| 1. Business goes to their overview<br>2. Clicks on create assignment and fills in requirements & goals with their assignment.<br>3. Sets a deadline and clicks on “create assignment”. |

*(Table 5: Use Cases)*

### 2.3.3 User stories

| User Group | Functions |
|---|---|
| Students | ● I want to find an assignment to work on.<br>● I want to find a team to collaborate with.<br>● I want to start a new team to work on a specific assignment.<br>● I want to make new connections and possibly find a job.<br>● I want to expand my experience and face new challenges. |
| Teams | ● We want to find new students to join our team.<br>● We want to find assignments that match our team skills & experience.<br>● We want to message applying students before selecting and denying students to join our team. |
| Businesses | ● I want to create assignments to be completed.<br>● I want to select and deny teams to work on my assignment.<br>● I want to chat with students/teams before selecting them for my assignment.<br>● I want to find new potential employees. |

*(Table 6: User Stories)*

## 2.3.4 User Scenarios

| **[Frank Jackson]** | | |
|---|---|---|
| <br>[1] | ● Has intermediate experience with websites.<br>● Business owner. | ● Find employees.<br>● Find teams to complete outsourced assignments.<br><br>**Key questions:**<br>● What qualities do the students need to complete my assignment?<br>● If I'm satisfied with the result, should I employ the student(s)? |

| **Scenario [1] – [Finding new employees]** | |
|---|---|
| Frank needs new employees for his company. He is not sure what qualities the students need, but based on the result from assignment. He will consider hiring a student or possibly more students from the team. | **Importance: 8**<br>**Frequency: 3** |

| **Scenario steps** | **Comments** | **Outstanding questions** | **Required functionality** |
|---|---|---|---|

[1]

| | | | |
|---|---|---|---|
| Frank enters the website as an unregistered user. | Frank has heard about the website from a colleague. | Will Frank know about the website? | There needs to be a way for him to know about the website. |
| Frank registers as a business. | A new page opens. | | The site needs to be universally made. |
| Frank clicks on Discover in the navigation bar. | A new page opens. | | Needs to be a “Discover” link he can click on the navigation bar. |
| Frank sees assignments, teams & students on a list and can see that he can filter based on categories. | A filter with categories shows up. | Does Frank know which categories to look for? | The page must have filters. |
| Frank filters the list based on teams and clicks on the first team. | A new page opens. | Will Frank know that he can click on posts to view them? | The page needs to have descriptions for each post. |
| Frank goes back to overview and clicks on create assignment. | A new page opens. | Will Frank know what to put in the description? | Needs to have a button for creating assignments. |
| Frank waits a few days and comes back to the site where he sees new updates. | The notification bell in the top right corner will be red with the number of notifications unread. | Will Frank know what the bell means and click on it? | Needs to have a notification bell which is updated actively upon applying students for the assignment. Simple UI design with reactive icon. |
| Frank clicks the bell and on the latest update. He is brought to his assignment page and the latest application is opened. | A new page opens. | Does Frank know to click the notification to bring him to it? | Notifications need to be clickable and bring you to the selected notification. |

| | | | |
|---|---|---|---|
| Frank reads the application and decides that he wants that team to complete it. | Option to accept or deny application below description. | Does Frank know about the other applications? | Needs option for accepting or denying applications on the selected application. |
| Frank checks in frequently and keeps in contact with the team while the assignment is completed. After its completion he is happy with the result and wants to employ one of the students. He clicks on the student he wants to employ to send him a message. | A new page opens with the student profile and a chat button. | Does Frank know where to find the single student in a team and that he can communicate with him/her privately? | Needs a student profile with a chat button. |

| **[Kylie Grover]** | | |
|---|---|---|
| [2] | • Likes to socialize online.<br>• Is very experienced with websites. | • Find teams..<br>• Find assignments.<br><br>**Key questions:**<br>• Will I find a team that suits me?<br>• Are there assignments that I can complete for my experience & skill level? |

| **Scenario [2] – [Finding a team]** | |
|---|---|
| Kylie is looking to find a new team to collaborate with. | **Importance: 7**<br>**Frequency: 5** |

[2]

3

| Scenario steps | Comments | Outstanding questions | Required functionality |
|---|---|---|---|
| Kylie enters the website as an unregistered user. | Frank has heard about the website from a fellow student. | Will Kylie know about the website? | There needs to be a way for her to know about the website. |
| Kylie registers as a student. | A new page opens. | | Needs a register function with options about what user you are. |
| Kylie clicks on “Discover” to look for teams. | A new page opens. | | Need a list over teams, assignments etc. |
| Kylie filters the list to only show teams. | | | Filters on Discover page |
| Kylie clicks on a team she is interested in. | A new page opens. | | Needs a description page for each post. |
| Kylie clicks on “send application”. | A new page opens. | | Needs a button for “send application”. |
| Kylie writes a short application and clicks on the button send. | | | Needs a text box to write an application and button to send it. |

3

| | | | |
|---|---|---|---|
| Kylie receives an email with an update that she has been accepted into the team. | Email sent when a notification appears on the site. | Did she sign up to receive emails? | Needs an email service for notifications that has the option to sign up for or unsubscribe from. |
| Kylie clicks on the notification from the email. | A new page opens. | | Email needs a clickable link to the notification. |
| Kylie lands on the team page where she is now a member and can see the latest status posts. | | Does she need to chat with the other team members? | Team page with members & status posts. |

*(Table7: User Scenarios)*

### 2.3.5 Functional Requirements

We have decided to put functional requirements into two tables to separate the web and desktop application. The tables include three topics:

**What:** The name of the functional requirement.
**Description:** A short and precise description of the functional requirement.
**Priority:** What priority the functional requirement has in the development process. 5 is the highest priority, 1 is the lowest priority.

| What | Description | Priority |
|---|---|---|
| User registration | Users are able to create a user account which gives permission to create posts, create/join teams; send applications, work on assignments. | **5** |
| User login/log out | Users able to log in and log out when visiting the site after registering. | **4** |
| CRUD Users | Users able to be created, viewed. Business users can edit their company info. | **5** |
| CRUD Assignments | Create assignments, view them, edit them and delete them. | **5** |
| CRUD Teams | Users are able to create, join, invite to, and leave teams. | **5** |

| | | |
|---|---|---|
| CRUD Requests | Users able to send requests & answer requests. | **5** |
| Overview | List of teams looking for students and businesses looking for assignments to be completed. | **4** |
| Message/Inbox function | Users are able to communicate with each other. | **3** |
| Filter on Overview | Filter assignments on the overview list | **3** |

*(Table 8: Functional Requirements)*

### 2.3.6 Non-functional requirements

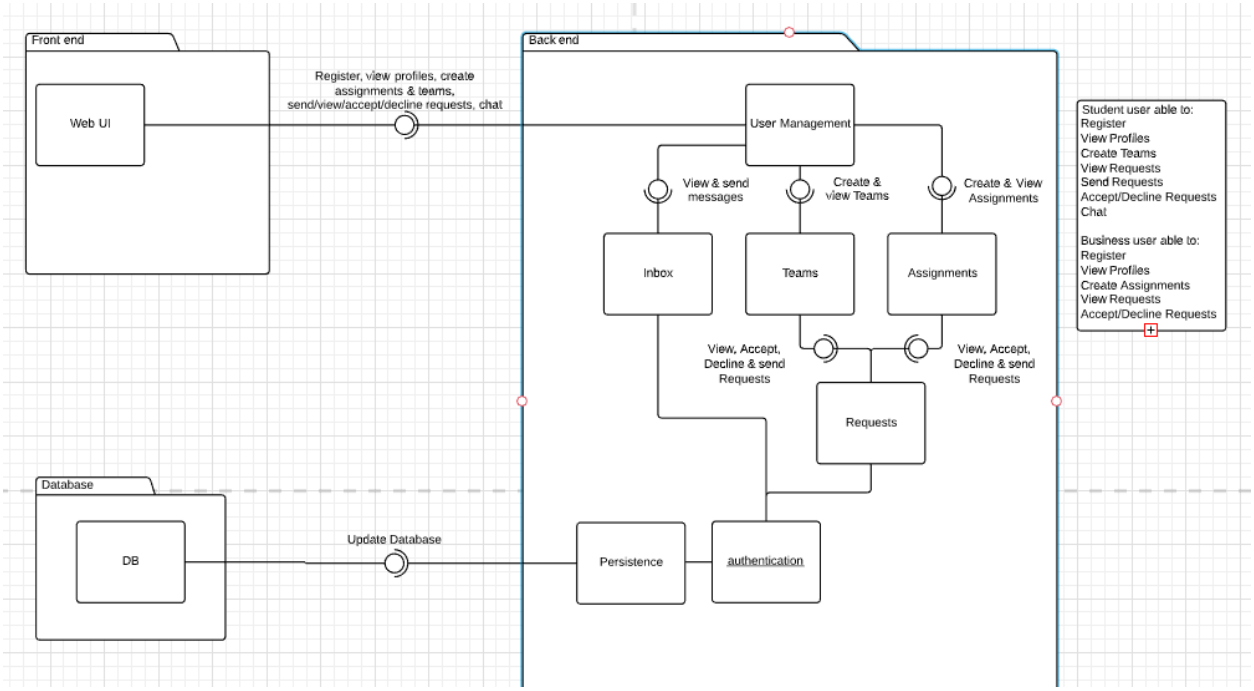
| **Requirement** | **Description** |
|---|---|
| Performance | ● Pages should load within 0.35 seconds.<br>● Database requests should be returned within 0.5 seconds. |
| Security | ● All form inputs should be SQL injection proof.<br>● Completely automated public turing test to tell computers and humans apart to prevent spam.<br>● All passwords should be encrypted with ASP.NET Core System.Security.Cryptography.<br>● The system should have end-to-end encryption using SSL/TLS-certificate(s).<br>● The system should have DDoS protection to prevent downtime.<br>● The system should have brute-force protection to prevent unauthorized logins. |
| Usability | ● The UI should be simple and user friendly for any user.<br>● Performing any action should take a few clicks.<br>● Minimal, if any documentation, should be necessary for the user.<br>● Adhere to WCAG 2.0. |
| Support | ● Users should be able to recover their account if they lose their password. |
| Availability | ● The web application should |

| | |
|---|---|
| | have a monthly uptime of at least 99.5%.<br>● A backup of the latest version of the application should always be available. |
| Localizability | ● English, Norsk Bokmål, Norsk Nynorsk |
| Operation | ● The application will be administered by an admin with developer tools.<br>● Several moderators will moderate the website to maintain a safe environment standard. |
| Legal | ● The application must follow GDPR. |
| Portability | ● The application should be able to move to another hardware environment in case performance and network latency gets too bad. |

