

Eksamensprojekt

Said Masih Ashna (06/08/1998) og Erdal Erdem (21/04/2002)

Dat22C

31. maj 2023

Indholdsfortegnelse

| | |
|---|-----------|
| Indholdsfortegnelse | 1 |
| Introduktion | 3 |
| Problemstilling | 3 |
| Proaktiv Risikostrategi | 4 |
| SWOT-analyse | 4 |
| Strengths | 5 |
| Weaknesses | 6 |
| Opportunities | 6 |
| Threats | 6 |
| Risikoanalyse | 7 |
| Risiko Formindskelse af vores weaknesses | 8 |
| Risiko Formindskelse af vores threats | 9 |
| Aktionsplan | 10 |
| Risikoovervågning | 11 |
| Interessentanalyse | 13 |
| Feasibility study | 15 |
| Teknisk | 15 |
| Økonomisk | 15 |
| Juridisk | 16 |
| Tidsmæssig | 16 |
| Arbejdsprocessen | 17 |
| Daily stand-up møder | 17 |
| Sprint Planning | 17 |
| Sprint Review | 18 |
| Sprint Retrospective | 20 |
| Arbejdsvurdering | 20 |
| Svagheder og styrker | 20 |
| Forbedringer | 21 |
| Teknologivalg | 21 |
| Domænemodel og database | 21 |
| Hvad er firmaets håb med dette system? | 22 |
| User stories. | 23 |
| Sprint 1. | 23 |
| Sprint 2. | 24 |
| Konklusion af sprint | 24 |
| Structured Query Language | 25 |
| SpringBoot | 26 |
| Thymeleaf | 27 |
| Cascading Style Sheets | 28 |
| Teknologi | 29 |
| Produktionsmiljø | 29 |

| | |
|---|-----------|
| Udviklingsmiljø. | 30 |
| Software Design. | 30 |
| Arkitektur. | 30 |
| Klassediagram. | 31 |
| Design Overvejelser. | 32 |
| Enkelhed(Simplicity). | 32 |
| Høj sammenhæng(High Cohesion). | 33 |
| Lav kobling(Low Coupling). | 33 |
| Konsistens(Consistency). | 33 |
| Opdeling og modulopbygning (Decomposition and Modularity). | 34 |
| Navigations Flow mellem skærbilleder. | 34 |
| Log ind. | 34 |
| Menuen. | 35 |
| Projektsiden. | 36 |
| Designet bag vores program. | 36 |
| Hvem er vores brugere? | 37 |
| Hvad vil de gerne have? | 37 |
| Hvad problem løser vi for dem? | 37 |
| Hvordan reflekterer det her med jeres produkt mål? | 37 |
| Forståelse af vores brugere. | 37 |
| De gyldne regler indenfor UI design | 38 |
| Sekvensdiagrammer | 40 |
| Særlige forhold | 42 |
| Udvalgte kodeeksempler. | 43 |
| Tests. | 46 |
| UserRepository. | 46 |
| ProjectRepository. | 48 |
| ModuleRepository. | 48 |
| Konklusion | 49 |
| Litteraturliste | 50 |
| Bilag | 51 |
| Github repository link | 52 |
| https://github.com/MasihSaidAshna/EksamensProjekt | 52 |

Introduktion

Denne rapport gennemgår vores udviklingsproces gennem hele projektperioden. Vi vil gennemgå vores analyser og implementeringer af systemet. Formålet med vores system er at levere et projektstyringsprogram til Alpha-solutions, en IT- og e-handelsvirksomhed. Alpha-solutions ønsker et virtuelt og mere funktionelt projektstyringsprogram, der kan hjælpe dem med deres projektplanlægning. Under hele projektet har vi haft tæt samarbejde med Lenka, vores product owner (PO), som har repræsenteret kunden under eksamen.

Rapporten vil også beskrive, hvordan vores gruppe har håndteret de udfordringer, vi blev stillet over for og problemløsning til disse. Vi vil forklare vores begrundelser for valg af forskellige værktøjer og gennemgå vores tanker og overvejelser igennem hele projektperioden.

Problemformulering

Alpha-solution står over for en udfordring, da de mangler et digitalt værktøj til at understøtte deres planlægning af komplekse systemer for deres kunder. Vi har fået muligheden for at udvikle et projektstyringsprogram, der vil fungere som et centralt system for alle Alpha-solutions' fremtidige planer.

Vores program vil primært kunne håndtere to typer brugere:

Ansatte - Disse enkeltpersoner vil være involveret i forskellige projekter, som Alpha-solutions har til rådighed.

Managers/projektledere - Disse individer vil oprette og administrere projekter, som de ansatte kan tilslutte sig.

Formålet med vores program er at forbedre effektiviteten hos Alpha-solutions ved at tilbyde et centralt hub, hvor størstedelen af deres projektplanlægning kan foregå. Dette vil skabe en mere tilgængelig hverdag for alle ansatte og bidrage til en mere effektiv arbejdsproces.

Proaktiv Risikostrategi

Vi har valgt ikke at fokusere på reaktive risikostrategier, da den reaktive risikostrategi er kortsigtet og kun forholder sig til problemer, som danner sig undervejs i projektet. Derfor har vi valgt at benytte os af en proaktiv risikostrategi, da den danner sig et billede på lang sigt, som vil hjælpe projektet, hvis det skulle støde på nogle problemer.

Den proaktive risikostrategi består af fire hovedelementer, som øger chancen for et succesfuldt projekt. Det første hovedelement er at identificere alle de realistiske og potentielle risikomomenter for projektet, produktet (Projekt Kalkulationsværktøj) og basisorganisationen.

Derefter vil der laves en risikoanalyse. Risikoanalysen undersøger sandsynligheden for uvelkomne indtræffelser af elementer, som kunne have risiko for projektet som helhed, og hvad konsekvenserne vil være af disse risikomomenter.

Når risikoanalysen er blevet istandsat, og risiciene er blevet identificeret, vil tredje hovedelement blive igangsat. Denne indeholder en aktionsplan, hvis mål er at lave en plan for at undgå risiko-momenterne, som projektet kunne støde på.

Det fjerde og sidste hovedelement er en risikoovervågning, som tager udgangspunkt i at fokusere på eventuelle andre risikomomenter, der kan træde til, som ikke var taget højde for.

SWOT-analyse

¹

Udgangspunktet i planlægningen er nu, at hindre risiko-momenterne, dette opnås ved at isolere momenterne, der kan udvikle sig til en stor problematik gennem projektperioden. For at identificere disse problematikker skal vi benytte risikoanalysen og SWOT-analysen. SWOT-analysen identificerer projektets styrker og svagheder ved at isolere dem med fire hovedmål, projektets strengths, weaknesses, opportunities og threats.

¹ 2022-Zwisler, Projekt og analyseredskaber - (læst 06/05-2023)

| Positive faktorer i forhold til projektet | Negative faktorer i forhold til projektet |
|---|--|
| <p>Projektets interne styrker (S = Strength).</p> <p>S1: God kommunikation og let at aftale møder.</p> <p>S2: Vi er velkendte med hinandens evner, da vi har arbejdet sammen i andre projekter.</p> <p>S3: Arbejdsmoralen er god og motiveret.</p> | <p>Projektets interne svagheder (W = Weakness).</p> <p>W1: To-mands hold (kun et problem, hvis en af medlemmerne bliver syge).</p> <p>W2: Mangel på vejledning.</p> <p>W3: Ustabilt fremmøde til møder.</p> |
| <p>Muligheder i omverden (O = Opportunities)</p> <p>O1: Vi har muligheden for at vise vores styrker som en samarbejdende gruppe, hvilket kan forbedre vores ry og skabe gode relationer med andre interessenter.</p> <p>O2: Vi har muligheden for at levere et program, der ikke kun fungerer optimalt, men også har et flot design. Dette kan imponere vores kunde og øge deres tilfredshed med projektet.</p> <p>O3: Som nye datamatikere har vi muligheden for at vise vores færdigheder og ekspertise i praksis. Dette kan bidrage til at opbygge vores omdømme og skabe fremtidige muligheder for os som professionelle.</p> | <p>Trusler i omverdenen (T = Threats).</p> <p>T1: Andre producenter leverer et bedre produkt end os.</p> <p>T2: Omkostningerne af produktet er for høje.</p> <p>T3: Kunden (Alpha-Solutions) er ikke tilfreds.</p> |

Strengths

Projektet har et element som vi har valgt at isolere, da vi mener, at de vil have stor betydning for resten af projektet og kvaliteten af produktet. God kommunikation er en vigtig og betydningsfuld kvalitet at have i et projekt, da det giver et overblik over udviklingsprocessen i projektperioden. Vores projektgruppe mener, at vi har denne kvalitet i vores arsenal, da vi tidligere har haft succesfulde projekter, der blev afsluttet til tiden med den forventede kvalitet, som udbyderen har efterspurgt.

Det at gruppen også ikke inkluderer mange andre faktorer, fordi gruppen består af to personer, ekskluderer mange små risikomomenter, som kan have stor betydning for det endelige produkt.

Weaknesses

Projektets svageste element, som vores gruppe mener vil udgøre den største udfordring for os, er, at vores gruppe kun består af to projektmedlemmer og mangler vejledning.

I det øvrige afsnit konkluderede vi, at en af vores styrker som gruppe er, at vi er en lille projektgruppe på to medlemmer, som har arbejdet tæt sammen før. Dog er der også en risiko forbundet med vores størrelse, da hvis et af vores medlemmer bliver syge, vil det reducere vores produktionskapacitet med 50%, hvilket vil have en betydelig indvirkning på det endelige produkt. Derfor har vi også valgt at prioritere vores medlemmers sundhed for at undgå udmattelse eller stress på grund af fejl i vores program eller rapport.

Vejledning er også en udfordring for vores gruppe, da vi tidligere har haft svært ved at bede om hjælp og kan være stædige. Derfor har vi valgt at holde hinanden ansvarlige og opmuntre til at bede om hjælp, når det er nødvendigt.

Opportunities

Vi anser denne projektopgave som en glimrende mulighed for os til at demonstrere vores evner som nye datamatikere samt som en velfungerende gruppe. Vi ønsker at vise, hvad vi er i stand til både i form af kodning og i skriftlig form gennem en veldokumenteret rapport. Denne opgave giver os en chance for at håndtere udfordringen med fokus på agil projektledelse, hvilket vil være afgørende for vores fremtid og uddannelsesforløb. Vi er overbeviste om, at dette projekt vil give os mulighed for at demonstrere vores potentiale på en imponerende måde.

Threats

Gruppen har identificeret flere trusler, som vi er særligt fokuserede på. Vi er bekymrede for produktets kvalitet i forhold til forventningerne fra Alpha-Solutions og om vores program lever op til branchens standarder i forhold til andre virksomheder. For at imødekomme disse bekymringer har vi besluttet at

lægge stor vægt på kvalitet og sikre, at vi arbejder tæt sammen med produktansvarlig (PO) for at udvikle det bedst mulige program inden for vores evner og kompetencer. Vi vil opretholde klar kommunikation mellem os og PO og udnytte så meget vejledning som muligt for at sikre succes.

2

Risikoanalyse

Risiko Momenterne er nu opstillet på baggrund af benyttelsen af SWOT-analysen. Derfor kan vi anvende det andet element i den proaktive strategi, der tager udgangspunkt i, hvor katastrofale de øvrige risikomomenter nævnt i SWOT-analysen er for det endelige projekt. Dette beregnes ved hjælp af en simpel formel, der multiplicerer sandsynligheden (S) med konsekvensen (K). Resultatet giver os en risikofaktor, der kan variere mellem 1 og 25. Hvis risikofaktoren er lav, betyder det, at sandsynligheden og konsekvensen for det specifikke risikomoment ikke er høj og dermed ikke vil have en ekstremt stor betydning for projektet, hvis det skulle ske. På den anden side af skalaen betyder en høj risikofaktor, at risikomomentet vil have stor betydning for projektet og kan forhindre os i at levere et fuldendt produkt til vores kunder, Alpha-solutions.

| Risiko hentet fra SWOT analysen | Sandsynlighed (S) (vurdering fra 1-5) | Konsekvens (K) (Vurdering fra 1-5) | Risikofaktor (K * S) | Reducer Sandsynlighed | Reducer Konsekvens |
|--|---------------------------------------|------------------------------------|----------------------|---|---|
| W1 - To-mands hold (kun et problem, hvis en af medlemmerne er syge). | 1 | 5 | 5 | Holder os sunde og raske sikre at få søvn og ikke overarbejde | Ved overholde dette minimere det risikoen for at vi bliver stresset eller syge. |
| W2 - Mangel på vejledning. | 1 | 3 | 6 | Spørger vejlederne mere og ikke være stædig. | Vi vil bruge mindre tid på små problemer og få lavet mere. |
| W3 - Mødestabil. | 2 | 3 | 6 | Få den rigtige mængde søvn og disciplin | mere produktiv og bedre produkt |
| T1 - Andre producenter levere et bedre produkt end os. | 3 | 5 | 15 | Arbejde fokuseret og gør brug af vores redskaber (vejledning osv) | For lavet et bedre produkt som kunden og vi selv være stolte af. |

² 2022-Zwisler, Projekt og analyseredskaber - (læst 07/05-2023)

| | | | | | |
|--|---|---|----|---|--|
| T2 - Omkostninger af produktet er for højt. | 1 | 5 | 10 | Benytte af skolen har givet os. | Vi får næsten/ingen omkostninger |
| T3 - Kunden(Alpha- Solutions) er ikke tilfredse | 2 | 5 | 10 | Have en fokuseret struktur når der er PO møder og forstår hvad kunden gerne vil have | Vi levere et bedre produkt og vi sammen med kunden for mere fokuseret møder |

Risiko Formindskelse af vores weaknesses

Vores W1 (Weakness 1) udgør en unik problemstilling, da vi er kommet frem til, at sandsynligheden for, at denne risiko indtræffer, er lav. Men hvis det skulle ske, vil det have betydelige konsekvenser for vores gruppe og projekt, da den vil reducere vores team med hele 50%. Dog har vi konkluderet, at sandsynligheden for dette er lav, og det er meget usandsynligt, at det vil ske. Dette skyldes, at vores gruppe består af unge og raske individer uden allergier eller diverse sygdomme. Desuden har vi også taget årstiden i betragtning, da vi nærmer os varmere vejr, hvilket reducerer sandsynligheden for at blive forkølet eller pådrage sig andre sygdomme, der kan påvirke produktionen. Derfor ender vores risikofaktor på fem, hvilket ikke er særlig højt, men hvis risikoen indtræffer, kan det have betydelige konsekvenser.

Vi har identificeret W2 som en risikofaktor i vores projekt: mangel på tilstrækkelig vejledning fra underviserne. I tidligere opgaver har vi som gruppe haft en tendens til ikke at spørge underviserne om hjælp, og det har ført til, at vi nogle gange sidder fast på relativt enkle problemer, som kunne være blevet løst med vejledning. Dette resulterede ofte i, at vi spildte værdifuld tid på mindre væsentlige dele af programmet.

For at minimere dette problem har vi som gruppe besluttet at fokusere ekstra på at benytte os af de tilgængelige ressourcer omkring os. Vi vil aktivt søge hjælp fra underviserne og bruge code-lab til at konsultere mere erfarne programmører. Derfor har vi vurderet W2-risikofaktoren til at være på niveau 6. Vi er opmærksomme på, at dette problem har opstået i vores tidligere gruppeprojekter, og vi ønsker at holde hinanden ansvarlige for at undgå at bruge unødvendig tid på mindre betydningsfulde problemer, derfor ender risikomoment så lavt, da vi har fået den personlige erfaring til at tackle problemet.

Den sidste udfordring, vi har identificeret i vores risikoanalyse, er mødestabilitet. Vi mener, at dette kan spille en afgørende rolle i projektet, da det kan føre til spild af tid i en allerede presset projektopgave. For at tackle dette har vi indført faste stand-up møder gennem hele projektperioden. Disse møder bidrager ikke kun til en mere pålidelig mødeplanlægning, men har også andre positive effekter. De giver os et indblik i, hvad hvert enkelt gruppemedlem har afsluttet eller er i gang med. Dette skaber et godt overblik over resten af opgaven og de kommende udfordringer. Derfor har vi vurderet denne udfordring til at have en risikofaktor på 6. Selvom det ikke er umuligt, at nogen kan komme for sent, mindskes sandsynligheden betydeligt ved at have faste stand-up møder, hvor gruppemedlemmerne har en fastlagt mødetid, som de skal overholde.

3

Risiko Formindskelse af vores threats

Trussel 1 (T1) består i risikoen for, at vores kunde, Alpha-solutions, kan få leveret en bedre og mere velfungerende softwareløsning fra en anden leverandør end os. For at håndtere denne trussel har vi valgt at gennemføre flere analyser af potentielle risikomomenter, der kan påvirke vores produkt og produktivitet negativt. Disse analyser hjælper os med at forbedre vores evne til at håndtere situationer, der kan have en negativ effekt på kvaliteten af vores produkt eller forhindre os i at opfylde kundens behov og færdiggøre vores projekt.

Vi har vurderet denne trussel til at have en betydelig risikofaktor på 15, da vi mener, at konsekvenserne kan være alvorlige for vores gruppe og projekt som helhed.

Trussel 2 (T2) vedrører omkostningerne ved vores produkt. Selvom vores omkostninger generelt ikke er så høje, da vi typisk benytter værktøjer, som skolen stiller til rådighed, har vi alligevel inkluderet dette som en risikofaktor. Årsagen er, at omkostningerne stadig kan have en betydelig indvirkning på vores produkt, især hvis de skulle stige, og vi ikke har budget til at fuldføre projektet som planlagt. Derfor har vi tildelt denne trussel en risikofaktor på 5. Selvom chancen for, at vores gruppe står uden tilstrækkelige ressourcer, er meget lille, vil det have en markant indvirkning på hele vores projekt og gruppe, hvis det skulle ske.

³ 2022-Zwisler, Projekt og analyseredskaber - (læst 08/05-2023)

Trussel 3 (T3) handler om risikoen for, at vores kunde, Alpha-Solution, ikke er tilfreds med det færdige produkt og mener, at det ikke er på niveau med eller bedre end andre projektstyringsværktøjer på markedet. For at mindske denne risiko har vi valgt at arbejde tæt sammen med Product Owner (PO) og sikre, at der er tydelig fokus og klare spørgsmål ved hvert møde. Dette sikrer, at både vores gruppe og kunden er på samme side og hjælper os med at levere et fuldendt og imødekomende produkt, som kunden kan bruge og være tilfreds med.

