****

**Webprosjekt – Våren 2019**

|  |  |
| --- | --- |
| **Emnekode og emnenavn:** | **PRO101 Webprosjekt** |
| **Innleveringsdato:** | **7.06.2019** |
| **Antall sider:** |  |
| **Antall ord:** |  |
| **Gruppenummer:** | **Gruppe 31** |
| **Studentnavn:** | **Studentnummer:** |
| **Jonas Say** | **705351** |
| **Fredrik Holanger** | **705745** |
| **Fredrik Jansen** | **705571** |
| **Benjamin Opsal** | **706019** |
| **Sindre Fromeide** | **705599** |
| **Studentene bekrefter at de har gjort seg kjent med, og fulgt, retningslinjer for intellektuell redelighet.** | |

Innhold

[**1 Innledning 3**](#_Toc10717142)

[**1.1 Fokus til Gruppe 31 3**](#_Toc10717143)

[**1.2 Prosessen til ferdig produkt 4**](#_Toc10717144)

[**2 Idé og konsept 4**](#_Toc10717145)

[**3 Utviklingsmetodikk 5**](#_Toc10717146)

[**4.1 Bruk 6**](#_Toc10717147)

[**4.2 Funksjonalitet 6**](#_Toc10717148)

[**4.3 Teknologi 6**](#_Toc10717149)

[**4.3.1 HTML 7**](#_Toc10717150)

[**4.3.2 CSS 7**](#_Toc10717151)

[**4.3.3 JavaScript 7**](#_Toc10717152)

[**4.4 KILDEKODE 8**](#_Toc10717153)

[**5 Målgruppen, Usability, designprinsipper, universell utforming etc. 9**](#_Toc10717154)

[**6 Tekniske valg, JavaScript og Git 10**](#_Toc10717155)

[**7 Referanser 11**](#_Toc10717156)

# Innledning

På eksamen i faget PRO101 har vi fått i oppgave å lage en Kanban-lignende løsning som vi selv kunne brukt under utførelsen av eksamensoppgaven. Løsningen skal gå ut på å kunne strukturere prosjektarbeid over nettet slik at det er aksessibelt for alle medlemmer, samt tilby et GUI som tilfredsstiller usability-, UX og enkelhets-prinsippene. I tillegg til prinsippene skulle det også tilfredsstille kravene som stod til prosjektet. Kanban er et begrep som stammer fra Japan, og betyr i bunn og grunn «tavle». I dagens bruk vil Kanban referere til diverse verktøy som ofte assisterer gruppearbeid, både i liten og stor skala. For å gå i detalj så kan man si at Kanban er et verktøy som brukes for å strukturere oppgaver, fjerne uklarheter, vise progresjon og mye mer. Det er veldig populært og ofte brukt i smidige prosjekter, altså prosjekter der oppgaver blir lagt til og evaluert underveis. Det gjør det mer strukturert å kunne se progresjon på arbeidet man holder på med, samtidig kunne bruke mye av funksjonalitetene som Kanban tilbyr. Dette kan være alt fra å tildele oppgaver, gjøre noen gruppemedlemmer ansvarlige for en oppgave, men bunnlinjen er å kunne visuelt fremstille progresjon og for å kunne enklere kartlegge hvor lang tid forskjellige oppgaver bruker. Det gjør det enkelt får prosjektleder å tildele nye oppgaver, estimere tid og gi raske tilbakemeldinger til kunde.

## Fokus til Gruppe 31

Hovedfokuset til Gruppe 31 har vært å skape et verktøy som er slik at enhver bruker kan gå inn og finne ut av hvordan ting funker uten å måtte gjennomgå en «how to». Løsningen vår ble døpt «Erigo», som betyr effektivitet på Latin. Det ironiske bakteppet til oppgaven er at vi har brukt Trello, som er et smidig gruppearbeidsverktøy, noe som har gitt oss en fin pekepinn på hva vi skal og ikke skal gjøre i vår løsning. Videre har vi utnyttet flere verktøy for å sikre at vi har fått et så solid produkt som mulig. Disse verktøyene inkluderer FaceBook gruppechat, Github, Git, Google Docs og andre diverse arbeidsteknikker.

## Prosessen til ferdig produkt

Produktet vi har endt opp med har vært i linje med det vi har sett for oss og enda mer. Den tilbyr kun det man trenger slik at man slipper å tvinge seg gjennom en tettpakket nettside fylt med ønskede funksjonaliteter. Ved oppstart så hadde vi visse mål vi ønsket å nå, og vi føler at de to ukene vi hadde ble effektivt brukt til å sile ut idéer som var realistiske og ikke. Da vi starter å lage nettsiden begynte ting å gå på skinner. Vi la fort merke til at ting ble gjort veldig effektivt, takket være det høye kunnskapsnivået i gruppen. Dermed var det smertefritt for oss å løse problemer og komme med nye forslag til elementer vi kunne ha i nettløsningen vår. Selve utviklingsprosessen vil forklares mer i detalj lenger ned i rapporten.