*(Table 9: Non-Functional Requirements)*

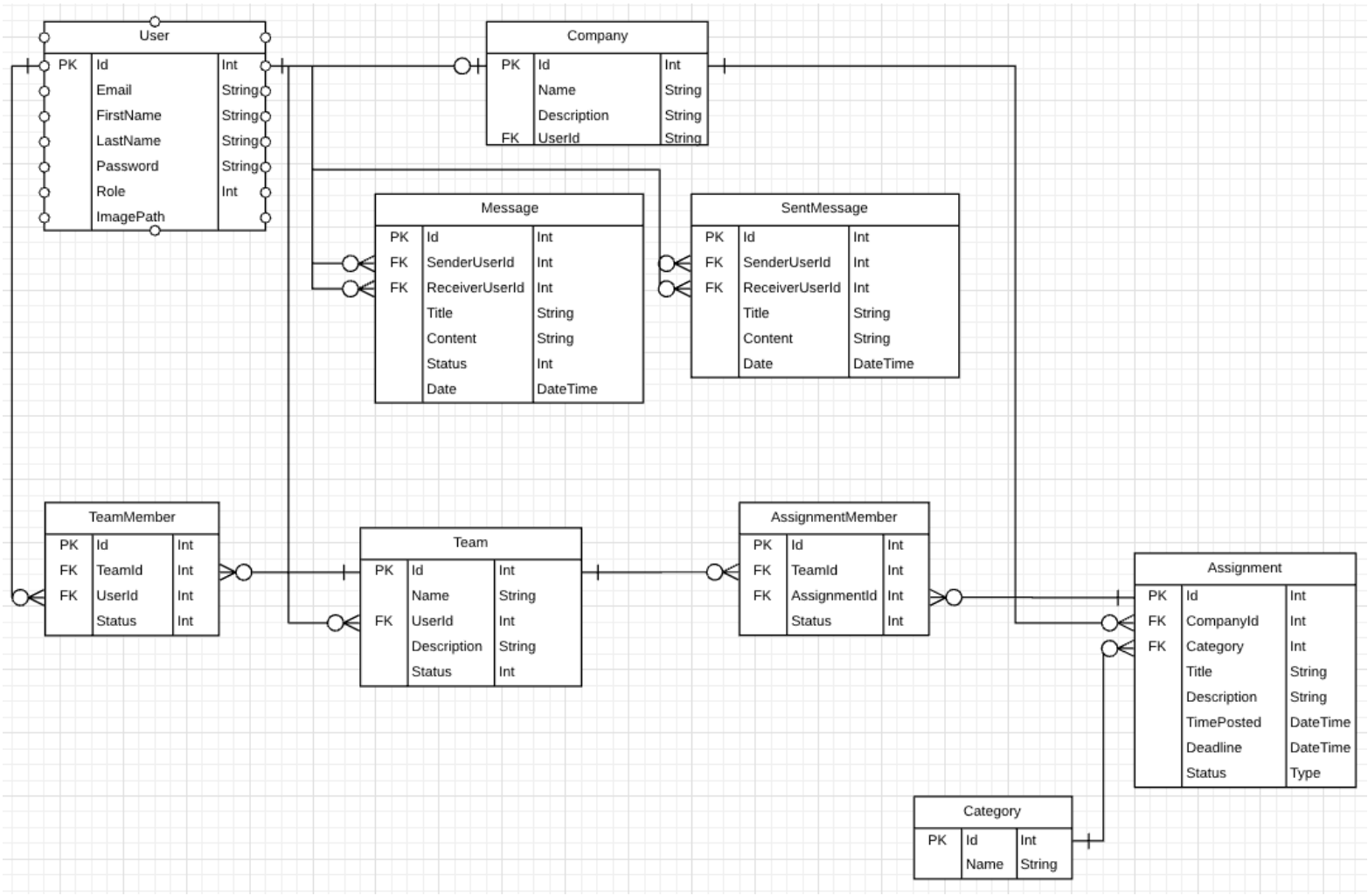
# 2.4 System Architecture

## 2.4.1 Main Component Diagram




*(Figure 1: Main Component Diagram)*

### 2.4.3 ER-Diagram




*(Figure 2: ER-Diagram)*

### 2.4.4 ER-Diagram Explanation

| Table Name | Explanation |
|---|---|
| User | Used to store user credentials. |
| Company | Used to store company names, linked to a user that is registered as a company. |
| Message | Used to store messages. |
| SentMessage | Used as a separate store for messages for the outbox. |
| TeamMember | Used to keep track of relationships between teams members, including invitations and requests. |
| Team | Used to store information about teams. |
| AssignmentMember | Used to keep track of relationships between teams and assignments, including invitations and requests. |
| Assignment | Used to store information about assignments. |
| Category | Used for categorizing assignments. |

*(Table 10: ER-Diagram Explanation)*

## 2.5 Risk Analysis

| | |
|---|---|
| **Scenario** | Competing sites |
| **Probability** | Low |
| **Severity** | Medium |
| **Action** | Differentiate platform from competitors. Better functionality and features. |

| | |
|---|---|
| **Scenario** | Low activity from users |
| **Probability** | Low |
| **Severity** | High |

| | |
|---|---|
| **Action** | .Introduce business & students to the platform through marketing. |

| | |
|---|---|
| **Scenario** | Fake users/trolls that spam the site |
| **Probability** | Medium |
| **Severity** | Low |
| **Action** | .Ban spamming users, actively admin the report feature |

| | |
|---|---|
| **Scenario** | SQL Injection |
| **Probability** | High |
| **Severity** | Very High |
| **Action** | Form validation & queries that are parameterized. |

| | |
|---|---|
| **Scenario** | XSS attack |
| **Probability** | High |
| **Severity** | Very High |
| **Action** | Sanitize form inputs, escaping tags |

| | |
|---|---|
| **Scenario** | MITM attack |
| **Probability** | Low |
| **Severity** | High |
| **Action** | Enforce SSL/TLS certificates with safe cyphers. HTTPS connection between server & client. |

| | |
|---|---|
| **Scenario** | Bots attack |

| | |
|---|---|
| **Probability** | Low |
| **Severity** | High |
| **Action** | Verify email, reCaptcha |

| | |
|---|---|
| **Scenario** | High load on server(s) |
| **Probability** | Low |
| **Severity** | Medium |
| **Action** | Stress test, upgrade hardware & bandwidth, lightweight code, more servers with load balancer |

| | |
|---|---|
| **Scenario** | DDoS attack |
| **Probability** | Medium |
| **Severity** | High |
| **Action** | Block IP Addresses, CloudFare |

*(Table 11: Risk Analysis)*

## 2.6 Target Audience

Our target audience are students and businesses. We need to offer multiple languages on our web application since there are international students and businesses that do not know Norwegian. Since the age span will be quite big, the application needs to be easy-to-use for everyone. Logical placement of features and names that describe the feature well.

### 2.6.1 Personas

| Ola, 20 | **Persona** |
|---|---|
| **Demographics** | Single, IT Student |
| **Interests** | Interested in computers, programming, games and likes to watch movies. |
| **Computer competence** | Uses computers daily for school work, social media, Netflix and personal programming projects. |
| **Wants and needs** | I Want to find a team. Needs to complete assignments to further his knowledge and make connections. |
| **Usage scenario** | Uses the site to look for other teams with IT education. Join a team and they apply for an assignment. Business reaches out through the message system on the web application. They exchange messages before the business accepts their application for the assignment. |

| Geir, 42 | **Persona** |
|---|---|
| **Demographics** | Married, CEO |
| **Interests** | Interested in soccer, history, whiskey. |
| **Computer competence** | Uses computer daily for work and at home to read and send emails. |
| **Wants and needs** | Wants an assignment completed for his business. Needs employees with certain skills. |
| **Usage scenario** | Creates assignment, goes through applications and contacts teams he is interested in, selects team after interview through message system. After the assignment is completed, contact a team member to employ. |

| Victoria, 23 | **Persona** |
|---|---|
| **Demographics** | Single, International Economics student |
| **Interests** | Makeup, Netflix, Horses |
| **Computer competence** | Uses computer daily in studies, social media & streaming. |
| **Wants and needs** | Wants to find team members for her group to cooperate with. |
| **Usage scenario** | Creates a group and adds all of the current members. Lists the group as “looking for members”. Reviews applications, messages applicants and accepts the chosen applicant’s application. |