Vi har vurderet denne risiko til at have en risikofaktor på 10, da vi mener, at sandsynligheden for, at PO og vores projektgruppe ikke er enige, er relativt lav. På den anden side har vi tildelt den en konsekvens faktor på 5, da det kan have betydelige konsekvenser, hvis kunden ikke er tilfreds med vores arbejde. Det kan føre til spildt tid og et produkt, der ikke opfylder de forventede standarder.

Aktionsplan

Det tredje element i den proaktive strategi er udarbejdelsen af en aktionsplan. Denne plan tager udgangspunkt i, at projekt medlemmerne skal sammensætte en strategi for at håndtere de øvrige risikomomenter, der er identificeret gennem risiko/SWOT-analysen, samt planlægge, hvordan projektperioden skal håndteres. For at opnå dette anvendes følgende hovedpunkter: mål, tidsramme, ansvar, risikohåndtering og kommunikation.

Målet i denne sammenhæng er at levere en klar og fokuseret beskrivelse af, hvad der skal opnås gennem aktionsplanen. Dette vil give et overblik over, hvordan projektopgaven skal håndteres.

For at udarbejde en aktionsplan, der kan hjælpe med at sikre et succesfuldt projekt, begynder vi med at etablere en klar tidsramme for projektperioden og hvordan tiden skal anvendes på en værdifuld måde. Som gruppe har vi prioriteret at afholde daglige stand-up møder kl. 11 for at vurdere vores fremskridt i forhold til backlog og user stories. Dette sikrer, at vi er i stand til at imødekomme de mål, vi har aftalt med Product Owner (PO) og os selv.

4

Herefter aftaler vi opgaver og ansvarsområder for dagen og identificerer, hvem der er ansvarlig for hvilke opgaver. Når dette er aftalt, påbegynder hvert gruppemedlem deres individuelle opgaver. Dette

⁴ 2022-Zwisler, Projekt og analyseredskaber - (læst 08/05-2023)

gentages gennem hele ugen, dog kan der være dage, hvor planen kan ændres, f.eks. hvis gruppen har brug for vejledning og derfor mødes på skolen eller hvis der er behov for code review.

En vigtig regel, som vi som gruppe har etableret, er fordelingen af ansvar mellem gruppemedlemmerne. Vi mener, at det er utroligt vigtigt for projektgruppen, at alle er klar over det ansvar, de har som en del af gruppen. Vi ønsker, at alle medlemmer af gruppen tager ansvar for at levere et godt produkt og en vellykket projektperiode, hvor alle er tilfredse med den indsats, der er blevet lagt i det endelige produkt.

Vi har fokuseret på at skabe en bevidsthed omkring individuelt ansvar og opmuntre til selvstændighed og ejerskab i forhold til opgaver og resultater. Dette skaber en stærk følelse af ansvar og engagement i gruppen, hvor hver enkelt medlem forstår, at deres bidrag har en direkte indvirkning på det endelige resultat.

For at sikre, at alle føler sig som en del af vores projektgruppe og har et ansvar for produktet, har vi valgt at lægge særlig vægt på kommunikation inden for gruppen. Vi har oprettet muligheden for kontinuerlig kommunikation, enten gennem sociale medier som f.eks. Messenger eller andre platforme. Vi mener, at denne frie og uformelle kommunikation fremmer et bedre samarbejds-klima i gruppen, da det ikke skaber en hierarkisk struktur, hvor visse medlemmer føler, at deres stemme ikke er lige så vigtig som andre, der måske er mere udadvendte. Derfor giver vi alle mulighed for at tale og give deres mening om, hvordan opgaven håndteres.

Til sidst valgte vi at fokusere på risikohåndtering ved at benytte os af flere analyser, der giver os et overblik over projektets risikomomenter og hvordan vi kan håndtere dem. En populær analyse, vi anvendte, var risiko/SWOT-analysen, som hjalp os med at identificere potentielle problemer, som vores projekt kunne stå over for. På baggrund af analysen kunne vi konkludere, at disse problematikker ikke var af så stor betydning, at de ville forhindre projektet i at starte eller imødekomme PO's forventninger. Vi er sikre på, at vores stærke analyse giver os en god forståelse af, hvordan vi kan håndtere disse risici, hvis de skulle opstå.

Risikoovervågning

Da vores gruppe konkluderede, at vi kan starte kodningsprocessen baseret på vores risikoanalyser, har vi også besluttet at foretage en yderligere analyse. Vi erkender, at der altid er en mulighed for, at andre⁵risikomomenter kan opstå, som ikke blev taget i betragtning. Derfor har vi valgt at udføre en

⁵ 2022-Zwisler, Projekt og analyseredskaber - (læst 09/05-2023)

risikoovervågning vurdering analyse, som kan identificere og håndtere eventuelle uforudsete risikomomenter, der måtte opstå.

| | |
|---------------------|---|
| Risikoovervågning | |
| Risikotype | Potentielle indikatorer |
| Teknologi | Computere bliver stjålet eller beskadiget. |
| Projektdeltagere | Motivationen af diverse Medlemmer går ned |
| Basisorganisation | Dårlig kommunikation mellem medlemmer |
| Kravspecifikationer | PO ønsker opnåelige mål PO klager |
| Estimater | Tidsplaner overholdes ikke og gruppen falder bagud. |

Vi har valgt at identificere disse risikomomenter, selvom de er usandsynlige, da det er af afgørende betydning at være opmærksom på potentielle tekniske udfordringer, der potentielt kan opstå. I tilfælde af uheldige tekniske risici har begge gruppemedlemmer sikret sig en backup af koden på GitHub og rapporten på Google Drev, hvilket giver os mulighed for at få adgang til opgaven fra enhver computer. Derudover har begge gruppemedlemmer også adgang til stationære computere, til rapportskrivning og kodning.

Da vi nærmer os slutningen af året, hvor mange studerende kan være påvirket af udbrændthed, er det yderst vigtigt for os at sikre, at alle gruppemedlemmer er motiverede til at håndtere dette omfattende projekt. Derfor lægger vi stor vægt på effektiv kommunikation og åbenhed, hvis en person har behov for en velfortjent pause. Vi har derfor også planlagt weekender som dedikerede hvileperioder, da vi er bevidste om, at det kan være udfordrende at arbejde uafbrudt uden passende belønning eller nødvendige pauser. Dette er også afgørende for at opretholde en positiv team dynamik og trivsel.

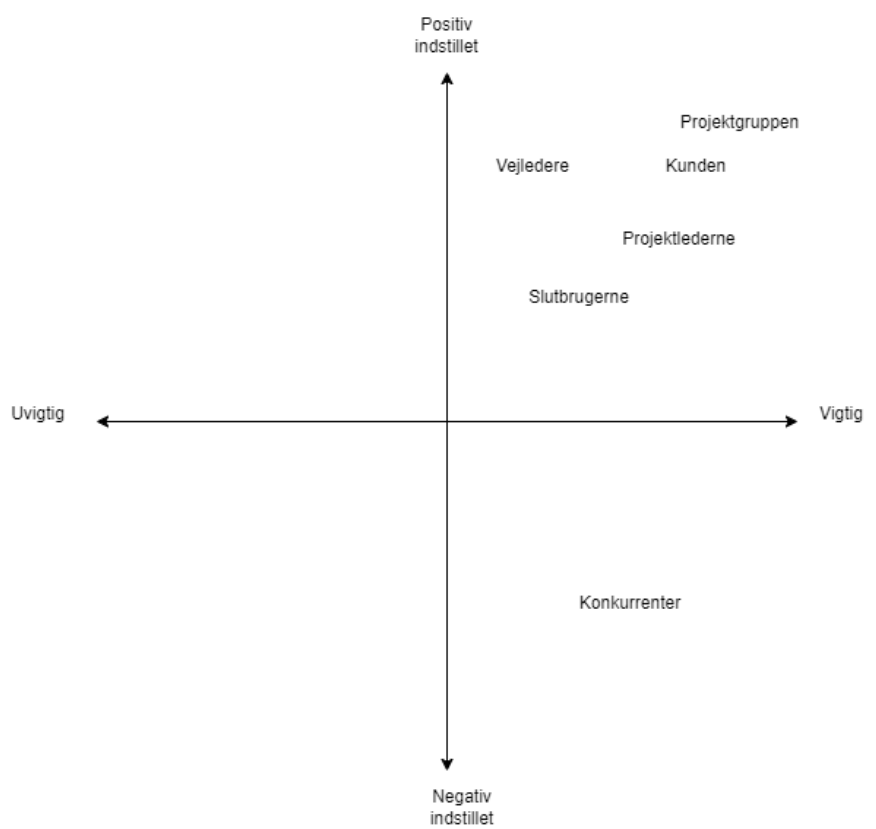
Kravspecifikationerne for både rapporten og koden/programmet har en afgørende betydning for os, og derfor har vi udarbejdet en detaljeret plan, der strækker sig over hele projektperioden. Formålet med denne plan er at sikre, at vi opfylder samtlige specifikationer og leverer et kvalitetsprodukt. Selvom vi har omhyggeligt planlagt og organiseret os, er der altid en risiko for at falde bagud. Hvis en sådan situation skulle opstå, er vi dog villige til at ofre vores weekender for at indhente det forsømte og sikre, at vi leverer et kvalitetsprodukt, som både vi og PO kan være fuldt tilfredse med.

Baseret på vores vel udarbejdede planer og proaktive foranstaltninger kan vi med sikkerhed konkludere, at udviklingsprocessen for programmet allerede er igangsat. Som en gruppe er vi dedikerede til at sikre, at det produkt, vi skaber, lever op til høje kvalitetsstandarder og opfylder PO's forventninger på bedst mulig vis.

Interessentanalyse

Vi som gruppe har også valgt at inkludere en interessentanalyse, da vi er overbeviste om, at den vil have en positiv indvirkning på vores projekt. Vi anser det desuden for yderst vigtigt at identificere de interessenter, der er involveret i projektet, dem der kan blive påvirket af projektet, samt dem der potentielt kan have indflydelse på det endelige produkts resultat.

For at opnå dette resultat er det afgørende at isolere interessenterne ved hjælp af simple, men effektive spørgsmål. Disse spørgsmål kan eksempelvis omfatte: Hvem stiller projektets opgave? Hvem vil drage fordel af projektets resultater? og så videre.



Som det fremgår af analysen, har vi valgt at fokusere på seks primære interessenter, som vi mener har den største interesse og indflydelse på vores opgave. Vi har taget højde for både negativt og positivt indstillede interessenter. På den negative side af analysen har vi identificeret vores konkurrenter, da de typisk vil have stor interesse i vores produkt for at sikre, at vores projektgruppe forbliver konkurrencedygtig på markedet inden for den specifikke sektor. På den positive side har vi også nævnt konkurrenter, da det er vigtigt for vores virksomhed at have en faktor eller aktører, der presser os til at opnå endnu bedre standarder end dem, vi startede ud med.

På den positive side af analysen har vi inkluderet projektlederne og projektgruppen, da vi anser dem som de mest væsentlige aktører for at kunne fuldføre projektet. De udgør grundstenen i vores arbejde, og deres engagement og indsats er afgørende for projektets succes. Derfor er det også dem, der har størst interesse i, at produktet bliver fuldført tilfredsstillende, da manglende opfyldelse af målene ville skabe frustration og skuffelse for dem.

Kunden og vejlederne er også blevet placeret i samme kategori, da begge parter har en interesse i et vellykket program eller produkt, der er tilfredsstillende og velfungerende. Et veludført projekt reflekterer positivt på deres kompetencer som vejledere og deres evne til at levere et kvalitetsprodukt. Det er vigtigt for både kunden og vejlederne at se, at deres vejledning og støtte har bidraget til et velfungerende projekt.

Slutbrugerne har vi inkluderet i den vigtige og positive analyse, da vi mener, at slutbrugerne er den endelige test i vores program. Dette skyldes, at slutbrugeren typisk er den faktor, der virkelig udfordrer og presser programmet til sit yderste. Vi er også af den overbevisning, at slutbrugerens interesse for produktet vil være positiv, da de ønsker det mest effektive produkt for dem.

Feasibility study

En feasibility study-analyse handler om projektet og det produkt, som vores gruppe aktivt arbejder på, og vurderer om det er gennemførligt. For at et projekt skal være gennemførligt, skal det analyseres ud fra forskellige perspektiver. Vi valgte at benytte en feasibility study, fordi den aktivt adresserer ⁶ områder, som vores proaktive analyse ikke dækker. Den proaktive analyse fokuserede primært på risikomomenter, der kunne påvirke vores produkt og projekt. Til gengæld undersøger feasibility study, om det endelige produkt overhovedet giver mening at producere. Dette opnås ved at undersøge fem underemner i feasibility study, nemlig teknisk, økonomisk, juridisk, og tidsmæssig gennemførlighed. Disse underemner giver et godt indblik i, om vores produkt og projekt er muligt og dermed gennemførligt.

7

Teknisk

Vi mener, at vores projekt er teknisk gennemførligt, da det produkt, som vi som gruppe og produktansvarlig (PO) har aftalt, kan realiseres med vores tekniske team. Vi vurderer dette ud fra vores tidligere erfaringer med lignende projekter, selvom de har været i mindre skala. Vi mener også, at vi er teknisk klar, da vi har alle de nødvendige værktøjer til rådighed for at opfylde PO's standarder og krav til produktet. Derudover har vi backup-komponenter og pc'er til rådighed, hvis vores eksisterende udstyr ikke fungerer korrekt. Derfor mener vi, at vi er teknisk gennemførlige.

Økonomisk

Vi vurderer, at vores projekt er økonomisk gennemførligt, da vi ikke står over for nogen økonomiske ulemper. Størstedelen af projektets betalingsprogrammer er finansieret af vores samarbejdspartner (KEA), hvilket betyder, at der ikke er nogen negative økonomiske konsekvenser forbundet med produktets gennemførelse. Der kan endda være potentiale for økonomisk vækst, hvis det produkt, vi

⁶ 2022-Zwisler, Projekt og analyseredskaber - (læst 11/05-2023)

⁷ <https://www.simplilearn.com/feasibility-study-article> - (læst 11/05-2023)

udvikler, har mulighed for at konkurrere med lignende produkter på markedet. Dog er det vigtigt at påpege, at vores primære fokus som en lille gruppe ikke nødvendigvis er på økonomisk vækst. Vi stræber i stedet efter at levere et fuldendt produkt, der opfylder PO's forventninger og krav.

Juridisk

Fra et juridisk perspektiv har vi prioriteret databeskyttelse og specifikt overholdelse af de nye EU-love vedrørende GDPR (General Data Protection Regulation). Vores primære mål er at sikre, at ingen personlige oplysninger om vores brugere ender i de forkerte hænder. For at opnå dette har vi implementeret en simpel tilmeldingsproces, hvor brugerne opretter en e-mail-adresse øjeblikkeligt uden behov for at angive personlige oplysninger. Dette reducerer mængden af personlige oplysninger, der er knyttet til den enkelte bruger og minimerer dermed risikoen for datalækage eller misbrug.

Derudover er vores projektstyringsværktøj kun beregnet til privat brug internt i Alpha-solutions. Det betyder, at kun autoriserede personer inden for virksomheden har adgang til brugeroplysninger. Dette ekstra sikkerhedsniveau giver ekstra beskyttelse og sikrer, at kun de rette personer kan få adgang til og håndtere brugeroplysningerne.

Ved at træffe disse foranstaltninger overholder vi GDPR og viser vores brugere, at vi prioriterer sikkerheden og fortroligheden af deres personlige oplysninger.

Tidsmæssig

Vi har en deadline, der er fastsat til den 31. maj. Vi er overbeviste om, at vi kan overholde denne deadline, da vi allerede har haft fokus på at drøfte og konkludere med product owner (PO), hvad vi kan nå i de forskellige sprints og hvordan det endelige program skal se ud. Vores plan er blevet grundigt udarbejdet for at sikre, at vi har realistiske mål og deadlines.

De første to uger er primært fokuseret på udviklingen af selve programmet, mens der kun er en minimal indsats på rapporten. Efter denne periode har vi planlagt et sekundært møde med PO, hvor vi kan afklare eventuelle spørgsmål og finpudse detaljerne. Derefter vil vi have et dedikeret fokus på rapporten, som forventes at have en længde på mellem 40 og 60 sider. Vi mener, at dette er realistisk og opnåeligt, da vi har udarbejdet en velfungerende plan, der vil guide os gennem processen og hjælpe os med at nå vores mål.

Derfor mener vi at vores projekt er **feasibel**.

Arbejdsprocessen

Daily stand-up møder

Som gruppe blev vi enige tidligt i arbejdsprocessen om at vores produktivitet er højest omkring eftermiddagen når vi mødtes. Derfor besluttede vi at holde vores stand-up møder kl. 11 om formiddagen. Dette gav os begge mulighed for at være godt forberedte, friske og fokuseret på at takle dagens udfordringer med fuld opmærksomhed og energi.

Under vores stand-up møder havde vi to primære mål. For det første ønskede vi at sikre, at alle i gruppen var klar over, hvilke opgaver der skulle takles. Vi mente, at dette var af afgørende betydning, da det er nemt at føle sig overvældet i et stort projekt med mange forskellige opgaver. Ved at skabe klarhed omkring opgaverne kunne vi mindske forvirring og sikre, at alle var på samme side.

For det andet fokuserede vi på at dele opgaven op i mindre dele og opdatere hinanden om vores fremskridt. Dette gav os et godt overblik over, hvilke mål blev fuldført, og hvad der eventuelt manglede. Det gjorde det også muligt for os at identificere områder, hvor der var behov for vejledning eller støtte fra vejledere.

Ved at implementere disse praksisser skabte vi en struktureret og produktiv arbejdsproces, hvor alle kendte til deres ansvar, og hvor der var plads til spørgsmål og vejledning. Dette bidrog til at skabe et positivt og effektivt samarbejds miljø i vores gruppe.