# Idé og konsept

Idéen vi hadde helt i starten var å lage noe som var tilrettelagt programmerere/IT-grupper. Det å kunne tilby et smidig gruppeverktøy virket helt brilliant, men vi konkluderte raskt med at GitHub allerede gjør denne jobben veldig effektivt med Git. Videre ønsket vi å lage et verktøy som var slik at man kunne legge ut kodesnutter slik at andre medlemmer i gruppen kunne endre på dette. Etter å ha jobbet med nettsiden i et par timer kom vi på at det bare ville vært en tungvinn metode og mindre effektivt enn å bare fortsette med Git.

Til slutt klarte vi å lande på et konsept som vi syns er fint for allmenn bruk i stedet for å være innsnevret på IT-grupper. **Intuitivitet** endte opp med å bli hovedordet vi la vekt på. Med intuitivitet mener vi at verktøyet skal være enkelt for enhver bruker å sette seg inn i og bli flink til å anvende det i løpet av noen få minutter med utforsking. Verktøyet skal være tilgjengelig for enhver gruppe som ønsker å jobbe smidig, om det måtte være IT-grupper eller litt mindre teknologisk begavede grupper, derav **intuitivitet.**

Listen ble satt høyt da vi iverksatte idémyldringen, og vi kom fort frem til at vi ønsket «state» i prototypen vår. Med dette mener vi at vi ønsker å koble dette opp mot en database og bruke cookies til smålagring slik at den husker hvilken tilstand siden var i når man går ut av den. Dette var en omfattende oppgave å sette opp en database for å utfylle state-kravet vi hadde satt for oss selv. I utviklingen fant vi ut etter en hel kveld med testing at det var et mye mer utfordrende oppgave enn det vi trodde, og dermed ble den idéen satt på park til vi ble ferdig med hovedproduktet først. Etter hovedkravene (og ekstrakravene) var utfylt satte vi oss ned igjen og prøvde oss på Firebase igjen, og klarte til slutt å lykkes. Vi tenkte først å kunne tilby real-time endringer i tillegg til å huske state hvis man går inn og ut av forskjellige prosjekter. Dette var enda en utfordring, men vi endte opp med å kun bruke FireBase til å holde state i stedet for å fokusere så mye på real-time endringer. Det er for øvrig lagt til rette i koden at man skal kunne koble opp mot database med minimale endringer. Dette var det neste steget i løsningen, men ble nedstemt ettersom at det var tidskrevende.

# Utviklingsmetodikk

Ved oppstart av prosjektet satt vi oss ned og først gjorde oss kjent med hverandre. Det å ha en gruppe som utfyller hverandre er viktig, derfor gikk vi gjennom styrker og svakheter med oss selv ved en veldig kort selvpresentasjon. Deretter hadde vi gode runder med idémyldring, og brukte CPS til en viss grad for å lokalisere «problemet» og løste det.

Under hele prosessen fra vi fikk oppgaven til vi ble ferdige med den var vi enige om at Git skulle brukes effektivt. For å mimikere hvordan dagens bedrifter jobber bestemte vi oss for å fordele oppgaver i Trello og jobbe på hver vår maskin og dele kode via Git. Vi ble også enige om at vi skulle lage egne brancher i Git, gjøre vårt, deretter oppdatere Master når vi ble ferdige med den delen. I tillegg til Trello og Git lagde vi en FaceBook-gruppe der vi holdt kontakt gjennom hele prosjektet. Vi føler FaceBook har hjulpet svært mye da det er så tilgjengelig og enkelt å kommunisere der.

Store deler av prosjektet har foregått på skolen, men mye har også vært hjemmearbeid. Det å sitte sammen har gjort det smertefritt å foreslå idéer og diskutere i stedet for å ta det over nettet. Samtidig som vi har brukt Trello så syns vi det også er viktig å fysisk jobbe sammen i grupper. Når vi har kommet med nye idéer og forslag som vi mener skal være en prioritet har vi satt det opp i Trello og satt en frist, selv om frister ikke er ideelt for best utfall, og jobbet mot det til dagen etter. Mye av tiden vi har brukt i grupper har gått til teknikker som ikke hadde hatt like godt utbytte over nettet som i virkeligheten.