*(Table 12: Personas)*

## 2.7 Privacy & GDPR

- Demand consent before storing any sensitive information about the user.
- Cookie warning needs to be in place
- The user has the right to get any data about them deleted if approval is revoked or they are no longer active on the application.
- The information we gather, must be made clear
- The user has the right to correct or update information stored on themselves
- Users can demand that the information about them is not used for any other purposes other than being stored on the site to use the application.
- The user can also stop an attempt where their information was to be used for other purposes. This has no exceptions and needs to be communicated clearly between the parties that it is understood.
- We must warn our users of a breach within 72 hours of discovering the breach.
- To ensure privacy with the information stored in our database we will use:
    - Hash Passwords
    - Validation of inputs
    - SSL/SSH
    - Verify data stream (if encrypted or not)
    - Restrict access based on user (developer, admin, regular user etc)
    - Use prepared statements to prevent SQL Injection

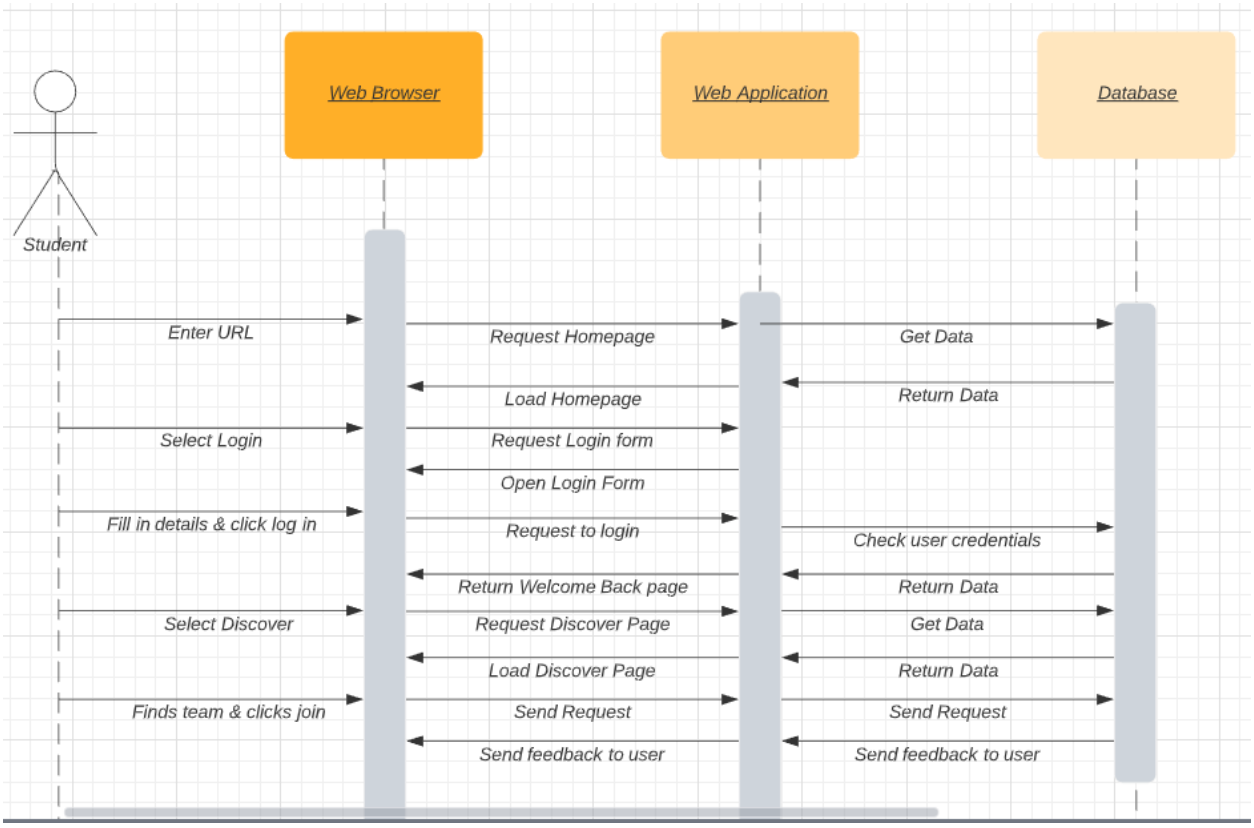
# 3.0 Design

## 3.1 User Groups

The main user groups for this application are the students at USN, USN Digital who will be running the application and the businesses and organizations posting assignments they want done. The last potential user group are other universities and university colleges.

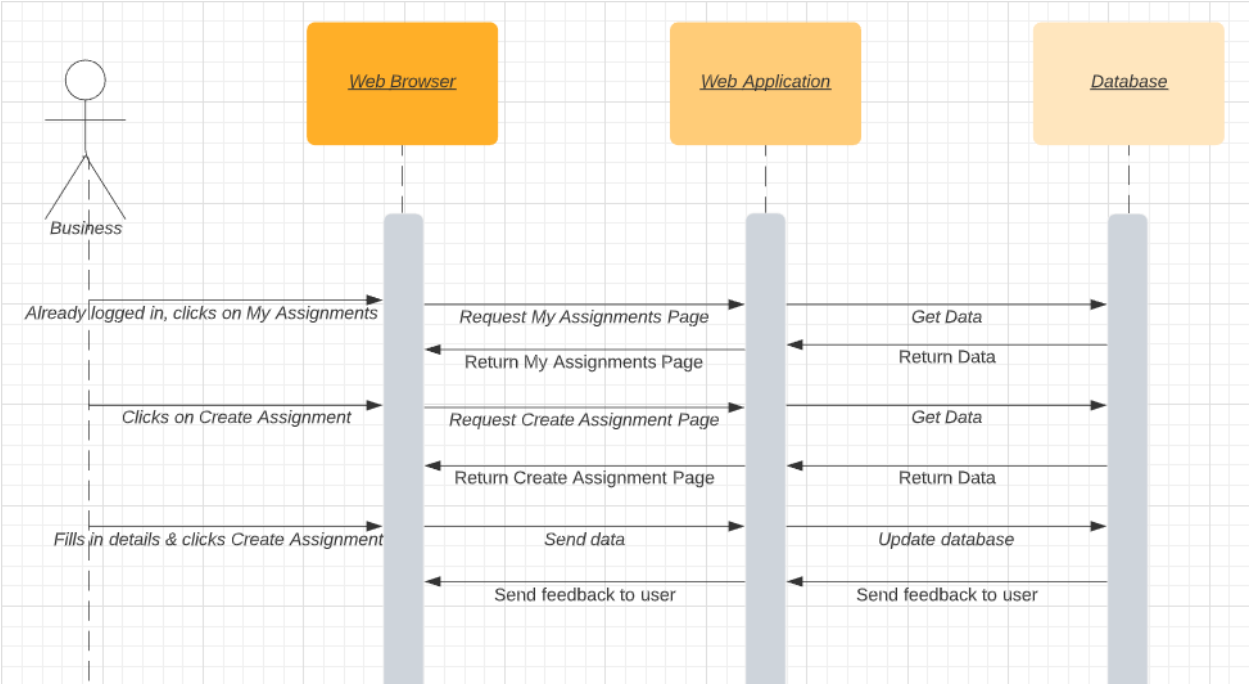
## 3.2 Sequence Diagram

### 3.2.1 Student login & join team




*(Figure 3: Student Sequence Diagram)*

### 3.2.2 Business create assignment




*(Figure 4: Business Sequence Diagram)*

## 3.3 Quality Control

### 3.3.1 “In-House” Testing

We did several rounds of testing in between ourselves to find problems with functions by letting someone else test it. Most of the tests were done towards the last couple of months of development, but we did some testing most of the way throughout development.

### 3.3.2 User Tests

We were not able to get user testing done because of the quarantine, but what we would have done is set up one or two laptops with the application and then found some students to give it a test run. There would have been a list of tasks for them to do as well as a portion for them to click around to see what they think and a form for them to fill out with how easy/hard they found the tasks to be, as well as a section for them to fill in with their own commentary about the application.

## 3.4 Chosen Technology

### 3.4.1 ReactJS

ReactJS or React is an open-source JavaScript library that is maintained by Facebook and a community of individual developers and companies. It makes it easy to build nice looking modular single page or mobile applications. It has VirtualDOM/DOM (Document Object Model) manipulation, this allows React to create an in-memory data-structure cache that computes the differences, and then updates the browser's displayed DOM in a quick and efficient manner. This allows the content to be dynamically changed, but this makes it dependent on additional libraries for state management and routing. The upside to this is that it allows for the use of whatever libraries the developers prefer to use to handle these tasks.

### 3.4.2 .NET Core

We chose to use .NET Core to create our back-end. .NET Core is a free open-source software. We have used the regular .NET framework before, but decided to go for .NET Core this time since it is the successor to .NET Framework and will therefore have support for longer. It is also cross-platform, uses NuGet packages, and it fully supports C#. In the future the application might be migrated from .NET Core to .NET 5, as .NET 5 will be the only .NET going forward after its release.

### 3.4.3 TypeScript

TypeScript is an open-source programming language that is developed and maintained by Microsoft. It is a superset of JavaScript that adds optional static typing to the language. TypeScrip works very well with React, as React uses either JSX (JavaScript XML) or JavaScript, and all JavaScript are valid TypeScript programs. We chose to use TypeScript because it can be seamlessly used with React, it also allows for seamless use of JavaScript code and libraries. As everyone has experience with JavaScript, it made sense for us to use TypeScript for our application.