Sprint Planning

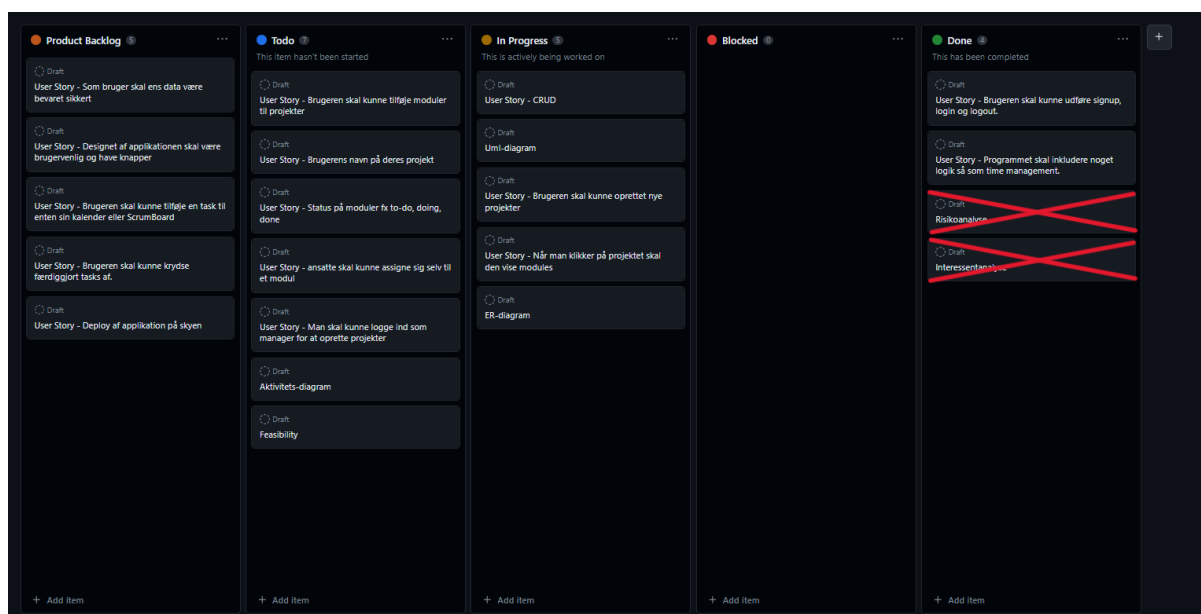
I vores sprintplanlægning havde vi som vores allerførste prioritet at definere klare og specifikke mål, der var fokuseret på vores backlog og vores Product Owner (PO) havde ønsket. For at opnå dette valgte vi at stille præcise og konkrete spørgsmål, som blev udledt fra den opgavebeskrivelse, vi fik udleveret i starten af projektperioden. Vi havde grundigt studeret opgavebeskrivelsen og mente, at vi allerede havde en god forståelse af, hvad PO ønskede. Men for at sikre, at PO var tilfreds, holdt vi tæt kontakt og sørgede for, at alle i teamet var fuldt ud klar over PO's forventninger.

⁸ 2022-Zwisler, Projekt og analyseredskaber - (læst 14/05-2023)
<https://www.simplilearn.com/feasibility-study-article> - (læst 14/05-2023)

Vi prioriterede en åben og løbende kommunikation samt et tæt samarbejde med vores PO for at sikre, at vores mål og forventninger var i overensstemmelse med PO's overordnede vision for projektet. Dette involverede regelmæssige samtaler og afklaring af spørgsmål for at sikre, at vi var på rette spor og i stand til at levere det ønskede resultat.

Ved at lægge vægt på tydelige mål og et tæt samarbejde med PO var vi i stand til at skabe en solid og veldefineret retning for vores sprint.

Vi lavede en grundig nedbrydning af user stories, hvilket gav os et gennemskueligt overblik over resten af projektet. Denne nedbrydning af opgaver gjorde det muligt for os at håndtere forskellige opgaver dag for dag og sikrede, at vi kunne bevæge os fremad på en struktureret og organiseret måde.



Screenshot fra slutningen af vores første møde med Product Owner (PO). (De røde kryds, indikerer at begge virksomheds analysere ikke var færdig, men placerede forkert).

I løbet af mødet diskuterede vi og blev enige om, hvad vi som gruppe mente ville være i stand til at fuldføre med god kvalitet inden næste møde. Vi besluttede også at undgå at påtage os for meget, da vi er et lille hold, der ønsker at opretholde fokus på både kvalitet og en grundig gennemgang af opgavebeskrivelsen. Vores mål var at skabe realistiske forventninger og sikre, at vores arbejde var håndterbart og af høj kvalitet.

Sprint Review

Efter afslutningen af vores første sprint kunne vi som gruppe konkludere, at vi havde valgt en passende mængde opgaver, som vi var i stand til at fuldføre. Vi nåede at færdiggøre stort set alle delopgaver. Dog havde vi stadig en vis arbejdsbyrde i løbet af ugen, der krævede vores

opmærksomhed. Vi var tilfredse med vores evne til at opretholde et godt arbejdstempo, samtidig med at vi sikrede, at alle opgaver blev udført på en tilfredsstillende måde.



Screenshottet viser hvad vi har opnået i løbet af første sprint, som strakte sig over en periode på to uger. Det er tydeligt, at vi har fuldført alt det, vi havde planlagt for dette sprint. Vi er stolte af vores præstation og indsats for at kunne levere Product Owners ønskede resultater inden for den fastsatte tidsramme.

Sprint Retrospective

Arbejdsvurdering

Efter at have arbejdet som gruppe i en periode på 4-5 uger, er vi enige om, at vores samarbejde har været tilfredsstillende og fokuseret. Vi har haft god kommunikation gennem hele processen, hvilket har spillet en afgørende rolle for vores succes med at fuldføre projektet. Ved at have åbne og effektive kommunikationskanaler har vi kunnet udveksle informationer, løse problemer og sikre, at alle var på samme side.

En vigtig del af vores samarbejde har været brugen af stand-up møder og vores Scrum-board. Scrum-boardet har været en nyttig visuel repræsentation af vores opgaver og status, hvilket har gjort det lettere for os at tackle og organisere arbejdet.

Vi mener, at rollen som Scrum Master var af stor betydning i vores projekt. Scrum Masteren var ansvarlig for at opretholde og opdatere vores Scrum-board kontinuerligt gennem hele projektperioden. Dette bidrog til at sikre, at alle opgaver var klart definerede og synlige for hele gruppen.

Selvom Scrum Masteren holdt os ansvarlige og fulgte op på vores fremskridt, fandt vi det gavnligt at have individuel kontrol over opdateringen af vores egne opgaver. Dette gav os en følelse af ansvar og ejerskab over vores arbejde og hjalp med at opretholde en høj grad af opdatering og nøjagtighed på Scrum-boardet.

Stand-up møderne har givet os en daglig mulighed for at opdatere hinanden, identificere eventuelle udfordringer og holde os fokuserede på vores mål. Disse møder har skabt en følelse af samhørighed og forståelse inden for gruppen.

Svagheder og styrker

En udfordring, der lejlighedsvis opstod i vores gruppe, var mødestabiliteten til vores stand-up møder. Dette skyldtes, at nogle af gruppemedlemmerne indimellem arbejdede sent på projektet og derfor ikke fik tilstrækkelig søvn. Som følge heraf kunne de komme lidt forsinket til stand-up møderne. Ikke desto mindre var vi som gruppe fleksible med hensyn til mødetidspunkter og formåede stadig at afholde effektive møder, selv når dette forekom.

En af vores styrker som gruppe i projektet var den kontinuerlige og åbne kommunikation, vi opretholdt gennem hele processen. Vi holdt konstant kontakt med hinanden og var altid parate til at

hjælpe, hvis der opstod problemer eller behov for assistance. Denne stærke kommunikation spillede en afgørende rolle i vores evne til at fuldføre projektet tilfredsstillende. Derudover brugte vi vores lille gruppestørrelse til vores fordel, da det er nemmere at delegere arbejde til få personer.

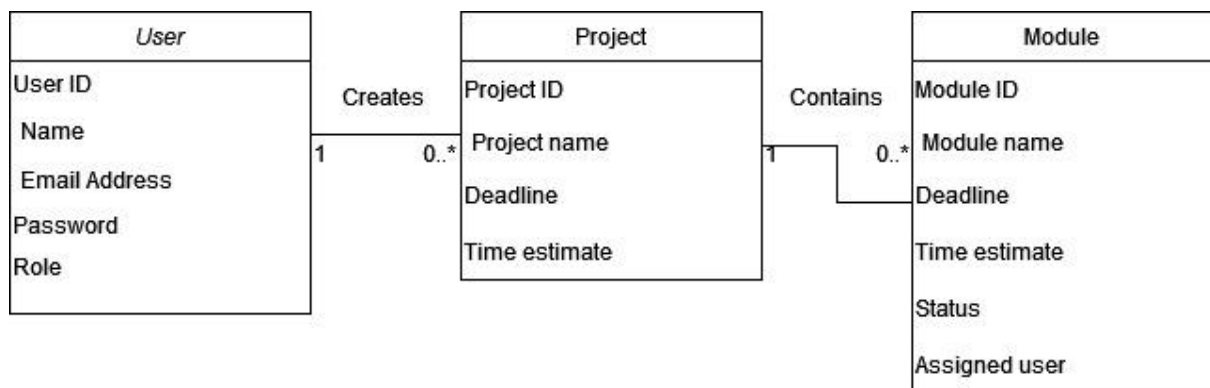
Forbedringer

Hvis vi skulle identificere et område, hvor vi kunne forbedre vores arbejdsproces, ville vi pege på mødestabiliteten. Det er vigtigt at være opmærksom på, at der var tilfælde, hvor mødestabiliteten ikke var helt på niveau med resten af projektet. Vi mener, at dette er et område, hvor vi kan stræbe efter at opnå større konsistens og pålidelighed.

Teknologivalg

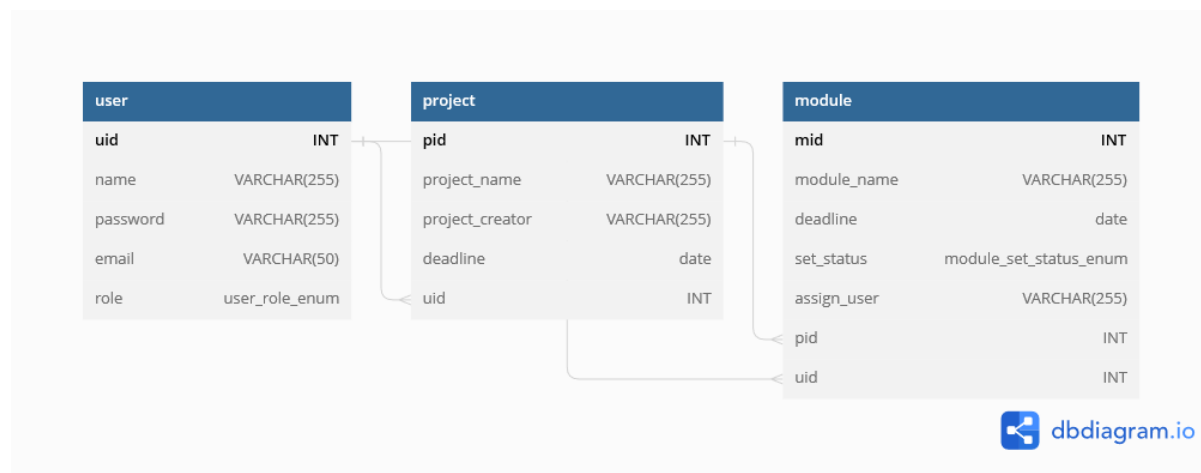
- IntelliJ IDEA 2022.2.4 (Ultimate Edition)
- Java 19
- MySQL Workbench - 8.0.32
- Springboot - 3.0.6
- Thymeleaf - 3.0.6
- HTML - HTML5
- CSS - CSS3
- Bootstrap - 5.0.2
- Maven - 4.0.0

Domænemodel og database



Ovenstående model er domænet for projektet, der viser relationerne mellem de forskellige dele af programmet. Hver tabel har sine egne attributter og hovedpunktet er at en bruger kan have nul eller

flere projekter, men et projekt vil altid være bundet til én bruger. På samme facon er moduler bundet til ét projekt, men et projekt kan have nul eller mange moduler.



I hver tabel bruges et automatisk inkrementeret ID nummer som primærnøgle; “uid” for user ID i “user” tabellen, “pid” for project ID i “project” tabellen og “mid” module ID i module tabellen.

I “project” og “module” tabellen bruges fremmednøgler for at kunne “joine” tabellerne sammen som bruges til SQL queries. Dette kan være brugbart når man skal finde f.eks. et modul som tilhører inde under et projekt. I næsten alle felter har vi valgt at attributter skal være “NOT NULL”, hvilket betyder at der skal være en specificeret værdi. De eneste rækker som der gerne må være null er dem som alligevel senere hen bliver opdateret med værdier. Et eksempel på dette er projects “time_estimate” hvis værdi afhænger af modulernes tidsestimat inde under projektet. I “module” findes der også ”assign_user” rækken som standard vil være “Unassigned” indtil en “manager” på programmet tildeler en “employee” til modulet.

For at vurdere kvaliteten af denne database struktur kan vi kigge på normalformer. Vi kan se, at hver tabel har sin egen primærnøgle og ingen af rækkerne er gentagne. Dette må betyde, at databasen må mindst være på første normal form. Derudover er alle attributter i tabellerne afhængige af primærnøglerne og disse attributter er alle uafhængige af hinanden. Dette opfylder både kriterierne for normalform 2 og 3⁹. Databasen er altså på tredje normalform.

Hvad er firmaets håb med dette system?

For at kunne forstå Alpha-solutions' vision er det vigtigt at kende til virksomheden. Alpha-solutions blev grundlagt i 2004 som en dedikeret IT- og e-handelsvirksomhed. Virksomheden har altid haft fokus på at levere de bedst mulige tekniske løsninger til deres kunder, hvilket har gjort dem til en af de mest kompetente virksomheder inden for IT-branchen i Norden. Alpha-solutions' overordnede mål er altid at fokusere på kvalitet og imødekomme kundens teknologiske behov.

⁹ https://en.wikipedia.org/wiki/Database_normalization

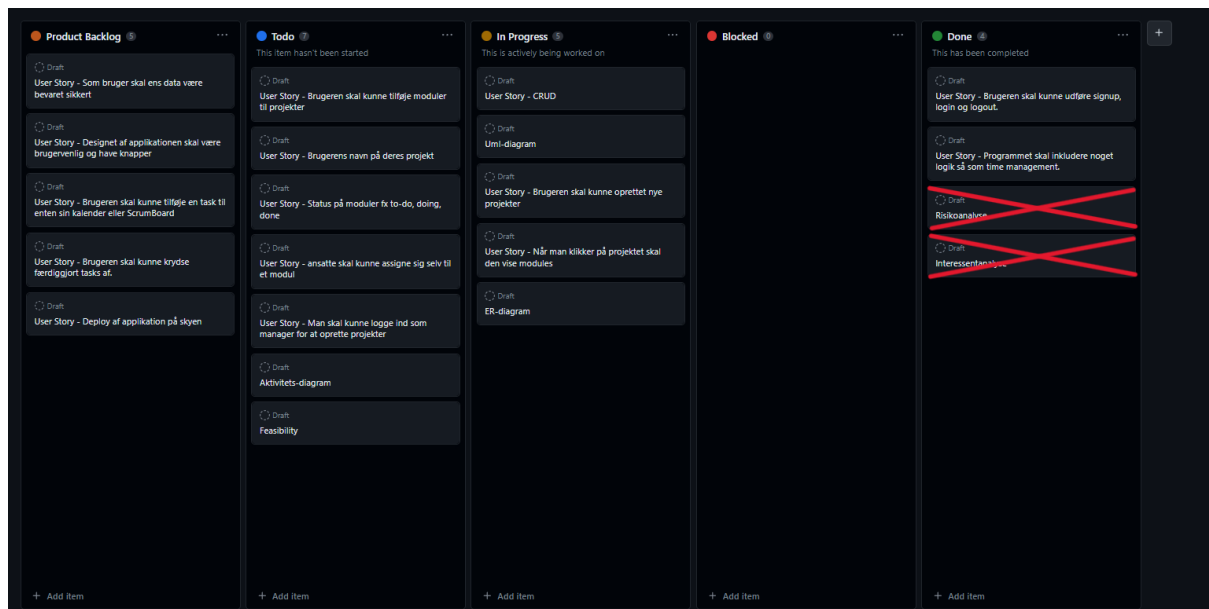
Vi har taget denne fokus på kvalitet til os i vores gruppe og har derfor valgt at arbejde tæt sammen med vores kunder. Vi mener, at Alpha-solutions' vision for dette produkt er at skabe et letlæseligt program, der kan introducere deres team til projektstyringsværktøjer samtidig med at det øger¹⁰ produktiviteten i hele virksomheden. Vi tror på, at vores projektstyringsprogram kan tilføre stor værdi til Alpha-solutions, da det fremmer produktivitet og samarbejde på tværs af teams og afdelinger.

Derfor har vores gruppe igangsat produktionen af Alpha-solutions programstyring program, som kommer til at dele de samme kvaliteter som vores kunde.

User stories

Sprint 1.

I vores første sprint havde vi stor fokus på to væsentlige aspekter, som vi mente var yderst vigtige for vores gruppe. Vores første fokus var at estimere, hvad vi kunne nå indtil næste sprint som en to-mands gruppe. For at skabe et sådant overblik valgte vi at fokusere på de vigtigste elementer i et projektstyringsprogram, som vi mente var nødvendige for brugerne at have. Ved at have fokus på disse nødvendigheder fik vi et indtryk og et overblik over, hvad vi som gruppe kunne nå, og det hjalp os med at danne et billede sammen med product owner (PO) i forhold til programmets omfang.