1. **PROGRAMVARE & PROTOTYPE BESKRIVELSE**

## Bruk

Gruppe 31 har i forbindelse med eksamen i faget PRO101 utviklet en web-applikasjon (prototype) i henhold til eksamenskriterier. Prosjektet finnes i sin helhet slik det ble levert på: https://github.com/FrikoPro/Gruppe31. Alle endringer som er utført etter 07/06/2019 klokken 15:00 inngår ikke i den leverte eksamensoppgaven.

## Funksjonalitet

Prototypen er et lettvekts prosjektstyringsverktøy. Applikasjonen simulerer en klassisk Kanban tavle der arbeidsoppgaver deles opp og plasseres inn i kolonner. Kolonnene illustrerer visuelt hvor i prosessen en arbeidsoppgave til enhver tid befinner seg. Applikasjonen skal fungere som et prosjekteringsverktøy som gir et helhetlig bilde av fremdrift og arbeidsoppgaver. Applikasjonen er tiltenkt mindre prosjekter der bruker ikke har tid eller ønsker å sette seg inn i et nytt prosjektstyringsverktøy. Brukergrensesnitt, ikoner og plassering av objekter følger generelle normer for god GUI-design. Applikasjonen er generell, ryddig og kan settes opp på under 5 minutter.

## Teknologi

Applikasjonen tar i bruk HTML (html5), CSS og JavaScript. Det er ikke tatt i bruk rammeverk utenfor det som regnes som standard i de respektive teknologiene. Det er fire HTML filer som sammen danner grunnlaget for applikasjonen. Disse har hvert sin CSS fil i tillegg til en felles CSS-fil som inneholder stiler som går igjen.

### HTML

Html benytter html5 syntax og head med en felles gjennomgående struktur: html -> head -> body -> main (content) -> footer

### CSS

Applikasjonen tar i bruk CSS-Grid og er skalerbar ned mot, men ikke helt ned til mobile enheter. Css-grid er i brukt for å plassere og gi egenskaper til større områder av grensesnittet. Css-flex er brukt på mindre områder. For å enkelt skape et gjennomgående design benytter applikasjonen en felles CSS-fil “common.css” i tillegg til en egen dedikert CSS-fil. Denne “common.css” inneholder felles stiler og gjør det enkelt å endre felles trekk i applikasjonen. Alle klassene i denne filen starter med bokstavene “cm” for common. For eksempel vil “cm-button” gi et html element en felles stil og animasjon.

### JavaScript

Applikasjonen benytter JavaScript til å utføre en mengde oppgaver. Mange av HTML elementene genereres dynamisk ved hjelp av JavaScript når brukeren utfører handling i appen. Hver hovedside, login.html, dashboard.html og project.html har sin egen dedikerte JavaScript-fil. I tillegg benytter applikasjonen egne JavaScript-filer som inneholder funksjonalitet som benyttes på tvers av prototypen. Ulike objekter inneholder ulik statisk og dynamisk data som kan hentes ut på tvers av siden.

Applikasjons-tilstand og overføring av data internt: Prototypen benytter “cookies” og “localStorage” for å simulere tilstand. Både “localStorage” og “cookies” benyttes til å lagre brukere. Prototypen benytter pr. dags. dato ikke noen form for database eksternt. Det er lagt til rette for slik funksjonalitet uten større endringer i applikasjonens struktur.

## KILDEKODE

Her finnes tekniske detaljer utover det som er kommenter i koden. Se kommentarer i de respektive filene hvis du ikke finner dokumentasjon her. I dette dokumentet omtales kun filer og funksjoner av en uvanlig art.

##### Filstruktur

* gruppe31\_app (mappe) ..\* css (mappe) ..\* img (mappe) ..\* js (mappe)
  + dashboard.html
  + landing.html
  + project.html
  + welcome.html

Strøm: welcome -> landing -> dashboard -> project. CSS (Mappe):

###### common.css

Inneholder en rekke felles klasser som skrives på formen cm-. Eksempel: “cm-grid” er en klasse som utvider “grid”. Her finnes også prosjektspesifikke klasser som ikke er skrevet på “cm” formatet, men som benyttes flere steder i applikasjonen. dashboard.css, landing.css, project.css, landing.css og welcome.css: Disse filene hører til html-filene med samme navn og leses fra topp til bunn. Samtlige filer starter med html,body og ender med en footer (hvis dette finnes på siden). Den respektive html-filen vil gi et inntrykk av hvor i css filen en klasse befinner seg. Ønsker du å finne et element der klasse-navnet starter med “cm” finnes dette i common.css.