### 3.4.4 MySQL

MySQL is one of the most widely used relationship databases and is both free and open-source. Since everyone learned to use MySQL databases in the database courses we had the second and third semester, it felt natural for us to use it here as well as everyone has experience with it while it also works with Azure. MySQL also works across all platforms and is easily maintained.

### 3.4.5 Azure

We are using Microsoft Azure to host our web application. We are using Azure because it is fast, stable and reliable. We also get $100 in start-up capital via the Azure for Students program. This is convenient as there is a cost to using MySQL databases on Azure. Since we are using Microsoft Visual Studio, we don't have to worry about incompatibility.

### 3.4.6 Axios

Axios is an alternative to the Fetch API, and is used to handle http requests and responses. Compared to the Fetch API it has more features and better error handling.

### 3.4.7 MobX

To manage the state of all the different controls in our web application such as forms and buttons we will be using MobX. The way this works is that React provides the tools that render the UI and MobX is used to synchronize the application state with the React components.

### 3.4.8 Visual Studio

We chose Visual Studio as our IDE because it's an IDE everyone in the group has used in a previous course where we also made an application with .NET. This made it an easy choice since everyone was familiar with it.
As Microsoft are the developers of both Visual Studio and ASP.NET, they are also sure to work well with each other.

### 3.4.9 GitHub

We have chosen to use GitHub for revision control and to share progress on the project. It can easily roll back if a mistake is made. It allows us to work together and have all our project files updated. If there is a conflict where two people have made changes on the same file, you can select if you want to keep the changes from the other person or override them with yours. In the end when the project is finished, the different branches can be merged for the finished project.

## 3.5 Safety Measures

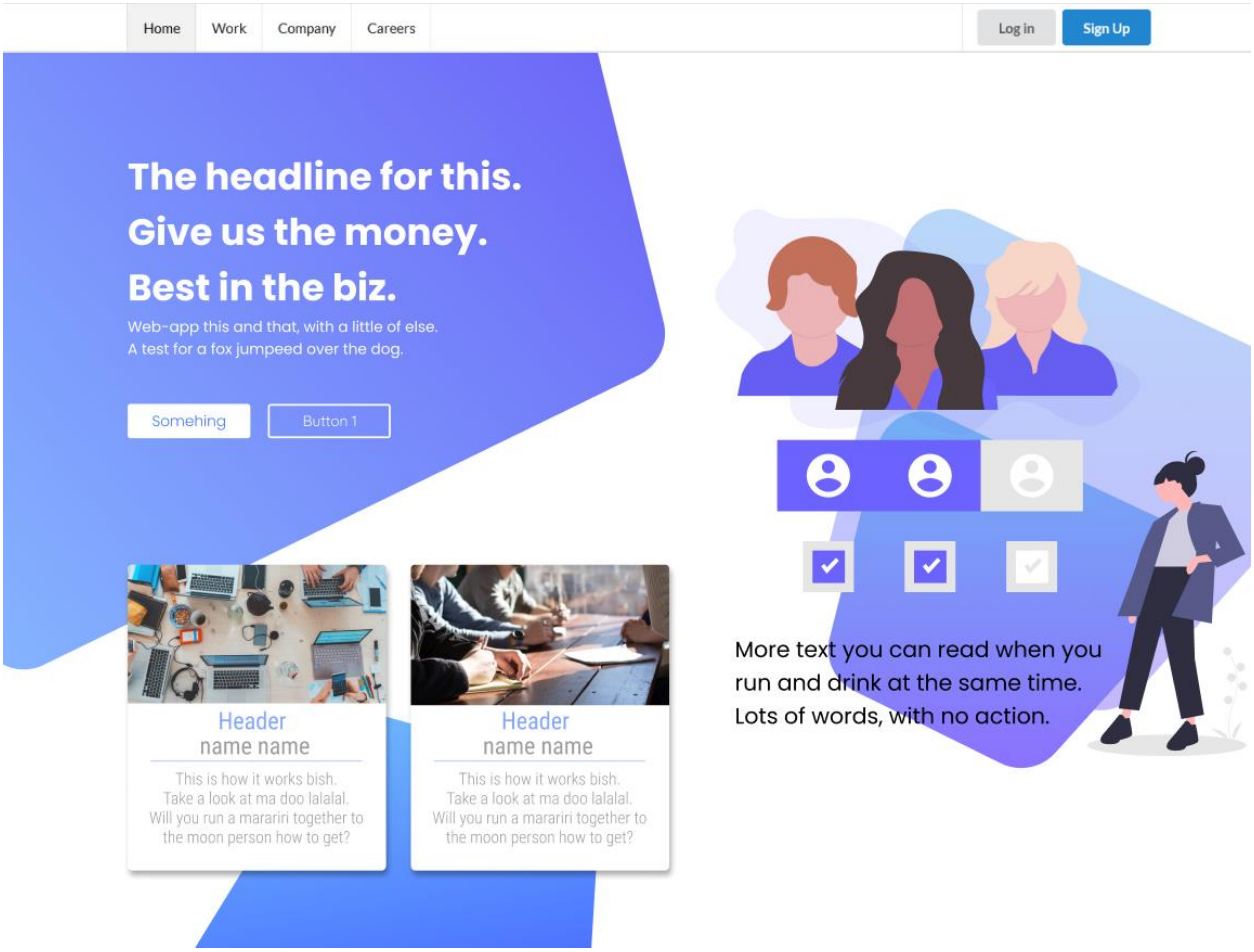
To keep the application safe from intruders certain safety measures need to be in place:

| |
|---|
| It is important to keep the site updated at all times. That means keeping plugins updated and if new loop-holes or bugs are discovered, they need to be patched. |
| Unnecessary code/plugins/themes etc. needs to be removed if not in use. If left, they can become a weakness later on that can allow intruders in. |
| We will also use captcha to verify users legitimacy along with two-factor authentication to ensure the user is who they say they are. Min ID and feide are options to consider integrating later on to even further verify the user credentials. |
| Passwords will be required to meet certain requirements to ensure its strength. |
| Backup of the site is essential in-case of malicious attack or sabotage. That way you can always roll back to the last version which was not infected and still have all the data you might have lost. |
| We will use SSL/SSH, HTTPS, CORS Standard and JSON Web Tokens to ensure our site has the protection it needs and cannot be taken advantage of easily. Along with encrypting passwords and any other sensitive information stored to protect users. |