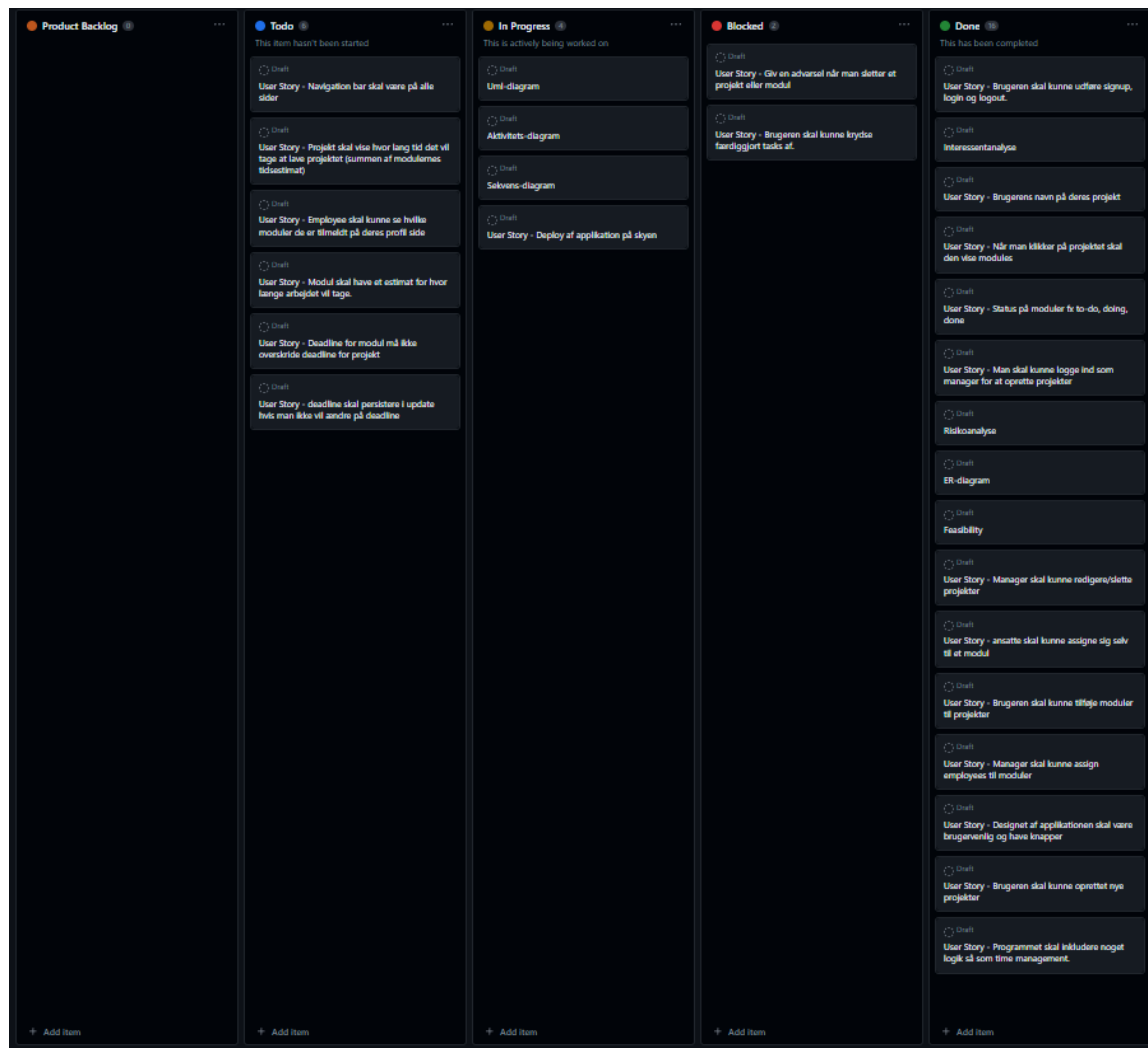
Dette var konklusionen, vi nåede frem til i samarbejde med kunden. Vi holdt det simpelt og fokuseret. Desuden konkluderede vi, at vi kunne implementere flere elementer fra vores backlog, hvis det var muligt inden for rammerne af det første sprint.

Sprint 2.

Efter vores første sprint konkluderede vi, at estimeringen af de valgte opgaver sammen med kunden var tilstrækkelig, og vi havde endda lidt ekstra tid til rådighed. I denne ekstra tid lykkedes det os at implementere nogle ekstra user stories, som måske ikke var af højeste prioritet, men som forbedrede programmet og brugeroplevelsen. Vi stod dog over for to user stories, der blev blokeret på grund af problemer. Disse user stories valgte vi ikke at prioritere, da de ikke var nødvendig for opgaven, men øvede brugerens livskvalitet minimalt.

Konklusion af sprint

Efter at have afholdt disse værdifulde møder med vores PO, er vores gruppe overbevist om, at vi er godt rustet til at tackle og fuldføre vores projekt. Disse møder har været af stor værdi, idet de har givet os et omfattende indblik i kundens forventninger til et projektstyringsprogram. Dette har ligeledes hjulpet os med at danne et klart billede af, hvordan vi bedst kan håndtere dette omfattende projekt med fokus på kvalitet og kundetilfredshed.



I vores andet møde med PO fokuserede størstedelen af diskussionen på brugervenlighed og hvordan vi kunne skabe en bedre brugeroplevelse. Vi valgte at implementere relativt simple, men effektive funktioner, der ville give brugerne en forbedret oplevelse, når de benyttede vores program. Et eksempel på dette er implementeringen af en taskbar, der er tilgængelig gennem hele systemet. Dette skaber både en mere konsistent oplevelse for brugerne og muliggør nem adgang til funktioner som log ud.

Structured Query Language

MySQL (Structured Query Language) blev første gang introduceret i 1974 og blev hurtigt populært, inden det blev standardiseret i 1986. MySQL er et relationelt databaseprogrammeringssprog, hvilket betyder, at det understøtter tabeller, der har relationer med hinanden ved hjælp af nøgler. Ved at bruge SQL kan man organisere og manipulere store mængder data effektivt.

Vi valgte at bruge MySQL som udgangspunkt, fordi det er en passende back-end til vores overordnede program. Da vi skulle indsamle relationelle data fra vores brugere, gav MySQL os mulighed for at håndtere disse data på en effektiv måde. Vi kunne oprette og administrere tabeller, etablere relationer mellem dem ved hjælp af nøgler og udføre komplekse forespørgsler for at hente og manipulere data efter behov.

```
CREATE TABLE user (
  uid INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  password VARCHAR(255) NOT NULL,
  email VARCHAR(50) NOT NULL,
  role ENUM('ADMIN', 'MANAGER', 'EMPLOYEE') NOT NULL,
  PRIMARY KEY (uid)
);

CREATE TABLE project (
  pid INT NOT NULL AUTO_INCREMENT,
  project_name VARCHAR(255) NOT NULL,
  project_creator VARCHAR(255),
  deadline date NOT NULL,
  uid INT NOT NULL,
  PRIMARY KEY (pid),
  FOREIGN KEY (uid) REFERENCES user(uid) ON DELETE CASCADE
);

CREATE TABLE module (
  mid INT NOT NULL AUTO_INCREMENT,
  module_name VARCHAR(255) NOT NULL,
  deadline date NOT NULL,
  set_status ENUM('TO_DO', 'DOING', 'DROPPED', 'DONE') NOT NULL,
  assign_user VARCHAR(255) default 'Unassigned',
  pid INT NOT NULL,
  uid INT NOT NULL,
  PRIMARY KEY (mid),
  FOREIGN KEY (pid) REFERENCES project(pid) ON DELETE CASCADE,
  FOREIGN KEY (uid) REFERENCES user(uid) ON DELETE CASCADE
);
```

Dette billede viser vores SQL-statement, hvor vi har tre tabeller: "user", "project" og "module". Vi har valgt at inkludere disse tabeller, da vores programstruktur er afhængig af dem for at administrere relationelle data. Disse tabeller giver os mulighed for at organisere og opbevare informationer om brugere, projekter og moduler på en sammenhængende måde, ved at definere relationer mellem tabellerne.

SpringBoot

I vores webapplikation benyttede vi os af Spring Boot. Spring Boot er et open-source framework, der leverer mange nyttige værktøjer, som øger produktiviteten ved at minimere behovet for manuel konfiguration. En af fordelene ved Spring Boot er, at det inkluderer en indbygget webserver, som kan hostes lokalt. Dette er en stor fordel, da det giver os muligheden for at få et hurtigt overblik over webapplikationen, før den bliver udgivet i offentligheden.

```
#Port application
server.port=3306
spring.datasource.url=jdbc:mysql://localhost:8080/productmanagementtooldatabase
spring.datasource.username=root
spring.datasource.password=
```

(Eksempel på vores SpringBoot.)

Ved at benytte Spring Boot har vi haft adgang til effektive værktøjer, der har bidraget til vores produktivitet. Det har også gjort det muligt for os at teste vores webapplikation inden den bliver deployed i skyen. Ved at benytte Spring Boot har vi haft adgang til effektive værktøjer, der har bidraget til vores produktivitet og hjulpet os med at levere et endnu mere fuldendt produkt.

Thymeleaf

Vi mente, at Thymeleaf ville have en nyttig indvirkning på vores program. Det er fordi Thymeleaf er i stand til at generere dynamisk indhold til vores webapplikationer. Dette er særligt vigtigt, da vores webapplikation involverer brugerinteraktioner gennem hele programmet. Ved at bruge Thymeleaf kan vi opdatere og præsentere data i virkelig tid baseret på vores brugeres handlinger. Dette skaber et mere effektivt miljø både for os som programmører og for brugerne af vores program. Thymeleaf sikrer, at vores webapplikation ikke er statisk, men i stedet kan tilpasse sig og levere relevant indhold i overensstemmelse med brugerens handlinger og behov.¹¹

¹¹ <https://en.wikipedia.org/wiki/SQL> - (læst - 18/05-2023)
<https://en.wikipedia.org/wiki/Thymeleaf> - (læst - 18/05-2023)

```

<!DOCTYPE html>
<html lang="en">
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Login</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
  <link rel="stylesheet" href="/login-style.css">
</head>
<body>
<div class="container mt-5">
  <div class="row justify-content-center">
    <div class="col-md-6">
      <div class="card">
        <div class="card-body">
          <h1 class="text-center mb-4">Login</h1>
          <form th:action="@{/login}" method="post">
            <div class="form-group">
              <label for="email">Email</label>
              <input type="email" class="form-control" id="email" th:field="*{loginForm.email}" th:placeholder="Email" required>
            </div>
            <div class="form-group">
              <label for="password">Password</label>
              <input type="password" class="form-control" id="password" th:field="*{loginForm.password}" th:placeholder="Password" required>
              <div> <a href="/signup">Dont have an account?</a>
            </div>
            </div>
            <button type="submit" class="btn btn-primary btn-block mt-4" th:href="@{/profile}">Login</button>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>
</body>
</html>

```

Her er et eksempel på vores brug af Thymeleaf i vores HTML-fil. Vi starter med at erklære Thymeleaf-namespace ved at tilføje `xmlns:th="http://www.thymeleaf.org"` i `<html>`-taggen. Dette giver os mulighed for at bruge Thymeleaf-attributter og udtryk i vores dokument.

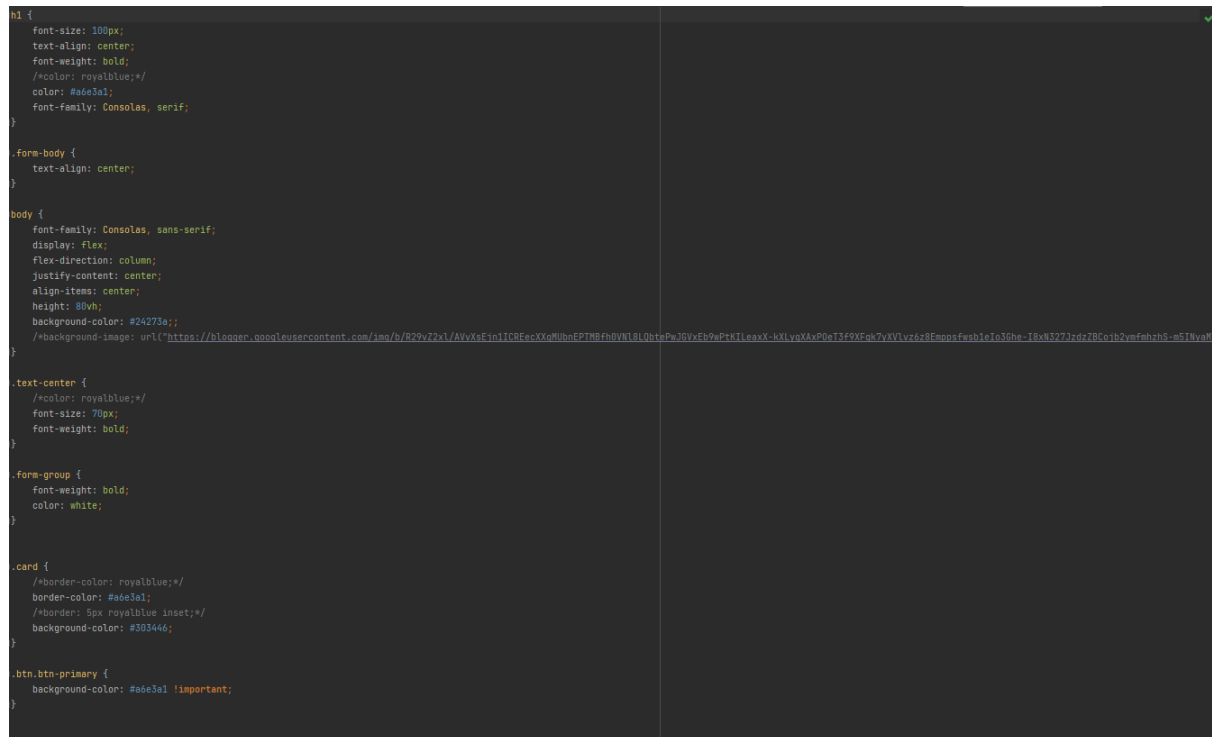
I koden ser vi flere steder attributter, der starter med `th:`. Disse attributter bruger vi til at definere dynamiske handlinger i vores program. For eksempel bruger vi `th:action` til at specificere handlingsattributten for vores formular og `th:field` til at binde formularfelter til Java-objekter.

Cascading Style Sheets

Brugen af CSS i vores program er af stor betydning, da det er med til at skabe det visuelle miljø, som brugerne oplever, når de interagerer med vores program. Vi har valgt at benytte CSS, da det er et kraftfuldt stilarkssprog, der giver os mulighed for at definere og ændre udseendet af både enkelte websider og hele webapplikationen.

CSS hjælper os med at tilpasse farver, skrifttyper, layout, animationer og mange andre visuelle egenskaber. Ved at anvende CSS kan vi skabe en ensartet og professionel brugeroplevelse på tværs af

vores webapplikation. Vi kan også tilpasse designet efter vores brand eller tema, hvilket bidrager til at skabe genkendelighed og identitet.



Her er et billede af vores stylesheet fra vores egen webapplikation. Vores stylesheet er simpelt, men effektivt. Det er designet til at skabe et pænt miljø, som vores brugere vil føle sig velkomne i.

Teknologi

Produktionsmiljø

Vi valgte at benytte Docker som et effektivt værktøj til vores program, da det tilbyder en række fordele, som vi finder brugbare og meningsfulde.

Docker giver os mulighed for at isolere vores applikation og dens afhængigheder i separate containere. Dette betyder, at vores applikationer kan køre uafhængigt af hinanden uden nogen form for indbyrdes påvirkning. Isolationen hjælper også med at sikre, at vores program kan fungere effektivt på tværs af forskellige miljøer, da Docker reducerer risikoen for konflikter mellem forskellige softwarekomponenter.

Ved at oprette separate containere, der indeholder alle de nødvendige komponenter til vores program, bliver det også muligt at genskabe programmet på forskellige systemer uden at opleve negative virkninger.

En af de store grunde til, at vores gruppe valgte at benytte Docker, er netop den nemme genskabelse af vores program. Ved at implementere Docker kan vi nemt distribuere vores applikation hurtigt og problemfrit på tværs af næsten alle miljøer.

Et argument som kan bruges imod vores beslutning for at tage Docker som værktøj cloud deployment er at når der foretages ændringer i programmet, kan genopbygningen af et nyt Docker image være tidskrævende. Hvis man vil køre Docker lokalt på sin computer, er der også en indlæringskurve, som man skal passe sig til. Ydermere, når der skal genopbygges applikationen på Render, tager det også meget lang tid, da der bruges gratis service.

Udviklingsmiljø

I vores gruppe har vi valgt at benytte os af GitHub branching for at let implementere tests, der kan adskilles fra det produktionsklare produkt. Vi valgte GitHub branching, da det giver os mulighed for nemt at adskille vores master-branch fra resten af vores tests, som vi vidste ikke var klar til at blive implementeret endnu. Vi har derfor valgt at bruge feature branches, da det giver os mulighed for at implementere og teste nye metoder og funktionaliteter, inden de får det endelige godkendelsesstempel.

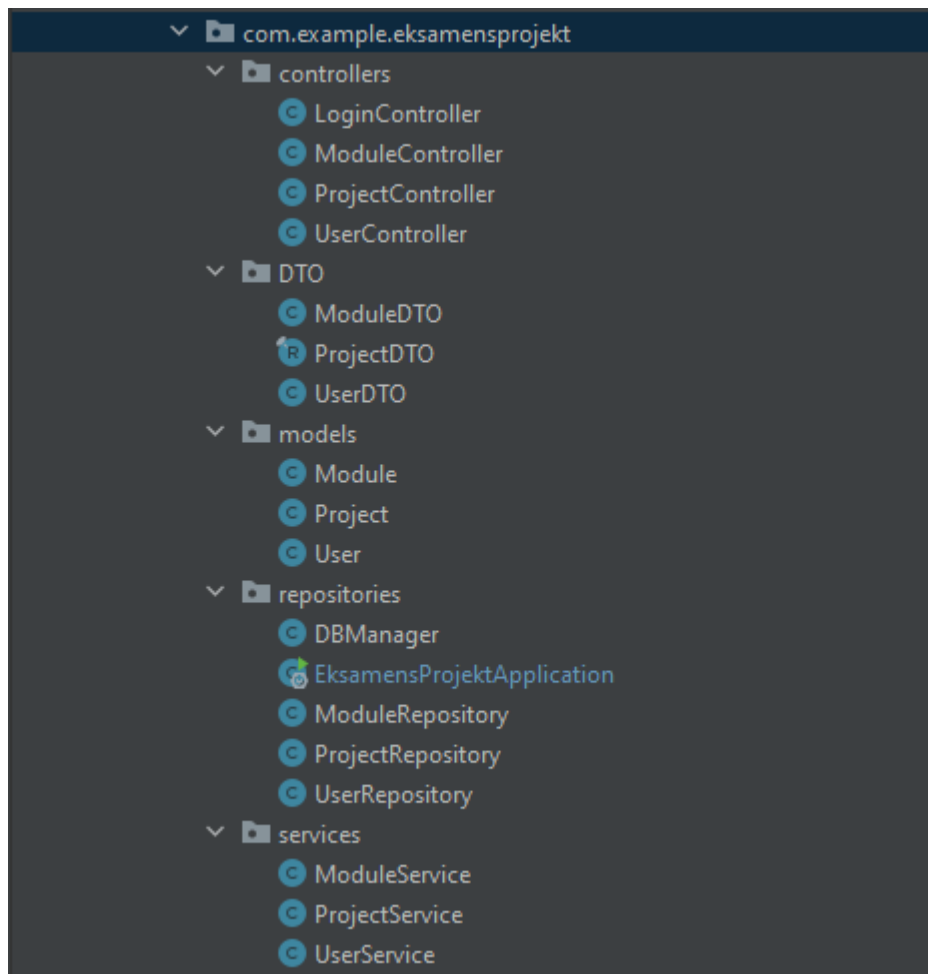
Den effektive del af GitHub branching er også pull requests og code review. Dette giver resten af holdmedlemmerne mulighed for at gennemgå vores tests og sikre, at de kan merge med master-branchen, når de er klar til det. Ved at oprette en pull request kan vi indhente feedback og foreslåede ændringer fra andre holdmedlemmer, hvilket bidrager til kvaliteten og stabiliteten af vores kode.