JS (Mappe):

###### cookies.js

Denne filen inneholder funksjonene getCookie(cname) og setCookie(cname, cvalue, exyears). Disse brukes til å hente ut data og sette data inn i cookie strengen.

###### darkmode.js

Endrer og styler ulike elementer for å lage “dakmode”. Lagrer også dette i cookie.

# Målgruppen, Usability, designprinsipper, universell utforming etc.

Da vi startet på prosjektet tenkte vi først og fremst at målgruppen skulle være prosjekter som har mye programmering i seg, men vi kom fort på at GitHub egentlig gjør den jobben for oss, men at det hadde vært effektivt om dette var integrert i webløsningen. Naturligvis blir dette vanskelig da vi antakelig må ha lisenser og diverse tillatelser for å samarbeide med GitHub, så vi endret målgruppe til oss selv. I oppgaveteksten står det at vi skal lage en løsning som vi selv ville brukt under denne eksamen, da falt det oss naturlig inn å bruke oss selv som målgruppe da vi vil best vite hva vi er og ikke er ute etter i en slik løsning. Samtidig som vi har satt oss selv som målgruppe så har vi også tatt med i betraktning at det skal være tilgjengelig for alle som driver med gruppearbeid og ønsker å gjøre det smidig. Med det mener vi at vi ønsker å tilby dette verktøyet til alle, uten å sette noe særlig krav til teknologisk forståelse.

Videre så vektlegger vi intuitivitet i løsningen vår, da dette er et viktig prinsipp når det kommer til brukertesting. Målet er at brukere skal oppleve prototypen vår som enkel og selvforklarende. Vi har nøye gått gjennom usability-prinsippene for å oppnå mest mulig brukervennlig løsning. I tillegg til det som står oppgitt i PowerPointene har vi også lagt inn egne momenter som vi mener er veldig greie å ha, som f.eks en historikk over eldre prosjekter som er ferdige, beroligende design og bl.a. info når man hover’er over visse knapper. For å oppnå best mulig prototype har vi nøye gått gjennom hvert steg av usability, enkelhet og UX-prinsippene på prototypen vår. Mye av fokuset har ligget på det at en bruker kommer inn til nettsiden og skjønner fort hva man skal gjøre for å navigere seg videre. Vi har prøvd å eliminerer mest mulig knapper for å holde det enkelt, en samtidig tilby funksjonalitet som forventes (og enda mer) i et slikt prosjektplanleggingsverktøy. Når det kommer til usability har vi passet på å få med de viktigste prinsippene som er: god design, brukskvalitet, funksjonalitet og tilbakemeldinger (Bakke, 2019). For oss er det veldig viktig at verktøyet er lett å forstå og krever lite forkunnskap for å kunne operere det. Dette føler vi at vi har oppnådd på en svært greit måte da vi har eliminert ekstraknapper, slått sammen noen funksjoner slik at det blir minst mulig redundans.

Når det kommer til UX (User eXperience) er det slik at nettsiden skal møte ens forventninger. Dvs. at det skal være selvforklarende at når man eks. klikker på «logg inn» at man går videre til neste side i stedet for at man bare blir logget inn og må selv navigere seg videre. Det er også en viktig del av det å få brukere til å like nettsiden. Hvis noe ikke faller i smak med en gang så er det mindre sjanse for at man bruker det igjen, derimot hvis alt er som forventet og nettsiden er veldig enkel (men fortsatt tilbyr funksjonaliteten man trenger) så er det mye lettere for brukeren å anbefale videre og bruke senere. I vår løsning har vi prøvd å oppnå en «avslappende» atmosfære med rolig bakgrunn, frosted elementer som skal gjøre at det ikke er altfor «på» brukeren, men heller mer laid-back. Som nevnt ovenfor har vi tatt i bruk CSS grid til utformingen av nettsiden, noe som har gjort utformingen av elementene i høyest grad smerteløst å endre på. Nettsiden er veldig lys og kan virke tærende på øynene om det ses på over lenger tid, derfor har vi introdusert en «night mode» øverst på nettsiden. Dette omgjør hele nettsiden til en mørkere versjon som er mye mildere for øynene å se på, og gjør det mer behagelig å jobbe med verktøyet vårt over lenger tid.

# Tekniske valg, JavaScript og Git

Vi har valgt å bruke en IDE vi selv er komfortable med i stedet for å tvinge oss til å lære en felles IDE. Noen har brukt WebStorm, SubLime, Brackets, Vistual Studios, etc.. Det var felles enighet om at vi skulle prioritere produkt ovenfor IDE, derfor konkluderte vi med at vi skulle bruke hva vi selv var komfortable med. I tillegg har vi har valgt å bruke ren HTML, CSS og JavaScript. Vi har brukt noen koder som er fra rammeverk, men hovedfokuset har ligget på ren HTML, CSS og JS.