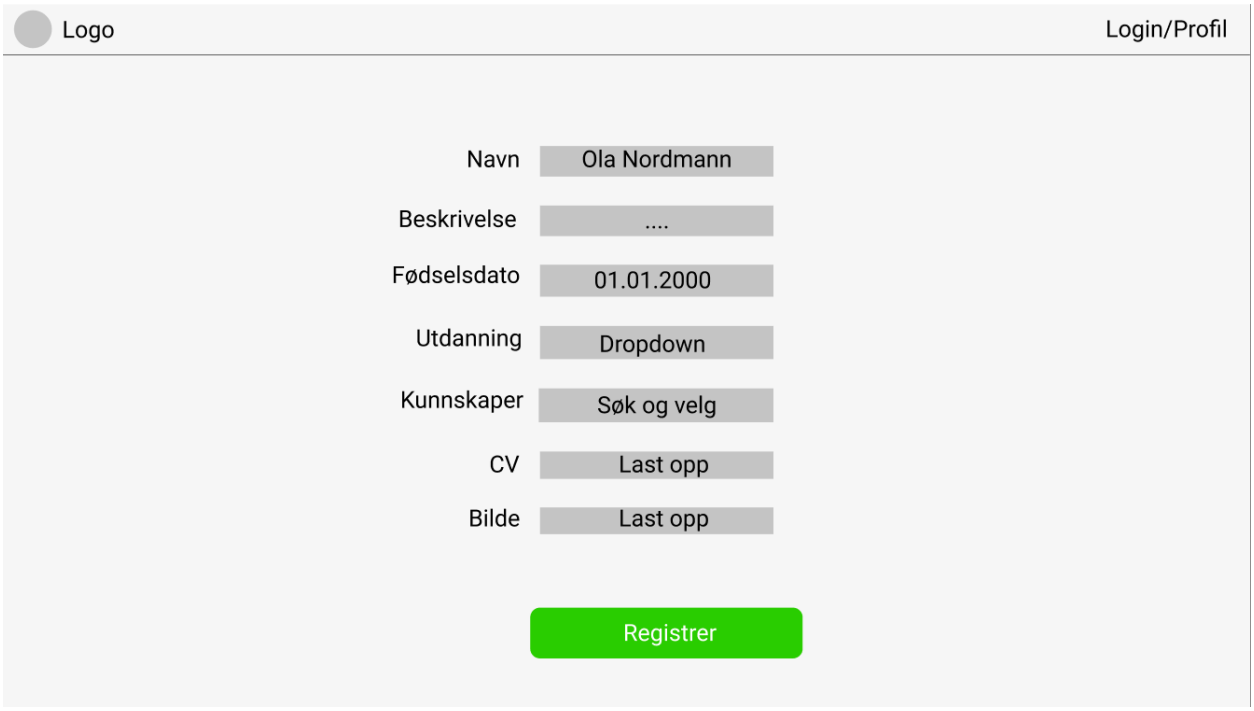
*(Table 13: Safety Measures)*

## 3.6 Wireframes

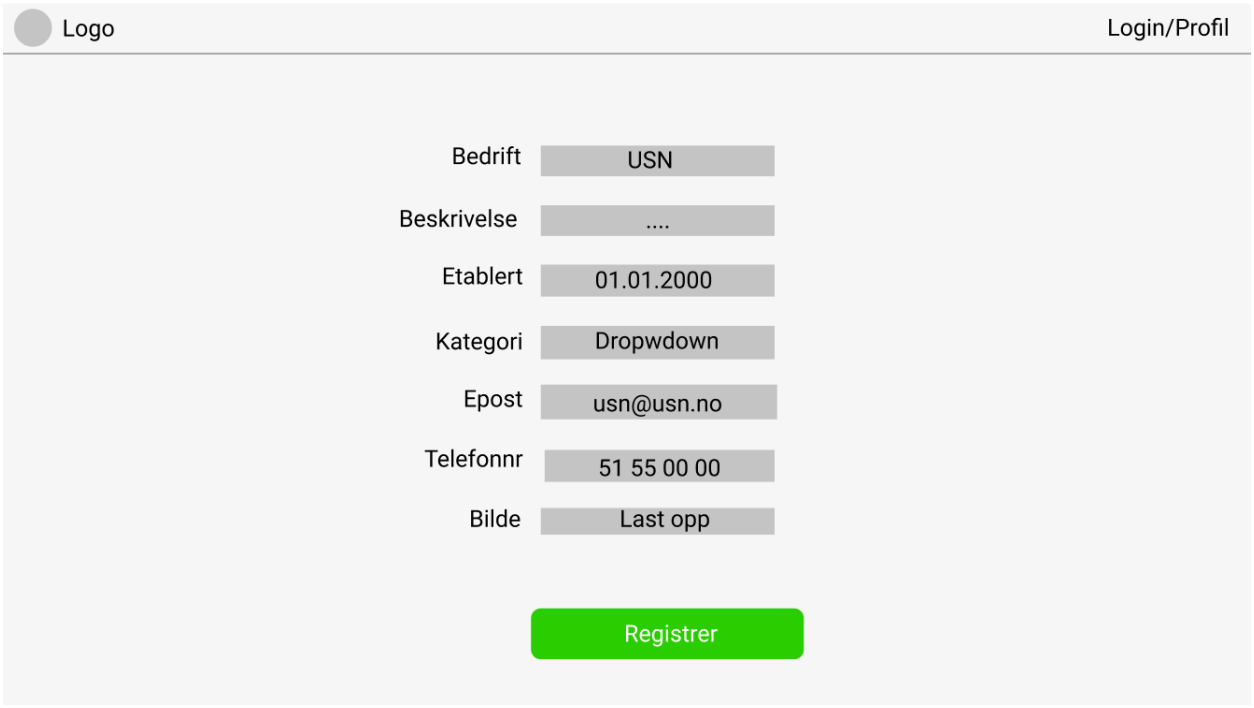
These were the wireframes we made in the beginning and started building our application off of, the design naturally changed a bit as we got further into development and figured out what worked and what didn't. The design did not turn out vastly different despite us taking Semantic UI in use, it stayed close to the original plans for the application.