Denne tilgang giver os mulighed for at opretholde en god udviklingspraksis og organisere vores arbejde på en struktureret måde.

Software Design

Arkitektur

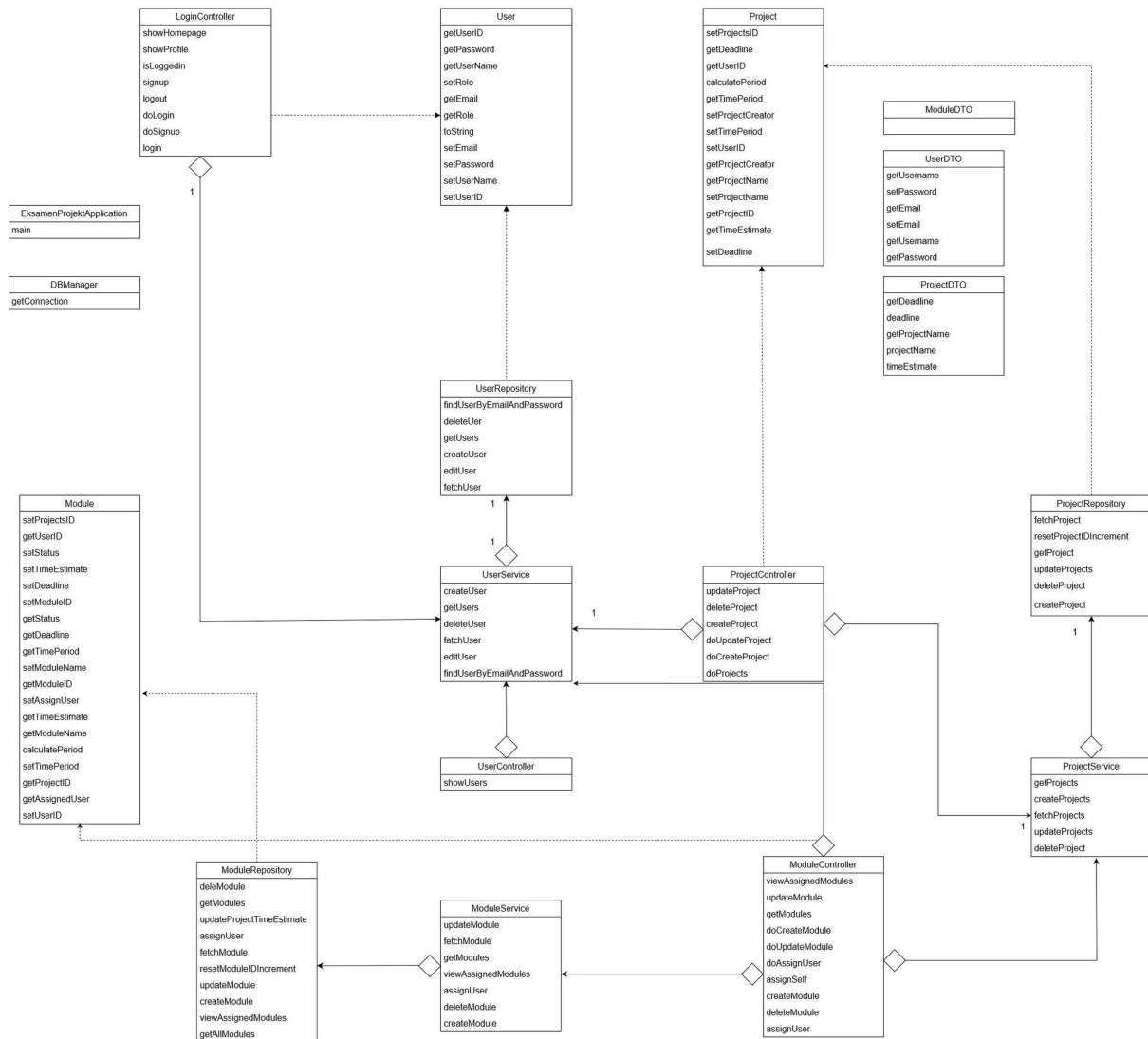
Vi mener, at vores pakkestruktur er intuitiv og overskuelig. Vi har bevidst valgt at opdele vores pakke på denne måde, da det fremmer en bedre forståelse af programmet og gør det lettere for andre programmører at identificere, hvad der er inkluderet. Dette bidrager til et mere effektivt arbejdsmiljø, da det sparer tid og ressourcer i forhold til at skulle bruge ekstra tid på at forstå et program, hvis det var opstillet på en mere uorganiseret måde.



Klassediagram

Grunden til, at vores gruppe valgte at lave et klassediagram, var, fordi det er et effektivt værktøj til at give klar forståelse for de forskellige klasser og deres relationer med hinanden. Desuden er et klassediagram nyttigt i forbindelse med teknisk gæld, da det giver et overblik over programmets struktur. Fremtidige programmører vil gavne dette, da man nemt kan se hvordan hele systemet er opbygget.

UML diagram



(OBS - Link til vores klasse diagram vil være inkluderet i bilag og github.)

Vi valgte kun at fokusere på klasserne og metoderne inkluderet i de forskellige diagrammet, da vi gerne ville have, at diagrammet var læsevenligt og kunne skabe et ordentligt overblik over vores program. Et klassesdiagram fungerer også som en form for kommunikation, da det kan være et fælles sprog mellem alle parter i projektet, lige fra designere til udviklere osv.

Design Overvejelser

Enkelhed (Simplicity)

Som nævnt før har vi valgt at gøre vores UML-diagram simpelt og læsevenligt. Dette er et vigtigt designprincip, som undgår komplekse hierarkier og unødvendige attributter, som blot skaber

forvirring til læsning af diagrammet. Dette princip er essentielt, da UML-diagrammet fungerer som en kommunikationsform til andre, der skal forstå vores programs opbygning, herunder personer, der måske ikke har været involveret fra starten.

Vi har bevidst valgt denne tilgang, da vi ønsker at sikre, at UML-diagrammet er let at læse og forstå.

Høj sammenhæng(High Cohesion)

Klasserne vi har valgt at inkludere i vores endelige program spiller alle en væsentlig rolle for at opnå en løsning som møder Product Owner's forventninger og opretholder kvaliteten på tværs af programmet. Vi har bevidst fokuseret på dette for at undgå, at uvæsentlige klasser fylder i resten af vores program.

Vores mål er at opnå høj sammenhæng mellem klasserne, hvor hver klasse har et veldefineret ansvarsområde og kun indeholder metoder og attributter, der er relevante for den pågældende opgave

Lav kobling (Low Coupling)

I vores endelige program har vi prioriteret at opnå lav kobling mellem klasserne. Dette betyder, at klasserne er uafhængige af hinanden og kun afhænger af hinanden, når det er nødvendigt. Vi har taget denne tilgang til at skabe et bedre programmeringsmiljø, hvor vores program bliver mere fleksibelt. Ydermere betyder det, at når programmører skal foretage ændringer i klasserne, vil andre dele af programmet forblive fejlfrie uden at skabe nye problemer. Systemets fleksibilitet øges og den tekniske gæld mindskes, da det er lettere at vedligeholde og udvide systemet i fremtiden.

¹²

Overensstemmelse (Consistency)

Som der vises i UML-diagrammet, er programmet opbygget med et fokus på konsistens ved at følge ensartede notationer. Dette bidrager til at gøre programmet mere læsevenligt og lettere at analysere for fremtidige interessenter.

Ved at bruge den samme notation i hele UML-diagrammet sikrer vi, at symbolerne og relationerne har en konsekvent betydning og fortolkning.

Opdeling og modulopbygning (Decomposition and Modularity)

¹² [https://en.wikipedia.org/wiki/Coupling_\(computer_programming\)](https://en.wikipedia.org/wiki/Coupling_(computer_programming)) (læst - 18/05-2023)
<https://stackoverflow.com/questions/14000762/what-does-low-in-coupling-and-high-in-cohesion-mean>
(læst - 20/05-2023)

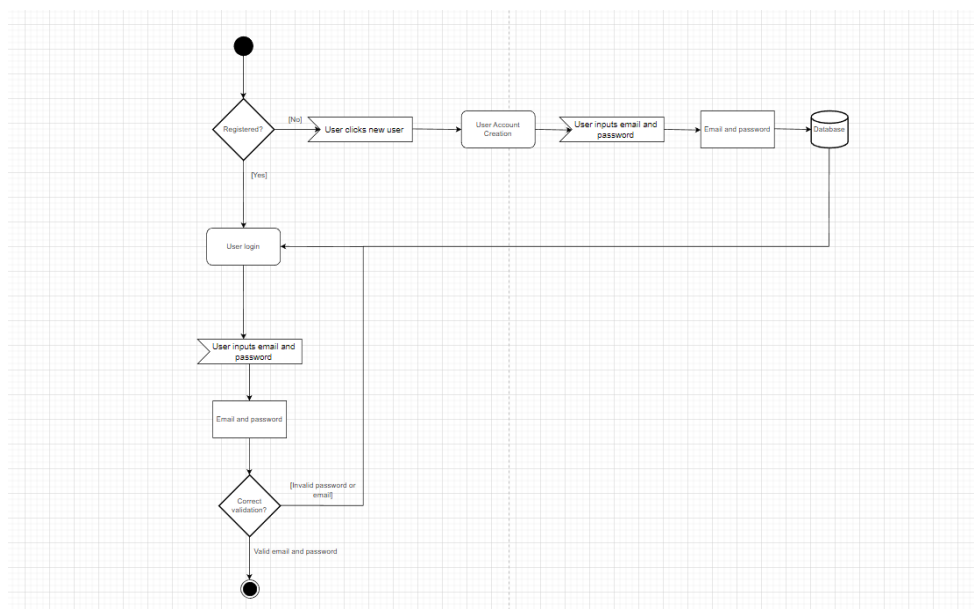
Dette princip kan ses i vores pakkestruktur, hvor vi har opdelt vores klasser i forskellige pakker. Denne opdeling bidrager både til forståelsen af programmet og til at etablere klare ansvarsområder for vores klasser. Det hjælper os med at skabe en mere overskuelig og struktureret kodebase, hvilket igen gør det nemmere at vedligeholde programmet over tid.

Navigations Flow mellem skærm billeder

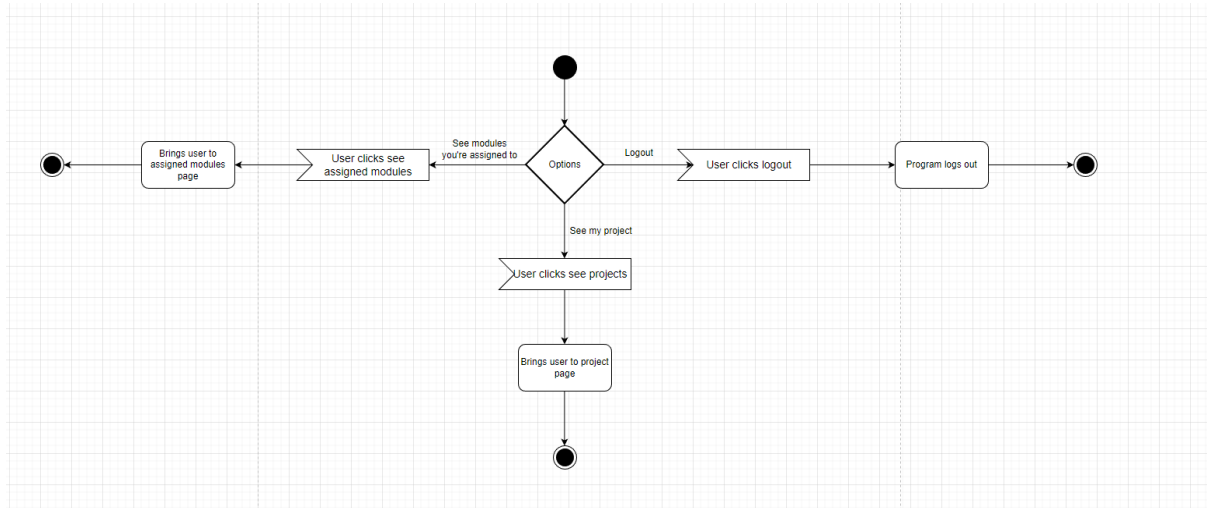
I komplekse systemer kan det være udfordrende at navigere, da der er mange sammenhængende websider, som fører brugeren til forskellige destinationer. For at skabe klarhed og oversigt har vi valgt at udarbejde aktivitetsdiagrammer. Disse diagrammer fungerer som visuelle værktøjer, der hjælper både os og brugeren med at forstå programmets struktur og navigation. Ved at analysere og planlægge brugerens interaktionstrin og det overordnede flow mellem forskellige sider og funktioner.

Log ind

Dette aktivitetsdiagram giver en visuel repræsentation af brugerens rejse på vores webside fra det øjeblik, de træder ind. Brugeren står over for to valgmuligheder: Har de allerede en profil, eller skal de oprette en? Hvis brugeren allerede har en profil, bliver de guidet til loginsiden for at indtaste deres loginoplysninger. Hvis disse oplysninger er korrekte, vil de få adgang til deres personlige profil. Hvis oplysningerne er forkerte, sendes de tilbage til startsidens for at forsøge igen. Hvis brugeren ikke har en profil, vil de blive bedt om at oprette en ved at indtaste deres e-mail og valgt adgangskode. Disse oplysninger bliver derefter gemt i vores database til brug for fremtidig adgang. Efter succesfuld registrering bliver brugeren ført tilbage til startsidens, hvor de kan logge ind med deres nyoprettede profil.



Menuen



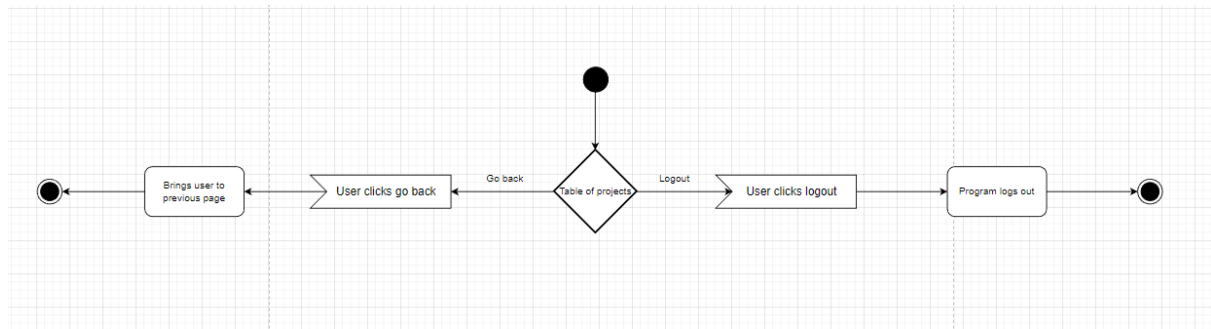
Dette er en visuel repræsentation af brugernes muligheder, når de har logget ind på vores webapplikation med succes. Efter login vil brugerne blive præsenteret for en menu, der tilbyder op til tre valgmuligheder, afhængigt af deres brugerrolle. Dette aktivitetsdiagram fokuserer primært på ansatte fra Alpha-solutions.

Der er tre grundlæggende valgmuligheder i menuen. Den første mulighed er at logge ud, hvilket vil føre til, at brugeren bliver logget ud af programmet. Hvis de ønsker at logge ind igen, skal de foretage en ny login-proces.

Den anden mulighed er at klikke på "See my projects"(se mine projekter). Programmet vil herefter dirigere brugeren til en ny webside, hvor de kan se en oversigt over alle deres projekter. Dette er også gældende for "Assigned Modules" (tildelte moduler). Brugeren vil blive omdirigeret til en webside, der viser specifikke oplysninger om de moduler, der er tildelt til dem.

Hermed skabes en brugeroplevelse, hvor brugerne har klare valgmuligheder i menuen og kan navigere til de ønskede oplysninger eller handlinger.

Projektsiden



Her beskrives brugerens rejse fra menuen til projektsiden. På projektsiden kan de ansatte se de aktive projekter, de har adgang til. Når en ansat navigerer på projektsiden, har de kun to valgmuligheder. Dette skyldes, at den ansattes rolle ikke tillader dem at ændre deres projekter. De kan kun se de projekter, de er tildelt. De har to valgmuligheder: at vende tilbage til den forrige side, hvor de kan vælge andre handlinger som f.eks. at se deres tildelte moduler, eller de kan vælge at logge helt ud af programmet.

Vi har valgt at have en "Log ud"-knap tilgængelig i hele programmet, da vi mener, det er vigtigt for brugerne at føle, at de kan afslutte sessionen når som helst.

Designet bag vores program

Vores gruppe var enige om vigtigheden af, at brugerne føler sig komfortable, når de benytter vores program. Derfor har vi lagt ekstra vægt på UI/UX (brugergrænseflade/brugeroplevelse) og har grundigt overvejet vores designvalg. Vi mener, at det er afgørende at forstå vores brugere for at kunne skabe god UI/UX.