Git og GitHub ble brukt veldig flittig, og det med god grunn. GitHub brukte vi for å ha et repository der vi hadde koden vår, og Git til å interagere med GitHub. Vi valgte for øvrig å bruke cmd-Git i stedet for GitBash, da vi kom frem til at vi ville jobbe så likt som mulig slik ordentlige selskaper (prosjekter) gjør. Noen utfordringer ved Git var at det kun var to-tre som hadde mye erfaring med Git, og dermed måtte vi ha flere gjennomganger, i tillegg til å se på tutorial-videoer, med gruppen.

# Problemstillinger & løsninger

I utviklingsprosessen til et produkt med en deadline er det uunngåelig å ikke møte på problemstillinger. Ved en problemstilling så må gruppen effektivt jobbe sammen for å finne løsninger på dette for å komme seg videre og møte deadlinen. Gruppen vår var ingen unntak, og møtte på en moderat mengde problemer som vi har løst. Måten vi fant ut av disse problemene på var først og fremst at gruppen testet nettløsningen for å se om det var noen umiddelbare bugs eller direkte feil som måtte fikses. Deretter lot vi fire forskjellige personer med ulik IT-bakgrunn teste nettsiden vår i underkant av 10 minutter. Begge prosessene var svært lærerike, og vi fikk nyttig feedback fra brukertestene.

Det var både større og mindre problemstillinger vi ble satt foran, og de fleste har blitt løst. Nevneverdige problemer er først og fremst FireBase. Til tross for at det ikke var et krav eller et «ekstrakrav» så ønsket vi å få til litt backend i løsningen vår. Grunnen til dette er for å få en helhetlig følelse og dermed bedre gjennomførbarhet i nettsiden. Ved en database i løsningen så kunne vi laget unike brukere som ville blitt lagret i tillegg til tilstand på nettsiden i forhold til hvilken bruker som er innlogget. Dette hadde gitt nettsiden en mye mer profesjonell følelse. Problemstillingen var at vi ikke hadde tilstrekkelig med tid til å fullføre FireBase-oppsettet. Det ble estimert at vi behøvde i underkant av tre dager til for å fullføre det. Det er vel å merke tilrettelagt for å sette det opp, men som tidligere nevnt så var det tidsrammen som hindret oss.

Det andre store problemet vi ble stilt ovenfor var brukerne på dashbordet. Når man appenda en ball til et prosjekt så ville userID som ble tildelt i prosjektet forbli det samme på de andre også. Dette ville si at hvis en ball skulle bli appenda til flere prosjekter så beholdt den samme userID. Vi prøvde å slette et medlem fra et prosjekt, men siden userID’en på ballen var det samme som i andre prosjekter så ble alle ballene slettet. Dette skapte et problem og stridde imot nøkkelordet, **intuitivitet,** som vi jobbet mot. Etter diskusjoner og testing kom vi til slutt frem til en løsning. SKRIV INN LØSNING HER

Andre mindre problemer var ved drag & drop-funksjonen inne på prosjektsiden. Her var det slik at hvis det var eksempelvis to kort i hver sone, altså to i «todo», «doing» og «done», så ville det oppstå et problem om man prøvde å dra et kort fra f.eks. «doing» til «done». Det var satt som default at hvis et kort ble sluppet av oppå et annet kort så ville ingenting skje, og da måtte man sikte på den tynne kanten i sonene for at det skulle fungere slik det skulle. Dette ble løst ved å sette z-index’en til alle kortene (unntatt det som blir dratt) til en indeks **bak** sonen når et kort blir dratt. Med denne løsningen så ville ikke kortene lenger utføre default-funksjonen selv om det tilsynelatende ble sluppet oppå et annet kort.

Medlemsballene utgjorde heldigvis en liten andel av problemstillingene, men et problem vi møtte på var at når man skulle dra disse elementene over på kortene for å appende til kort/prosjekt så ville ballene forsvinne fra hyllen øverst på nettsiden. Dette ble enkelt løst ved å klone nodene i stedet for å flytte på dem. Dette løste også problemet med at hvis man ønsket å slette en ball så ville ikke ballen bli slettet fra hyllen, men kun der man dro den fra.

# Referanser

Bakke, S. (2019). PRO101\_2\_Usability\_2019.pdf. Oslo, Norge.

PowerPoint-slides fra forelesning (Sturla Bakke)

Dribbble.com