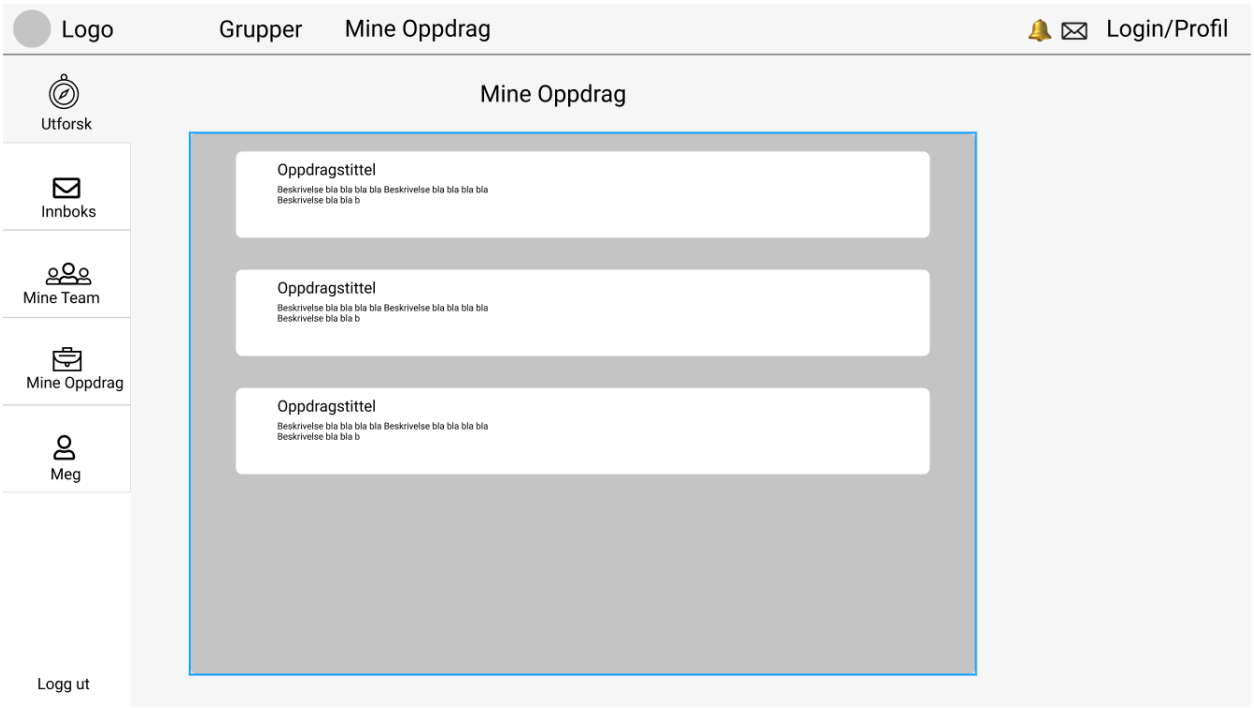

*(Figure 5: Wireframe Frontpage)*



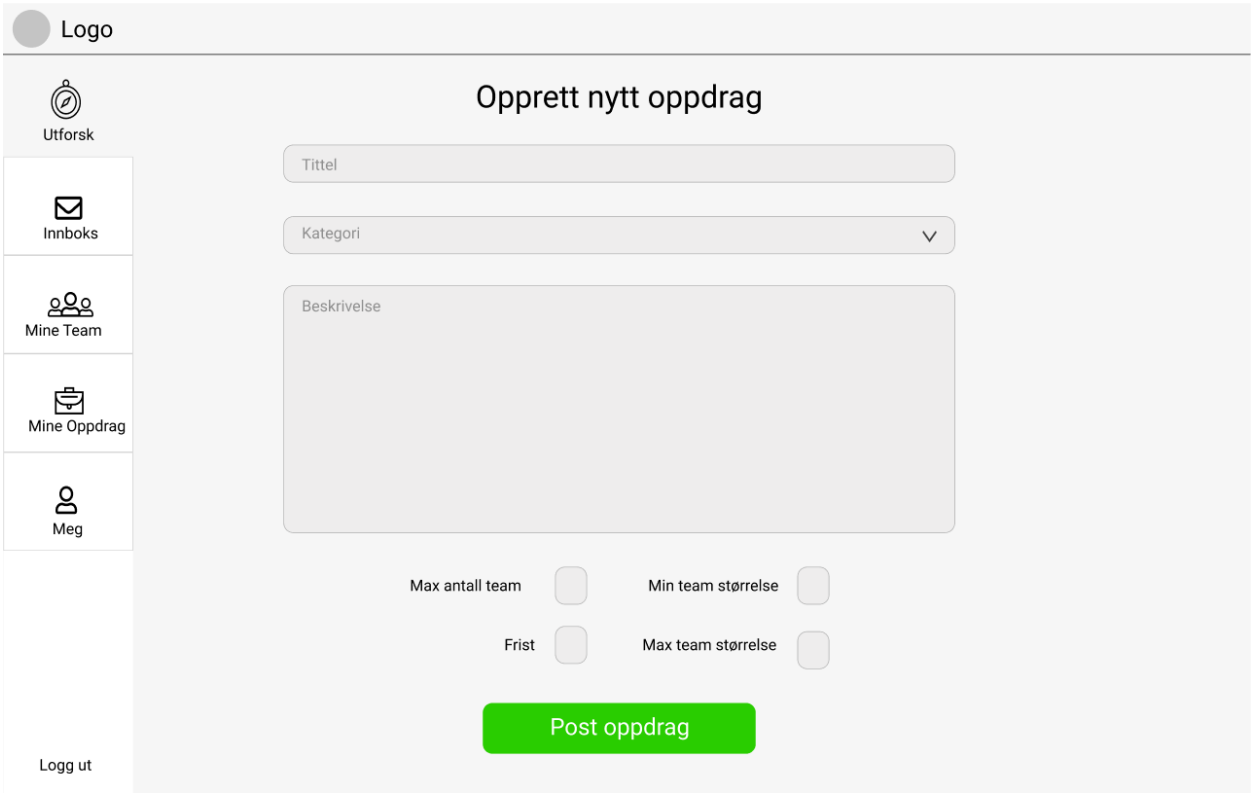

*(Figure 6: Wireframe Student Profile)*



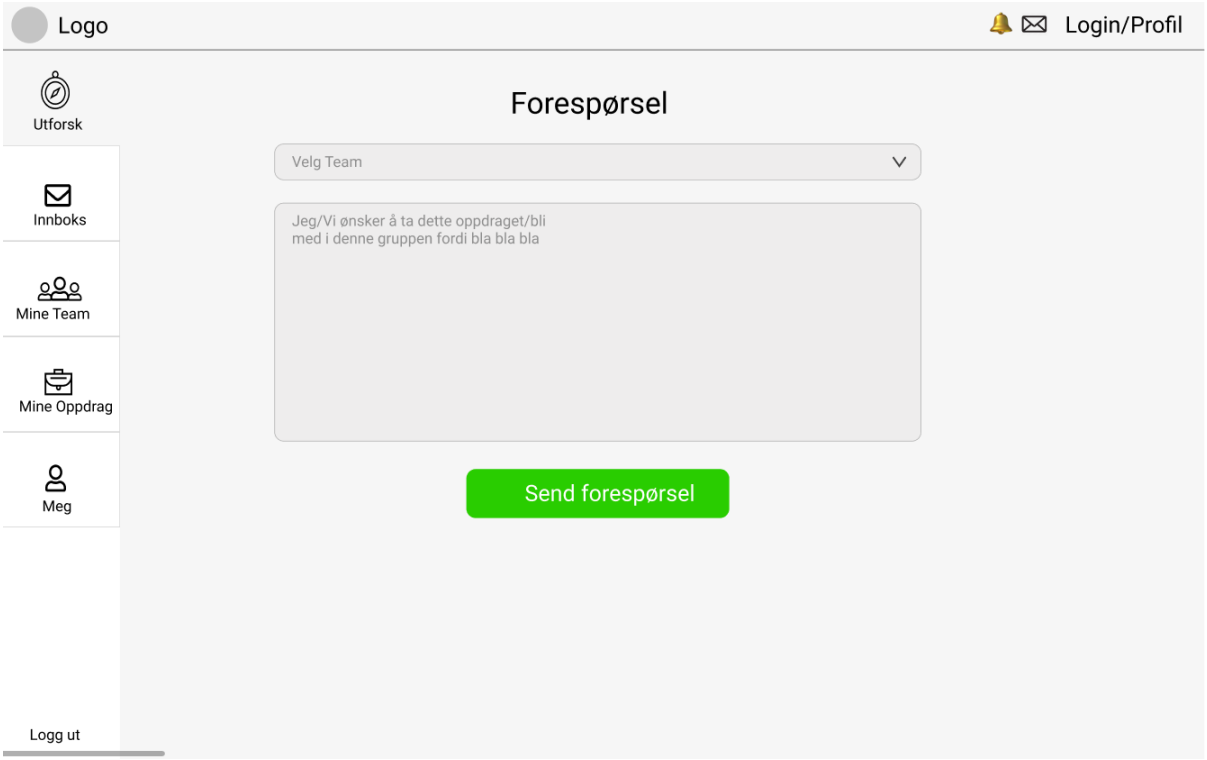

*(Figure 7: Wireframe Business Profile)*



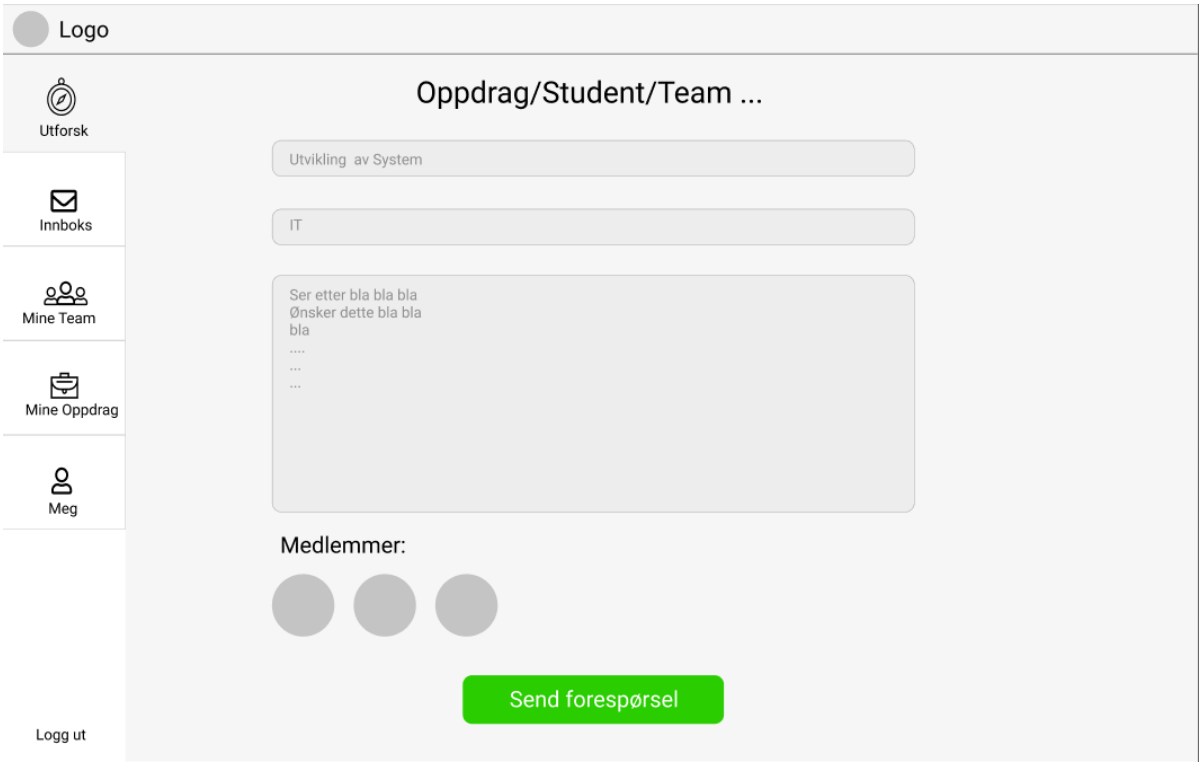

*(Figure 8: Wireframe My Assignments)*



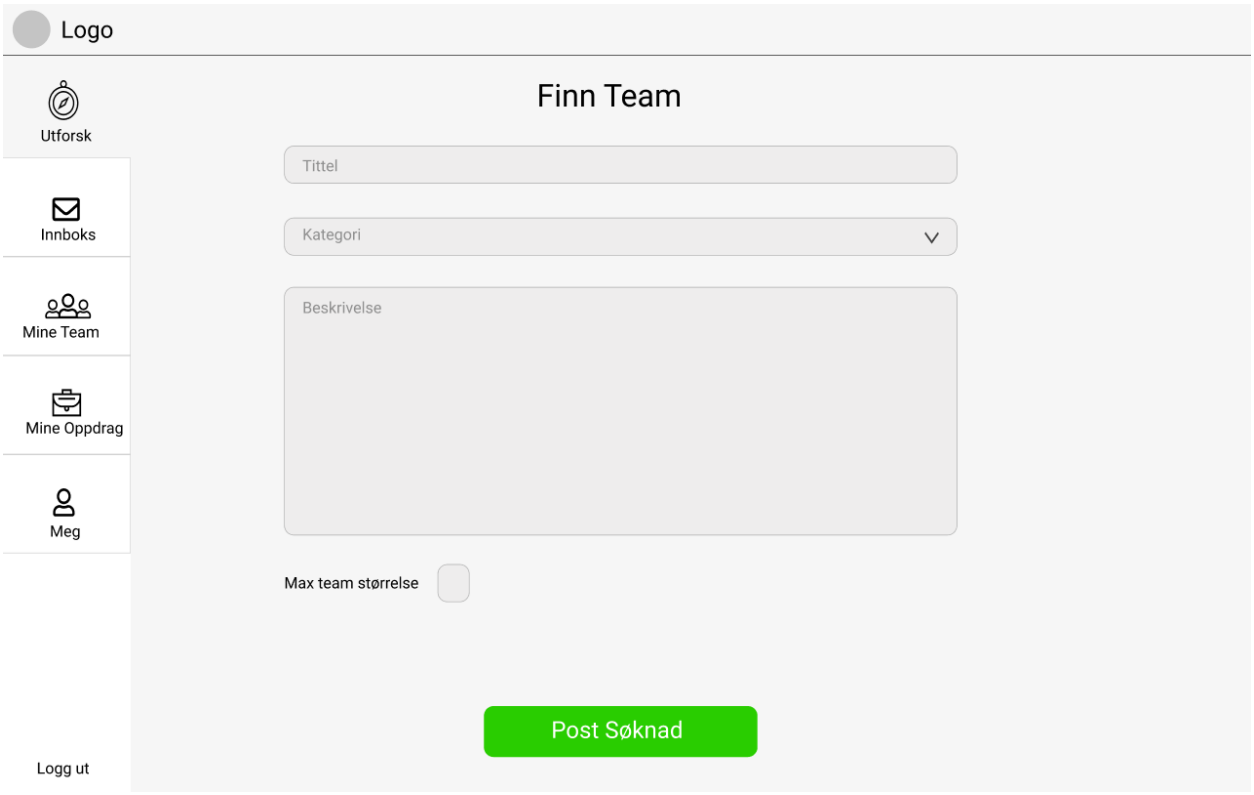

*(Figure 9: Wireframe Create Assignment)*



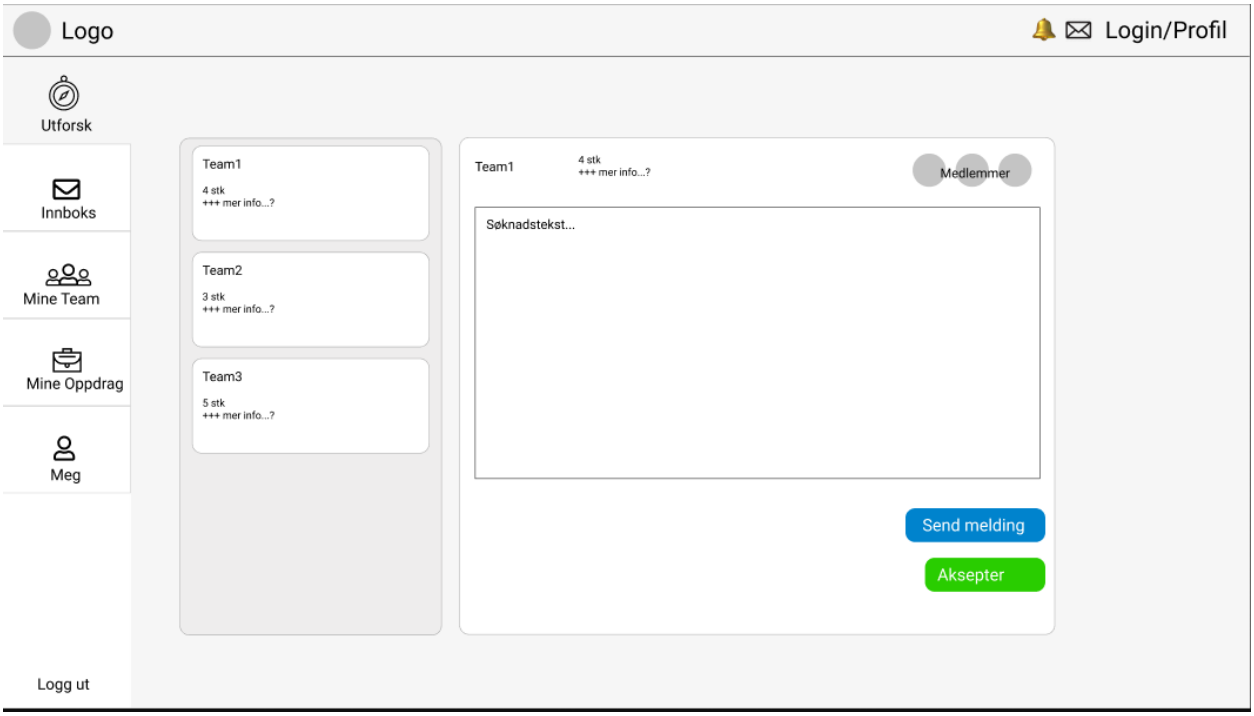

*(Figure 10: Wireframe Requests)*



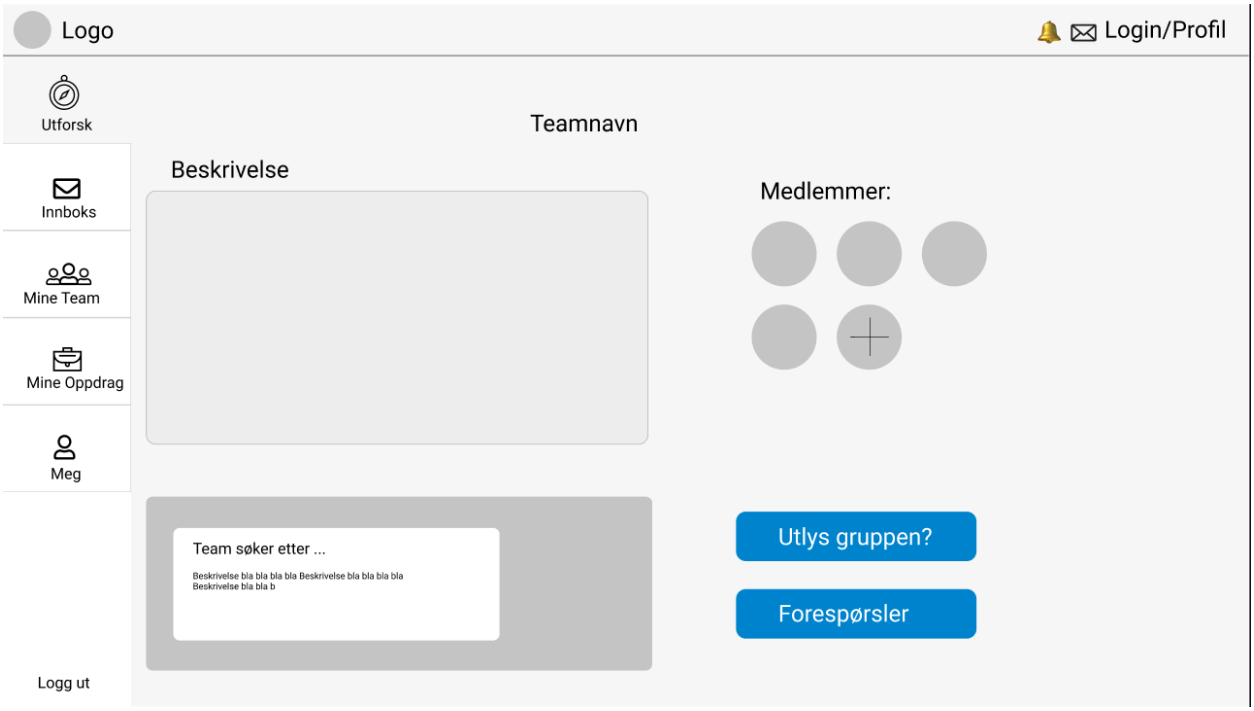

*(Figure 11: Wireframe Team Request)*




*(Figure 12: Wireframe Find Team)*




*(Figure 13: Wireframe Team Accept Requests)*




*(Figure 14: Wireframe Team Page)*

## 3.7 Communication & Tools

In big projects, communication is key. That’s why it was important for us to find proper tools to communicate and cooperate, especially during the last few months when all work had to be done from home and we couldn't meet up. We used Facebook for short messages and on the fly communication, while all meetings and from home work sessions were done over Discord. Most of the written work as well as planning was done on two separate google documents.

### 3.7.1 Discord

We chose to use Discord as our main communication and file sharing tool. Discord is a freeware VoIP application that is used by millions every day. It allows for both voice calls, screen sharing and file sharing, as well as leaving written messages to each other