Vi har bestræbt os på at skabe et intuitivt og brugervenligt interface, der gør det nemt for brugerne at navigere og udføre deres opgaver. Vi har brugt tid på at analysere og forstå vores brugeres behov, præferencer og arbejdsmønstre for at skabe en optimal brugeroplevelse.

Vi tog også Product Owners feedback i betragtning som prioritet når det kom til layout af applikationen.

Hvem er vores brugere?

Brugerne af vores projektstyringsprogram er ansatte hos Alpha-Solutions. Dette omfatter et bredt spektrum af medarbejdere, herunder projektledere, teammedlemmer, og andre interessenter, der er involveret i projektarbejdet.

Hvad vil de gerne have?

Alpha-Solutions ønsker et projektstyringsprogram, der giver dem mulighed for at oprette og administrere fremtidige projekter. Dette program skal give dem værktøjer og funktionaliteter til at planlægge, organisere og styre hvert trin af et projekt, fra start til slut.

Hvilket problem løser vi for dem?

Dette vil være med til at løse de planlægningsproblemer, de har stødt på i fortiden, fordi de ikke har haft adgang til et virtuelt planlægningsprogram. Ved at implementere dette projektstyringsprogram vil Alpha-Solutions kunne forbedre deres evne til at planlægge, organisere og styre deres projekter mere effektivt.

Hvordan reflekterer dette med produktmålet?

Vi mener, at vores produktmål er i overensstemmelse med Alpha-Solutions' ønske om et velfungerende produkt. Vi har fokuseret på kvaliteten af vores program og dets brugervenlighed, så Alpha-Solutions ikke behøver at bruge ekstra ressourcer på at træne deres medarbejdere i brugen af programmet.

Ved at prioritere kvalitet og en intuitiv brugeroplevelse har vi tilstræbt at skabe et produkt, der er nemt at lære og anvende. Vi ønsker at minimere indlæringskurven for ¹³Alpha-Solutions' ansatte, så de hurtigt og effektivt kan integrere programmet i deres daglige arbejdsrutiner.

Forståelse af vores brugere

Profession - trained software designers.

Level - Education is usually something software design.

Keyboard - Users are average or above average on keyboard.

Age - 20+ to 65+

Use frequency - This program will be an integral part of the work users do.

¹³ [What is User Interface \(UI\) Design? | IxDF \(interaction-design.org\)](https://www.interaction-design.org/) - (læst - 22/05 - 2023)

Language - Danish/English

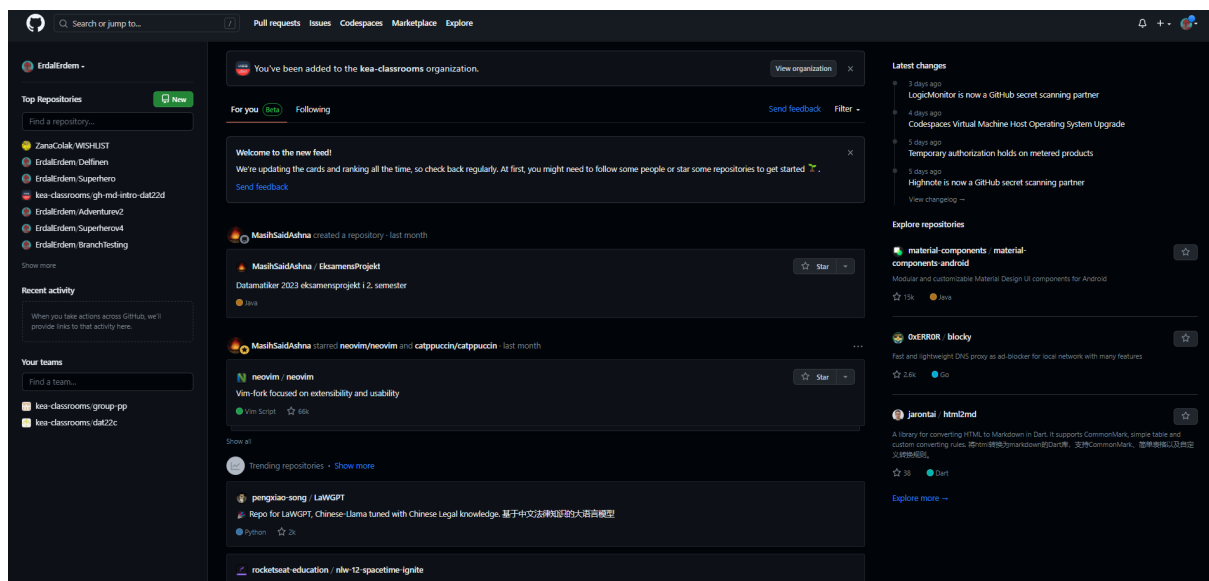
De gyldne regler indenfor UI design

Som nævnt tidligere, har vi valgt at fokusere på programmet's læringskurve, da vi anser det som yderst vigtigt, at vores program er let at lære at bruge. På den måde kan Alpha-solutions undgå at bruge ekstra ressourcer på at oplære deres personale i brugen af programmet. Vores mål er at gøre Alpha-solutions mere effektive, og derfor har vi bestræbt os på at reducere antallet af unødvendige og komplekse interaktioner i programmet, som kan mindske produktiviteten.

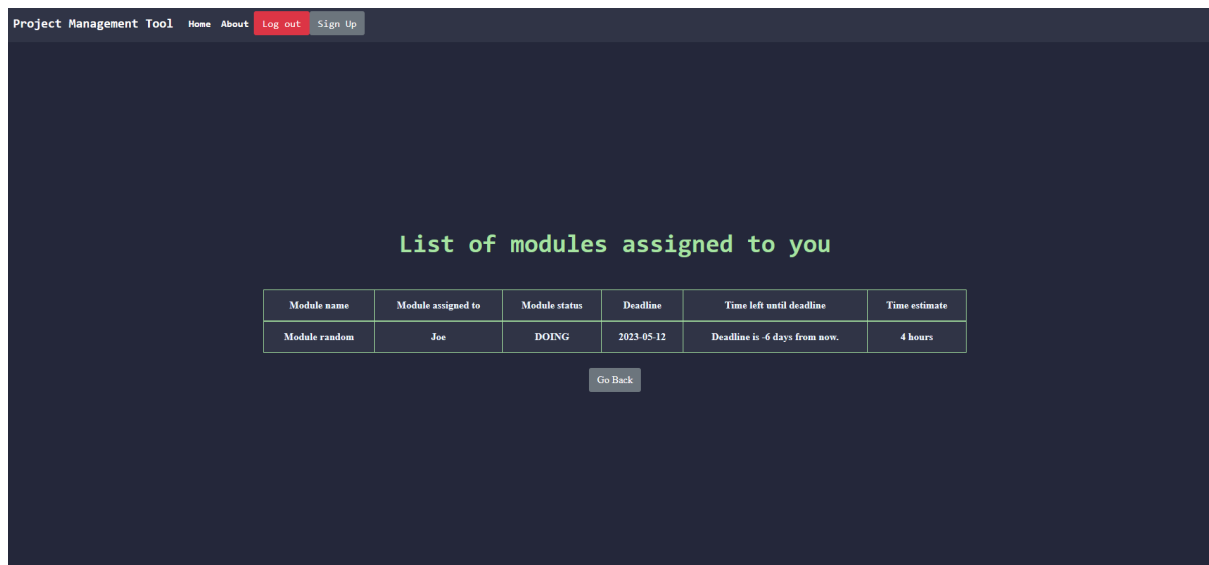
Vi har valgt at begrænse brugerens interaktionsmuligheder til mus og tastatur, da dette er almindeligt kendte og let tilgængelige værktøjer. Ved at holde os til disse standardinputenheder kan vi sikre en mere intuitiv og brugervenlig oplevelse for brugerne. Vi mener, at brugere med en standard computerkendskab vil kunne benytte vores program uden problemer.

Da vores kunder allerede har haft omfattende erfaring med kodning på højt niveau, har vi valgt at fokusere på en brugergrænseflade, som de kan genkende fra deres daglige arbejde. For at opnå dette har vi hentet inspiration fra det velkendte program Github, som er meget populært blandt softwareudviklere. Ved at præsentere en brugergrænseflade, som ligner det, de allerede er fortrolige med, ønsker vi at skabe en sømløs overgang og minimere indlæringskurven for vores brugere.

(Github)

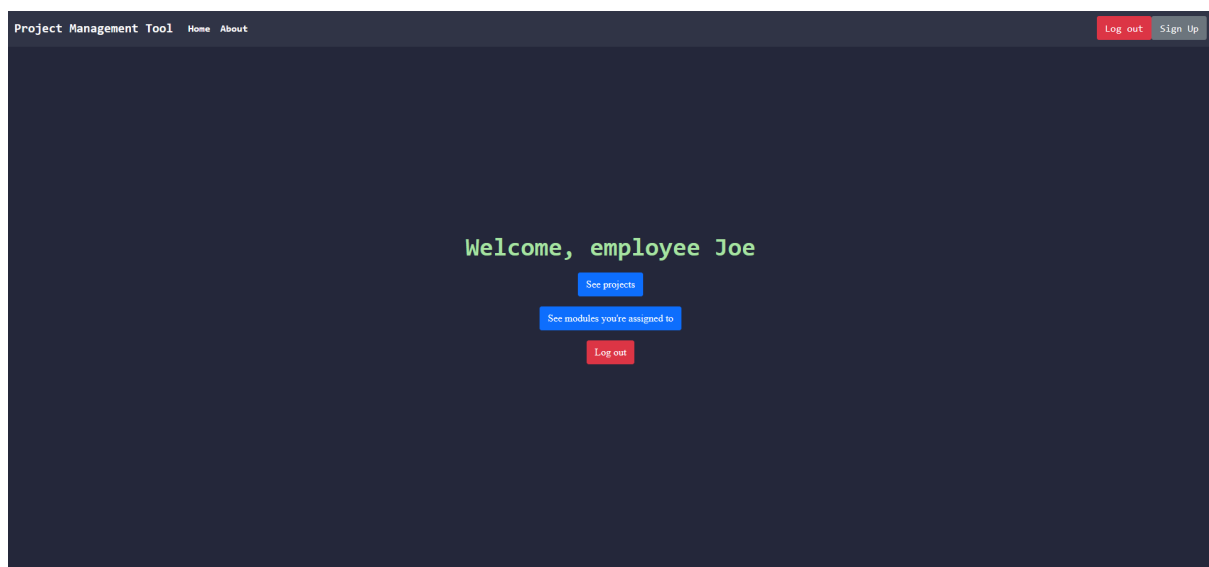


(Vores program)



Vi har også valgt at implementere en mørk æstetik som tema i vores program ligesom Github, da der i mange år har været en trend vedrørende mørke temaer til hjemmesider. Programmører som arbejder om natten vil gavne sig af dette, da der mindskes øjenbelastning. Dark mode er generelt set som en behagelig og skånsom brugeroplevelse blandt mange.

En anden vigtig faktor for os har været at forenkle navigationen i programmet. Vi har opnået dette ved at bruge konsistente knapper og farver gennem hele programmet, så brugerne ikke behøver at bruge tid på at huske, hvor hver enkelt funktion er placeret. Vi har også implementeret visuelt tydelige ikoner, så brugerne nemt kan identificere knappens funktion ved hjælp af ikonets konnotation



Et eksempel på et visuelt signal vi bruger er farven rød som præsentation af "log ud"-knappen. Vi har truffet denne beslutning med henblik på at reducere den kognitive belastning og gøre det nemt for brugeren at identificere og udføre denne handling. Ved at have en konsistent farve for "log ud" på

tværs af forskellige dele af programmet, undgår vi forvirring og sparer borgerne for at skulle bruge tid på at lede efter denne funktion.

Ved at reducere den kognitive belastning gennem konsistens og tydelige visuelle cues bidrager vi til et mere produktivt arbejdsmiljø. Vores brugere behøver ikke at bruge unødvendig tid og mentale ressourcer på at huske, hvor hver enkelt funktion er placeret. Dette frigiver mere tid og energi til at fokusere på de egentlige opgaver og øger deres produktivitet.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
```

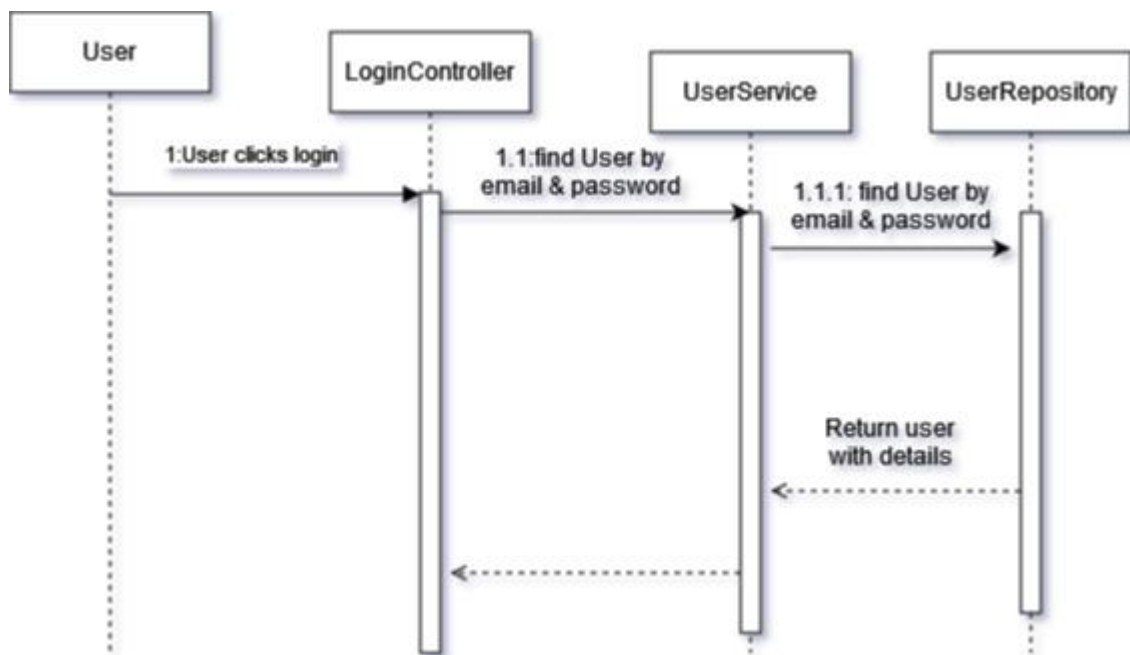
Vi har også valgt at benytte os af Bootstrap, et HTML-framework, i vores program. Dette framework bidrager til brugervenligheden på forskellige enheder, herunder mobilenheder. Selvom programmet primært er designet til at blive brugt på en computer, har vi taget højde for situationer, hvor brugeren måske ikke har adgang til en computer. Ved at anvende Bootstrap kan brugere stadig få en tilfredsstillende brugeroplevelse, selv når de bruger programmet på andre enheder.

Bootstrap hjælper med at sikre, at programmets layout og design tilpasser sig forskellige skærmstørrelser og enheder. Dette sikrer en ensartet og responsiv brugeroplevelse på tværs af forskellige platforme. Selvom vores primære fokus er på computerbrug, erkender vi vigtigheden af at kunne tilbyde programmet på alternative enheder, når det er nødvendigt.

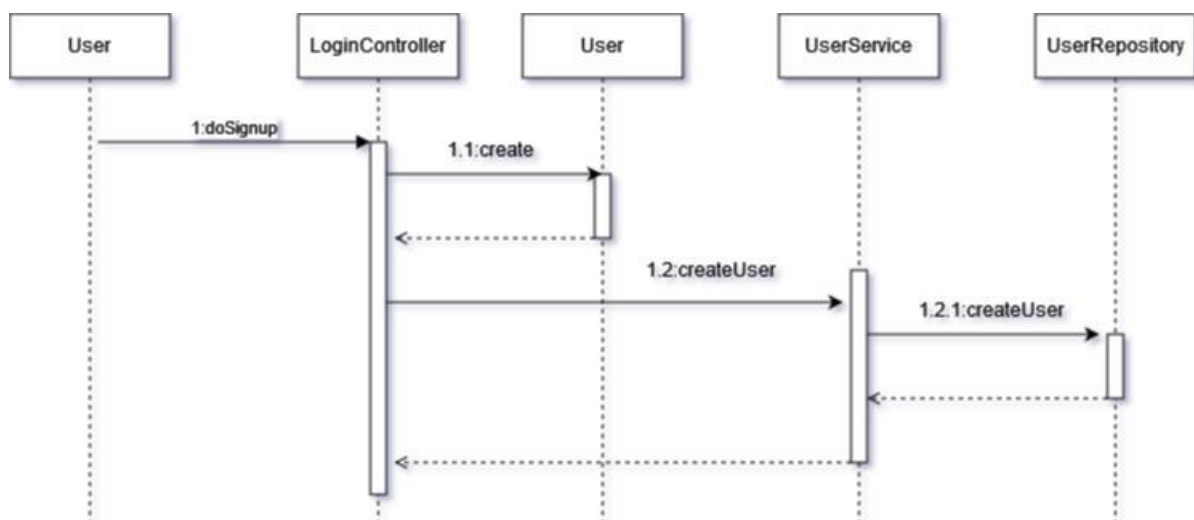
Sekvensdiagrammer

For at give overblik over hvordan programmet handler under brug af en aktør, har vi lavet følgende sekvensdiagrammer for at give en visuel forståelse af hvilke parter der kommunikerer med hinanden under processerne.

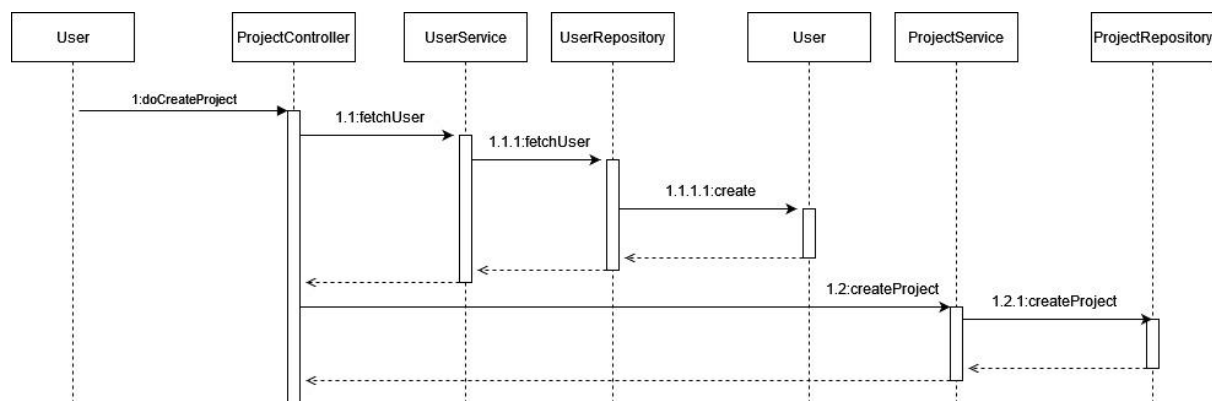
¹⁴ [What is User Interface \(UI\) Design? | IxDF \(interaction-design.org\)](https://www.interaction-design.org/l/learn/article/what-is-user-interface-ui-design?from_the_archive=true) - (læst - 23/05 - 2023)



I dette første diagram vises processen som udføres når brugeren indtaster sine oplysninger på login-siden og klikker “login”. Ligesom langt de fleste hjemmesider skal man indtaste sin adgangskode og email adresse for at logge ind på applikationen. Klassen “LoginController” har en funktion beregnet til login som tager udgangspunkt i de data som brugeren har indtastet og sender dem videre i systemet til klassen “UserService”, som er mellemmand imellem controller- og repository klassen. Det er i “UserRepository” hvor der rent faktisk behandles den indtastede data. Klassen har funktionen “findUserByEmailAndPassword” som skaber forbindelse til databasen og hvis databasen kan finde en bruger med samme email og adgangskode som der blev indtastet, returnerer den et “User” objekt tilbage som indeholder alle detaljerne til brugerens konto. Dette “User” objekt returneres tilbage til “LoginController”, som vil henvise brugeren til profilsiden eller fortælle brugeren at de indtastede oplysninger er forkerte.



I dette andet diagram vises processen der opstår under oprettelse af ny bruger i applikationen. Processen initialiseres så snart at brugeren har indtastet oplysninger til en ny brugerkonto, hvor "LoginController" klassens metode "doSignup" går i gang. Det første som funktionen gør er at skabe et nyt "User" -objekt med alle oplysningerne som blev skrevet på signup-formen. Derefter kaldes "UserService" klassen som så kalder "UserRepository" klassen for at skabe en bruger i databasen med det "User" objekt som blev oprettet tidligere. Repository klassen gemmer alle oplysningerne ind i databasen i "user" tabellen, hvor brugeren vil have sin egen række med ID, navn, adgangskode, e-mail og rolle. Hvis oprettelsen af den nye konto virker uden fejl, vil man som bruger blive omdirigeret til profilsiden. Hvis fejl opstår, forbliver man på signup-formen.



I dette tredje sekvens diagram vises et overblik af processen under oprettelse af et nyt projekt. Diagrammet starter med at brugeren initialiserer processen ved at klikke "create project" efter at have udfyldt en form for projektet. "ProjectController" starter med at først hente brugerens oplysninger, som bindes til et "User" -objekt. Dette kræver kommunikation mellem controller klassen, "UserService" og "UserRepository" klassen. Brugeren hentes i repository klassen ved at kigge efter personen i databasen og ved at skabe et nyt "User" objekt som indeholder alle disse oplysninger. Brugeren returneres således tilbage til controller klassen, hvor programmet kan fortsætte med at skabe projektet. Controller klassen kalder på "ProjectService" klassen som kalder på "ProjectRepository", som endelig vil indsætte dataen indtastet af brugeren ind i databasen som nyt projekt. Efter processen er færdig vil man blive omdirigeret til projektsiden som indeholder en liste af alle projekter i databasen, hvor der vil stå sit eget navn på de projekter man selv har lavet.

Særlige forhold

I vores applikation bruger vi "HttpSession" pakken i vores Java kode til at kunne opbevare brugerens oplysninger når der logges ind. Dette er nyttigt til at kunne finde brugerens ID, som gør at man kan

navigere rundt på siderne f.eks. se sin profil og når man opretter nye projekter eller moduler. Det skal nævnes at der er indstillet en "timer" for en session, så når der er gået 2 minutter uden brugerinput vil "HttpSession" blive invalideret og man bliver logget ud.

I programmet kan man oprette en bruger som er en af typerne: "ADMIN", "MANAGER", "EMPLOYEE". Admin har højest privilegium, da de har fuld adgang til databasen og alle sider. Manager er brugerrollen som kan oprette, slette og opdatere projekter. De kan også udpege en Employee bruger til at stå til ansvar for et modul de skal lave. Employee har lavest privilegium og kan ikke slette, opdatere eller oprette projekter. De har dog en side som viser alle moduler, som de er blevet tildelt af en manager.

Udvalgte kodeeksempler

Under udviklingen af applikationen stødte vi ind på et problem som kan løses på forskellige måder, da "deadline" skulle implementeres under projektklassen i Java ud fra "date" datatypen i SQL databasen. Vi besluttede at "LocalDate" pakken er en passende klasse til at præsentere datoer i vores applikation, da det er et datoformat uden en tidszone i ISO-8601 kalender-systemet som er populært blandt udviklere.

Når der foregår en "read" operation som søgning af et projekt, bliver deadline læst som en string fra SQL som konverteres til "LocalDate" type med "toDate()" funktionen.

```
LocalDate deadline = rs.getDate("deadline").toDate();
```

```
public Project fetchProject(int projectID) {
    try(Connection con = DBManager.getConnection()) {
        String SQL = "SELECT * FROM project WHERE pid = ?";
        PreparedStatement pstmt = con.prepareStatement(SQL);
        pstmt.setInt(1, projectID);
        ResultSet rs = pstmt.executeQuery();
        if (rs.next()) {
            int uid = rs.getInt("uid");
            String projectName = rs.getString("project_name");
            String projectCreator = rs.getString("project_creator");
            LocalDate deadline = rs.getDate("deadline").toDate();
            int timeEstimate = rs.getInt("time_estimate");
            return new Project(projectID, uid, projectName, projectCreator, deadline, timeEstimate);
        }
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
    return null;
}
```

Når der oprettes et nyt projekt vil der således bruges en “toString()” funktion til at konvertere deadline til en string datatype, og ved hjælp af en indbygget funktion i MySQL “STR_TO_DATE(?, '%Y-%m-%d')” konverteres strengen til “date” datatypen.

```
String SQL = "INSERT INTO project (project_name, project_creator, deadline, uid) VALUES (?, ?,  
STR_TO_DATE(?, '%Y-%m-%d'), (SELECT uid FROM user WHERE name = ?));";
```

Ovenstående kode eksempel stammer fra “createProject” funktionen i “ProjectRepository” klassen, som ses i det følgende:

```
public boolean createProject(User user, Project project){  
    try(Connection con = DBManager.getConnection()) {  
        String SQL = "INSERT INTO project (project_name, project_creator, deadline, uid) VALUES (?, ?, STR_TO_DATE(?, '%Y-%m-%d'), (SELECT uid FROM user WHERE name = ?));";  
        PreparedStatement pstmt = con.prepareStatement(SQL);  
        pstmt.setString(1, project.getProjectName());  
        pstmt.setString(2, user.getUserName());  
        pstmt.setString(3, project.getDeadline().toString());  
        pstmt.setString(4, user.getUserName());  
        pstmt.execute();  
        return true;  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```

```
public String getTimePeriod(){  
    String years = timePeriod.getYears() == 1 ? timePeriod.getYears() + " year" : timePeriod.getYears() + " years";  
    String months = timePeriod.getMonths() == 1 ? timePeriod.getMonths() + " month" : timePeriod.getMonths() + " months";  
    String days = timePeriod.getDays() == 1 ? timePeriod.getDays() + " day" : timePeriod.getDays() + " days";  
    String retval = String.format("Deadline is %s, %s, %s from now.",  
        years, months, days);  
    if (retval.contains("0 years,")){  
        retval = retval.replace("0 years,", "");  
    }  
    if (retval.contains("0 months,")){  
        retval = retval.replace("0 months,", "");  
    }  
    return retval;  
}
```

I ovenstående kode eksempel vises en funktion som returnerer et projekts tidsperiode mellem projektets deadline og datoen fra nu lokal tid. Funktionen går ud på at formatere strengen til at være mere læsevenlig og fylde mindre inde på projektsidens tabel. Koden tjekker om tidsperioden har 1 eller 0 antal i “years”, “months” og “days”, og ændrer strengen henholdsvis. For eksempel vil den ændre; “0 years, 1 months, 3 days” til “1 month, 3 days”.

Vi har desuden brugt “enum”, for eksempel i “User” og “Module” -klasserne. I “User” klassen bruges dette til at vælge hvilken type bruger man er. Der er 3 typer: “ADMIN”, “MANAGER”, “EMPLOYEE”.

```
public enum Role {  
    ADMIN,  
    MANAGER,  
    EMPLOYEE
```

I “UserRepository” klassen krævede det ekstra arbejde til at få SQL og Java koden til at kunne forstå hinanden og arbejde sammen. Til læsning af brugere bruger vi en indbygget funktion “valueOf” i et “ResultSet” til at matche en værdi på en string til en konstant i “enumerate”.

```
User.Role role = User.Role.valueOf(rs.getString("role"));
```

Under oprettelse af en ny bruger konverteres “role” attributen til en string som MySQL kan forstå uden problemer.

```
pstmt.setString(5, user.getRole().toString());
```

Kodeeksempel af “createUser” i “UserRepository”:

```
public boolean createUser(User user){
    try(Connection con = DBManager.getConnection()) {
        String SQL = "INSERT INTO user(vid, name, password, email, role) VALUES(?, ?, ?, ?, ?)";
        PreparedStatement pstmt = con.prepareStatement(SQL);
        pstmt.setInt( parameterIndex: 1, user.getUserID());
        pstmt.setString( parameterIndex: 2, user.getUserName());
        pstmt.setString( parameterIndex: 3, user.getPassword());
        pstmt.setString( parameterIndex: 4, user.getEmail());
        pstmt.setString( parameterIndex: 5, user.getRole().toString());
        pstmt.execute();
        return true;
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
```

Vores sidste udvalgte kode eksempler stammer fra vores database kode i MySQL. Koden omhandler “project_creator” rækken i “project” tabellen, hvor der står en brugers navn på et projekt som de har oprettet. Når “project” tabellen først laves under opbygning af databasen vil feltet være tomt, fordi værdien opdateres først senere hen i koden af SQL scriptet:

```
UPDATE project JOIN user SET project_creator = user.name WHERE project.uid = user.uid;
```

Dette statement opdaterer “project_creator” til at være lig “name” fra “user” tabellen hvor begge tabeller har samme bruger-ID. Dette var den eneste løsning vi kunne umiddelbart finde for at kunne opdatere feltet når databasen bygges.

I samme stil opdateres også “time_estimate” i “project” tabellen til at være summen af alle modulers “time_estimate” under samme projekt-ID:

```
UPDATE project SET project.time_estimate = (SELECT SUM(module.time_estimate) FROM module WHERE module.pid = project.pid);
```

Tests

I vores arbejdsproces havde vi kun brugt manuelle tests til at undersøge om programmet virker korrekt. Det vil sige, at man kørte programmet, klikkede på knapper, indtastede oplysninger og så videre for at teste applikationen. Dette er selvfølgelig en ineffektiv proces, men på den anden side blev der på denne måde testet for alle klasserne; controllers, services, repositories, models osv.

I programmet er der blevet testet for repository klasserne UserRepository, ProjectRepository og ModuleRepository med unit tests. Disse blev skrevet efter at størstedelen af applikation funktionerne var implementeret.

Funktionerne i repository klasserne er designet på den måde at alle statements og branches er dækket når funktionen kaldes, ellers kastes en exception hvis fejl opstår.

UserRepository

| UserRepositoryTest | CRUD | Dækningsgrad |
|--------------------|------------------|--------------|
| fetchUserTest | Read operation | 100% |
| getUsersTest | Read operation | 100% |
| createUserTest | Create operation | 100% |
| editUserTest | Update operation | 100% |
| deleteUserTest | Delete operation | 100% |

FetchUserTest er en test på funktionen fetchUser indeni UserRepository klassen. Den returnerer et User objekt ved at søge efter en User med et specifikt ID som er unikt til brugeren. Den er som nævnt en metode som udfører read operation da funktionen bruges til søgning af en bruger i databasen.

Dækningsgraden er 100%, da så længe der ikke opstår en fejl med forbindelse til databasen eller der gives et ugyldigt ID, vil programmet køre igennem alle statements og branches i funktionen når den skal finde brugeren. I testen oprettes et vilkårligt User objekt som har samme indhold som en User der allerede eksisterer i databasen som hentes med fetchUser og herefter sammenlignes indholdet af objekterne for at sammenligne.

GetUsersTest er en test som er meget ens med fetchUser metoden. Forskellen er at getUsersTest vil returnere en liste af alle users i databasen fremfor bare en enkelt user. Strukturen af getUsers metoden er opbygget på samme måde som fetchUser, så dækningsgraden er den samme. I testen oprettes en række users som har samme indhold som users der allerede er hardcoded i databasen og disse users

tilføjes så til 2 lister hver for sig og sammenlignes herefter. Sammenligningen sker i et while loop som itererer gennem de 2 lister samtidigt.

CreateUserTest er testen som undersøger om createUser metoden fungerer som den skal.

Dækningsgraden er også 100% da programmet kører hele funktionens kode igennem så længe ingen exceptions opstår. Testen udføres ved at oprette et vilkårligt User objekt med attributter og herefter kaldes createUser metoden til at gemme dette objekt til databasen. Herefter kaldes fetchUser metoden til samme user ID for at få et andet user objekt at sammenligne med. En svaghed ved denne type test er, at man selv skal holde øje med at det user ID man giver User objektet ikke har et ID som allerede er blevet brugt. Dette er kun et problem her i tests, for da når man bruger selve applikationen, vil der selv blive tildelt et ID til det nye User i databasen automatisk og man skal ikke selv indsætte et ID.

EditUserTest er testen som undersøger editUser funktionen for fejl. Dækningsgraden er også 100% på grund af samme struktur som de andre metoder. Der benyttes annotationen “@Order()” for editUserTest metoden da bliver testet på et user objekt som blev oprettet af createUserTest. Der vil ellers opstå fejl, da der ikke kan redigeres på en user som ikke eksisterer endnu. Testen går ud på at bruger editUser metoden på det user vi oprettede i createUserTest og ændre navnet. Herefter bruges fetchUser til at søge efter den nævnte user og sammenligne userens navn med den præcise streng, vi ændrede navnet til.

DeleteUserTest er den sidste test for UserRepository som tester for om deleteUser metoden faktisk sletter en user fra databasen. Da denne test afhænger af useren som blev oprettet af createUserTest er den også annoteret med “@Order()”, som den sidste test i rækkefølgen. Funktionen er meget simpel med 100% dækningsgrad, da den blot søger efter en user i databasen med et specifikt user ID og derefter sletter den fra databasen. Som sagt sletter den brugeren som blev oprettet i createUserTest via dens unikke user ID og herefter bruges en indbygget Junit funktion “assertNull” som tjekker om et objekt er null. Denne metode bruges på et fetchUser kald på samme ID som den user vi arbejdede med tidligere. Den undersøger grundlæggende om user objektet er blevet null efter vi har kaldt deleteUser.

Til UserRepository testene er der desuden blevet lavet en hjælpefunktion til at kunne sammenligne 2 User objekters indhold med hinanden. Dette var nødvendigt da den indbyggede JUnit funktion “assertEquals” ikke blot sammenlignede objekt indhold men også hukommelses adresser. Dette blev hurtigt problematisk fordi i disse tests oprettes der nye User objekter som har samme indhold som et User der oprettes fra UserRepository’s fetchUser metode. De to objekter har identisk indhold, men de er ikke identiske objekter.

ProjectRepository

| ProjectRepositoryTest | CRUD | Dækningsgrad |
|-----------------------|------------------|--------------|
| fetchProjectTest | Read operation | 100% |
| getProjectsTest | Read operation | 100% |
| createProjectTest | Create operation | 100% |

| | | |
|-------------------|------------------|------|
| editProjectTest | Update operation | 100% |
| deleteProjectTest | Delete operation | 100% |

Testene i ProjectRepositoryTest klassen er struktureret på samme måde som UserRepositoryTest bortset for at der i databasen håndteres projekt ID anderledes i projekt tabellen end user. Når et nyt projekt oprettes, vil der tildeles et automatisk inkrementeret ID til det nye projekt. Problemer opstår når der kaldes deleteProject metoden i test filen. Dette er fordi at værdien for det automatisk inkrementeret ID beholder værdier for selv slettede projekter. Dette betyder at selv hvis det seneste oprettede projekt har ID nummer 4, og det næste projekt har ID 5 som bliver slettet, vil det næste oprettede projekt få nummer 6. På grund af dette, var det nødvendigt at lave en hjælpefunktion som nulstiller den automatiske tæller til det forrige projekts ID nummer hvis en deletion operation sker.

ModuleRepository

| ModuleRepositoryTest | CRUD | Dækningsgrad |
|----------------------|------------------|--------------|
| fetchModuleTest | Read operation | 100% |
| getModulesTest | Read operation | 100% |
| createModuleTest | Create operation | 100% |
| editModuleTest | Update operation | 100% |
| deleteModuleTest | Delete operation | 100% |

ModuleRepositoryTest er struktureret på samme måde som ProjectRepositoryTest med den eneste forskel, at moduler beholdes inde under projekter i lister. Klassen har desuden også samme hjælpefunktioner bortset for at der ændres i modul tabellen i databasen frem for projekt.

Konklusion

Vores projektperiode startede med at identificere og undersøge, hvordan vores gruppe kunne tackle denne store opgave, der blev stillet med fokus på at opfylde Alpha-solutions' ønsker og krav til et projektstyringsprogram.

Derfor valgte vores gruppe først at udarbejde en plan for resten af projektperioden for at se, om vores endelige mål var realistiske. Dette gjorde vi ved at udføre en proaktiv analyse, der inkluderede mange effektive analyser, som gav indsigt i de potentielle risici, som gruppen kunne støde på. Vi anvendte også en feasibility study, som skulle danne grundlag for den endelige konklusion om, hvorvidt vores projekt havde grønt lys til at blive igangsat.

I gennemførelsesfasen benyttede vi et Scrum-board, hvor vi nedskrev vores krav både fra os selv og kunden ved hjælp af user stories. Disse user stories udgjorde grundlaget for udviklingen af vores applikation og dens fremtidige features, der skulle inkluderes.

En af de primære og væsentligste features, som vi skulle implementere, var CRUD-funktionaliteten. Denne funktion var afgørende, da den muliggjorde mange brugervenlige features, som typisk er inkluderet i et projektstyringsprogram. Uden CRUD ville vores program være ubrugelig i næsten alle tilfælde.

Vi gjorde brug af SQL til at opretholde alle brugerdata, hvilket gjorde dem let tilgængelige og håndterbare i Java koden.

Derefter deployed vi vores applikation online ved hjælp af Docker og Render. Dette valg blev truffet for at give vores brugere mulighed for at anvende programmet online uden behov for at downloade og installere ekstra software.

Som konklusion er vores projektkalkulationsværktøj som vi har lavet under projektperioden tilfredsstillende og velfungerende. Vi nåede frem til denne konklusion, da vores gruppe var enige om, at samarbejdet gennem hele perioden havde været effektivt, og alle havde været fokuserede og engagerede i processen. Vi mente også, at vores fokus på kommunikation havde været en væsentlig faktor for vores succes, da det var den afgørende årsag til, at vi kunne opnå vores forskellige mål i løbet af hele perioden.

Vores applikation blev færdiggjort med en tilfredsstillende mængde af funktioner og et flot design. Vores gruppe er tilfreds med det endelige produkt, taget i betragtning af at holdet kun bestod af to personer. Vi formåede at udarbejde en veldefineret rapport, som inkluderede de krav, der var blevet stillet. Derfor konkluderede vi som gruppe en succesfuld projektperiode.

Litteraturliste

Zwisler, Lars. (2005). 2022-Zwisler, Projekt og analyseredskaber. I: C. Straaberg (Red.), *Projekt og analyseredskaber* (s. 1-23). Forlaget Globe A/S. <https://bibliotek.dk/da/work/870970-basis%3A25941438>

Feasibility Study and Its Importance in Project Management. (2023, 8 . Maj). *Simplilearn*. Lokaliseret den 11. Maj 2023 på <https://www.simplilearn.com/feasibility-study-article>

UDEN MENNESKER LØSER TEKNOLOGIEN INGENTING. (s.d.). *Alpha-solutions*.
https://www.alpha-solutions.com/da/om-os?gclid=Cj0KCQjw4NujBhC5ARIsAF4Iv6fiM2V5TG3LvvNzlVs_7CdFsUS562pRHmp-6HnvGZLm_8Fd0HYcPSIaAnKGEALw_wcB

Anonym. (s.d.). User Interface (UI) Design. *interaction-design*.
<https://www.interaction-design.org/literature/topics/ui-design>

Anonym. (s.d.). Coupling (computer programming). I: *Wikipedia*.
[https://en.wikipedia.org/wiki/Coupling_\(computer_programming\)](https://en.wikipedia.org/wiki/Coupling_(computer_programming))

What does 'low in coupling and high in cohesion' mean. (s.d.). *Stackoverflow*.
<https://stackoverflow.com/questions/14000762/what-does-low-in-coupling-and-high-in-cohesion-mean>

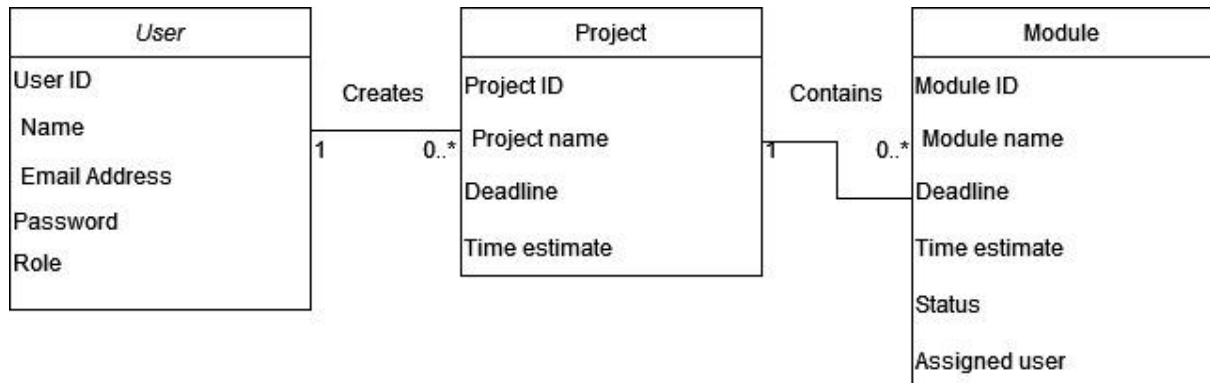
SQL. (s.d.). I: *Wikipedia*. <https://en.wikipedia.org/wiki/SQL>

Thymeleaf. (s.d.). I: *Wikipedia*. <https://en.wikipedia.org/wiki/Thymeleaf>

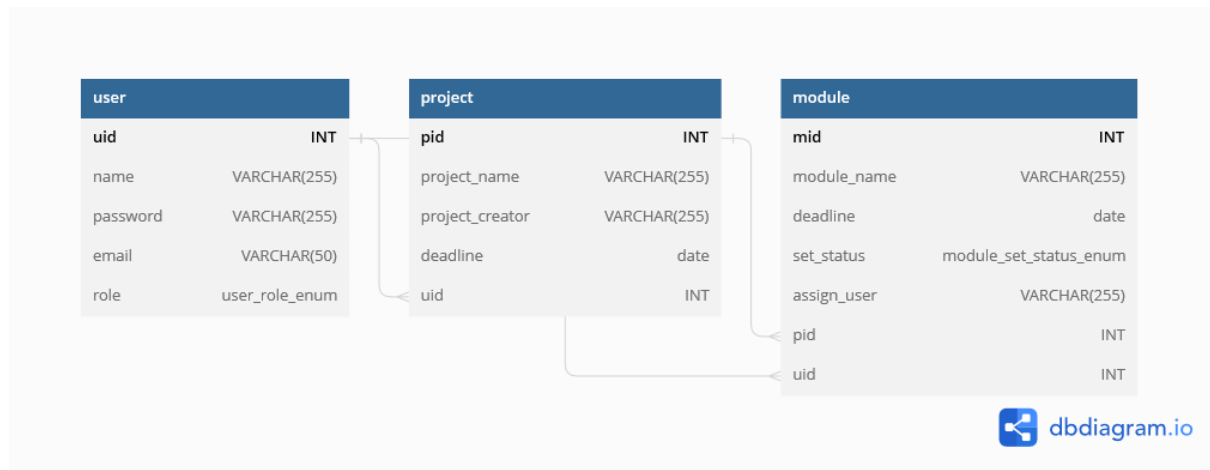
Database normalization: *Wikipedia*. https://en.wikipedia.org/wiki/Database_normalization

Bilag

Domænemodel:



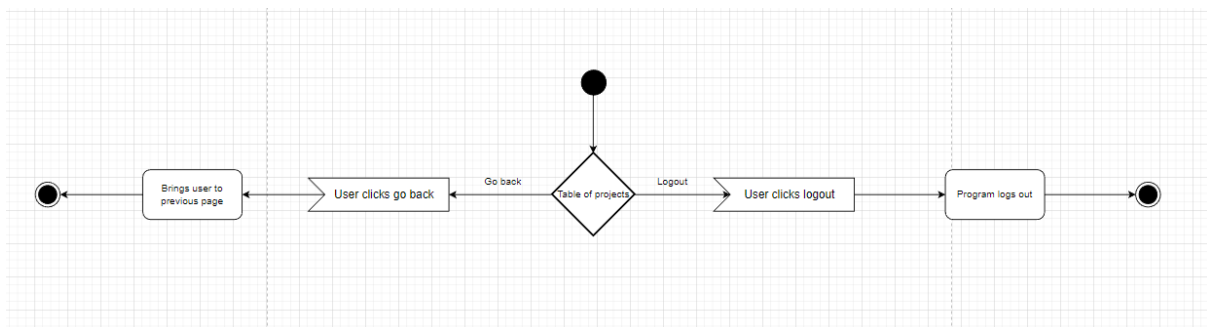
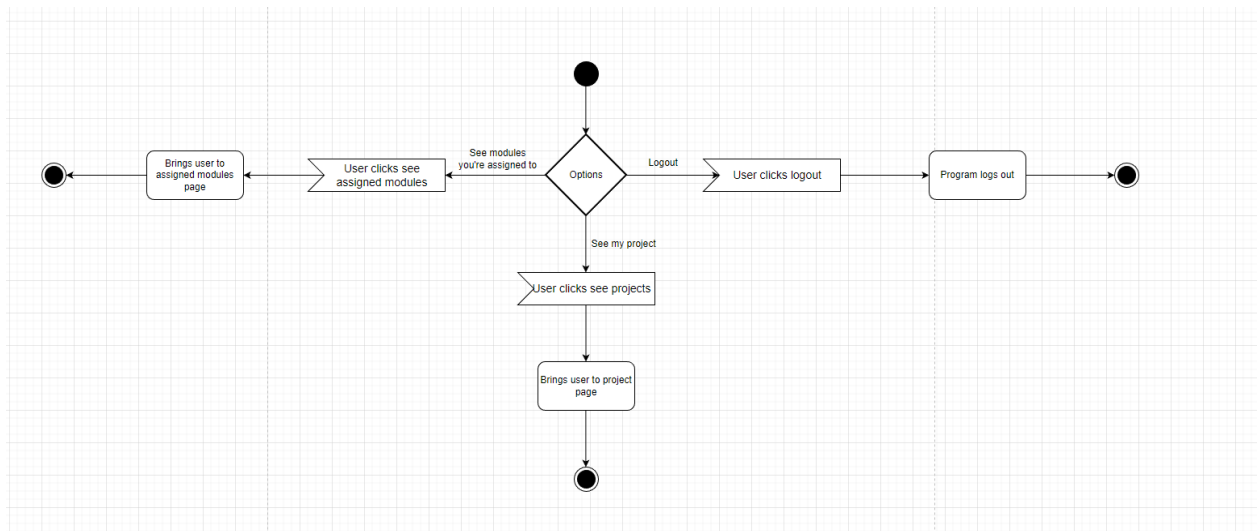
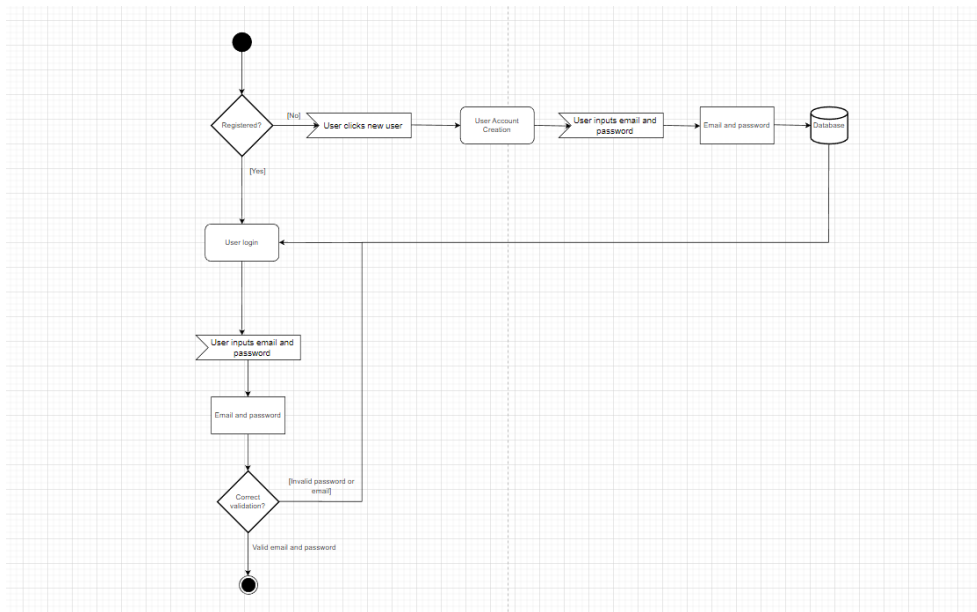
ER-diagram:



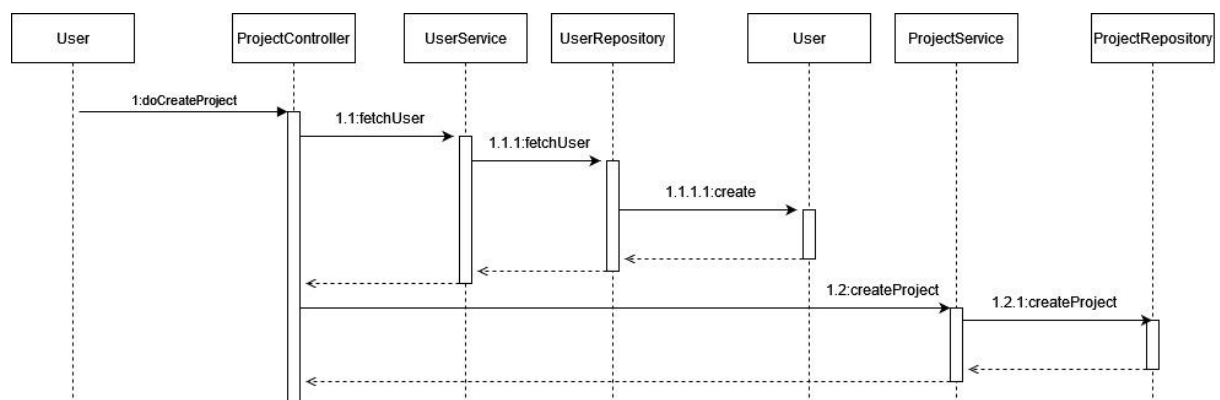
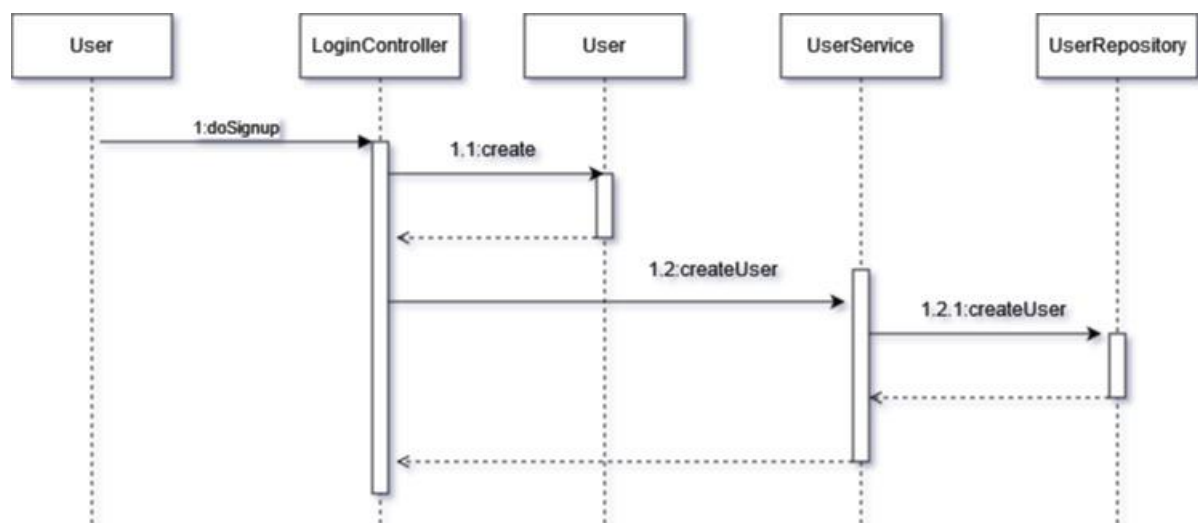
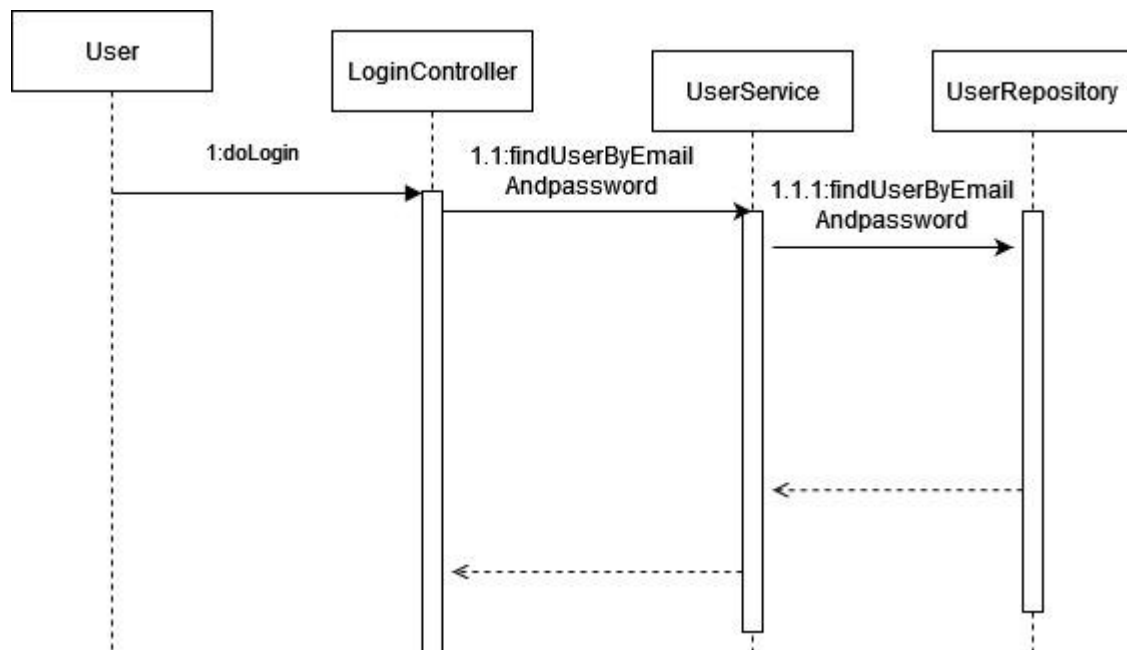
UML diagram



Aktivitetsdiagrammer:



Sekvensdiagrammer:



Github repository link

<https://github.com/MasihSaidAshna/EksamensProjekt>

Github navne:

MasihSaidAshna

ErdalErdem