### 3.7.2 Google Docs

For easy cooperative work on the report and to share ideas for the application, we chose to use Google’s Google Docs. Google Docs is a free to use cloud based service made by Google that is easy to use, automatically saves the work and can be converted to word files or PDFs with little effort.This allows multiple people to work on the same document at the same time and is safely saved in the cloud so we are not dependent on a single PC to keep it safe.

### 3.7.3 Facebook Messenger

Facebook Messenger is an easy to use instant messaging application that most people use to some degree every day. This made it an easy choice to use as a communication platform for us as everyone uses it often and has easy access to it.

# 4.0 Project Execution

This part of the report is dedicated to explaining how we have worked with the project. We chose to follow the agile Kanban method as everyone had used this before in other projects and it suited our group better than Scrum, who was the other option we were considering.

## 4.1 Methodology

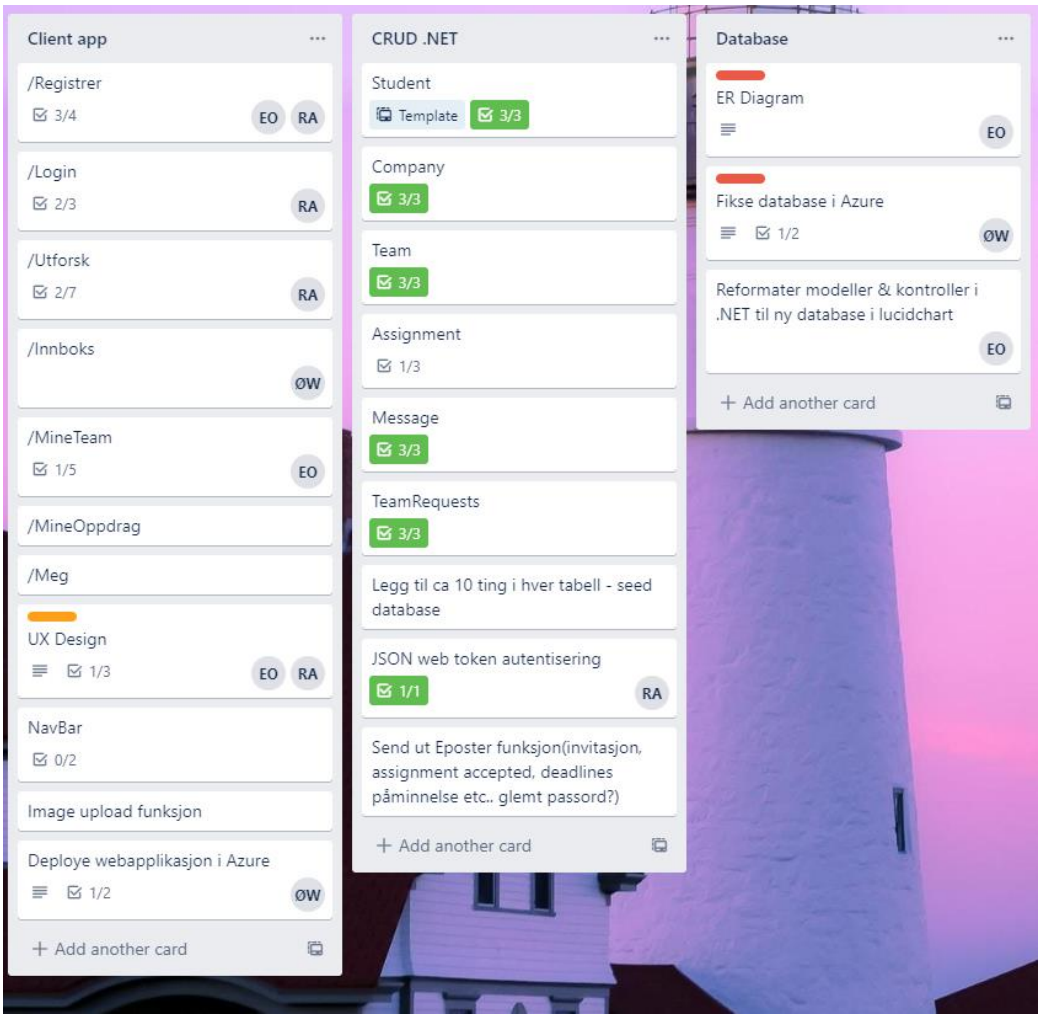
We chose to use the agile method known as Kanban. Kanban is Japanese for “visual sign” or “card”.  The Kanban works well by breaking bigger assignments down to smaller parts, this way we have more control over individual parts of different functions and can optimize these better.

## 4.2 Project Management

The Kanban method works by first creating a kanban-board with all the known tasks that need to be done, this can either a physical or digital board with post-it notes or cards. These cards should only contain one assignment, to limit the work-in-progress (WIP).
On the board there should be three sections, To-do, WIP and Complete, this is to make it easy to easy what is left to do, what is done and what is being worked on by others. This allows the group to be more efficient during development, but at the cost of predictability as it becomes harder to predict when certain functions are finished. Another great thing about Kanban is that it is very open for change, so new ideas and improvements can be added to the board at any time, just make sure that the board doesn’t get over complicated and cluttered by adding to many ideas that might not have been needed to begin with.

### 4.2.1 Management Tool

We chose to use an online board called Trello as our Kanban board. Trello allows for easy management of cards. It is easy to create new cards with a simple click, and the cards can be dragged around as needed.




*(Figure 15: Trello)*

We decided to sort the cards into a few piles for the different parts of the application, as well as one for the report itself. This allowed us to have even greater control over what had to be done where.

# 5.0 Reflecting

This part of the report is dedicated to looking at how our work went, what we did right and what we could have done differently, as well as what function remains and ideas for further development of the application.

## 5.1 Workflow

At the start of the year we had several meetings between ourselves where we started by discussing what basic features we wanted the application to have and to get some basic ideas of what it might need. We proceeded with creating the wireframes and agreeing on some basic ideas on how the application should work and what it would look like
We then had a few meetings with USN Digital and our supervisor and discussed our ideas with them and got a sense of what USN Digital wanted as well. After this we continued to refine the ideas for a bit as we started working on the basic CRUD functions and getting a local database up and running

After we got the database working, we managed to finish basic send & retrieve requests with .NET and React. While doing this we also started the work on the report, where we made a basic template with all the sections we knew we would need. As we worked on the application we made sure to also work on the parts of the report we could along the way, some of the parts needing to be written closer to the end as we had to have a more finished product to get an accurate report.

We continued to have meetings every week or every other week depending on what parts of the application was being worked on until the pandemic when progress halted for a bit.

The pandemic made working together troublesome as we couldn not meet up as easily to work on the application together, this slowed us down as we were not

prepared for this, added with different exams, this slowed down the progress, but it sped back up the last month or so of development.

At the end of development we split the group into two, one part to finalize the application and one to finalize the report. We chose to keep three people on the application and have two people focus on the report.

## 5.2 Experiences

We had a good start with a lot of meetings that allowed us to get a common vision on what we wanted the application to be, this was definitely a plus, but when the pandemic struck, we should have been a lot more active in trying to get together more proper meetings and not only messenger chat on Facebook, as well as more meetings with our supervisor and possibly USN Digital.

During development we switched over to using Semantic UI, this was a change that in hindsight just made a lot of sense and made it easier for us to achieve certain goals we had with the application of how it should work.
There was another drop in productivity when the exam period began as well, we should have tried to keep it at a somewhat higher level, this would likely have been done had we set up a few more meetings than we did.

The choice of using Discord as our main communication platform payed off when the pandemic struck, after we got our productivity up again, as it allowed for us to share what we were doing on our screen in realtime to brainstorm and solve problems on how to fix or create functions we were working on.

The decision to split our group in two at the end to focus on the application and the report separate allowed to us to get a lot more work done on the report than we would have had otherwise, we feel like this was a good choice for us, but it did put a bit more strain on the development than we would have liked. It might

have been better to have one or both of the people working on the report, partially working the application as well.

## 5.3 Future Development

| Name | Explanation |
|---|---|
| Notifications | A notification system to allow users to get pinged when they receive a message, a request or a request has been accepted etc. |
| Report system | A system to allow users to report other users for bad and/or malicious behaviour |
| Ban/blacklist | A system to allow admins to easily deactivate, ban or blacklist users who are found to have broken rules or acted in a purposefully malicious way |
| Multiple Language support | There are plans to translate the application to allow it to be shown in multiple languages as Norwegian (bokmål), Norwegian (Nynorsk) and English. |
| Moderator Tools | Tools to allow the admins and moderators to edit user information if needed and use the above mentioned ban tools. |
| Account Recovery | Allowing users themselves to recover accounts if they forget the password |
| Load Balancer | Azure has a built in load balancer as a paid option that can be worked into the application. |
| SSL/TLS-certificate | Azure offers this and it shouldn't be too much trouble to apply it to the application |
| Brute-force protection | To prevent unauthorized logins. |
| Team Leader Tools | Adding more tools for the team leaders to use to manage teams and workflow. |

*(Table 14: Future Development)*

# 6.0 Results

Our main goal is to create a platform that has all the essential features for students and businesses to meet and make connections. We want to deliver a platform that USN Digital can build upon. However we also wanted to include other features that made the platform more easy to use and more functional, so we set a high goal for ourselves by trying to get as many functions in as possible.

The end result is a web application that has the functions needed to handle a job like this, like allowing for multiple types of users i.e student users and business users. The different users have different options in the application, while student and business users have a lot of similarities, there are some distinct differences.

The students are able to create teams to take on assignments, message other users. They are also able to search for assignments and teams to join. There will be a team leader who creates the team, but everyone on the team will have the ability to accept the potential applications. After joining or creating a team, the student users have access to the team page that shows information about the team, such as team members and what assignment they are working on. They also have access to an assignment/team overview that allows them to see what assignments they are working on, what teams they are in and what outgoing and incoming requests they might have in the form of requests to join a team, or others requesting to join their team.

While business users have the ability to create assignments instead of teams. These assignments will be posted publicly so that the different teams can apply for them. They will also have access to an overview that allows them to see what assignments they have posted

Both user types have access to a personal profile, for the business user, this will allow for them to write the name of their business and write down basic

information. While the students will be able to add their name, other basic information when they sign up. These profiles are public so both students and businesses are able to access them on demand.

# 7.0 References


‘.NET Core’. In *Wikipedia*, 2 May 2020. https://en.wikipedia.org/w/index.php?title=.NET_Core&oldid=954353448.

Google Docs. ‘APP2000 Midterm Project Report’. Group 12, Accessed 24 May 2020. https://docs.google.com/document/d/1w6qp1Bx77CjSPavFHRxQhlvtmrGjD55g2OtAY-UiDTI/edit?usp=sharing&usp=embed_facebook.

Ayoola, Solomon . ‘Axios vs. Fetch?’. Pluralsight, 29 January 2020 https://www.pluralsight.com/guides/axios-vs-fetch.

‘Discord (Software)’. In *Wikipedia*, 29 April 2020. https://en.wikipedia.org/w/index.php?title=Discord_(software)&oldid=953787460.

‘Facebook’. In *Wikipedia*, 18 May 2020. https://en.wikipedia.org/w/index.php?title=Facebook&oldid=957360060.

‘Google Docs’. In *Wikipedia*, 1 April 2020. https://en.wikipedia.org/w/index.php?title=Google_Docs&oldid=948491824.

‘Kanban’. In *Wikipedia*, 27 February 2020. https://en.wikipedia.org/w/index.php?title=Kanban&oldid=942854074.

Chen, James. ‘Learn What a Stakeholder Is’. Investopedia. Accessed 7 May 2020. https://www.investopedia.com/terms/s/stakeholder.asp.

‘Microsoft Azure’. In *Wikipedia*, 6 May 2020. https://en.wikipedia.org/w/index.php?title=Microsoft_Azure&oldid=955287348

‘MobX’. Accessed 27 February 2020 https://mobx.js.org/README.html.

‘MySQL’. In *Wikipedia*, 4 May 2020. https://en.wikipedia.org/w/index.php?title=MySQL&oldid=954812399.

‘React (Web Framework)’. In *Wikipedia*, 9 April 2020. https://en.wikipedia.org/w/index.php?title=React_(web_framework)&oldid=949984059.

‘TypeScript’. In *Wikipedia*, 1 May 2020. https://en.wikipedia.org/w/index.php?title=TypeScript&oldid=954200125.

BusinessDictionary.com. ‘What Comes after Those Ellipses?’ Accessed 7 May 2020. http://www.businessdictionary.com/definition/stakeholder.html.

Smartsheet. ‘What’s the Difference? Agile vs Scrum vs Waterfall vs Kanban’. Accessed 7 May 2020. https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban.