|  |
| --- |
| Universitetet i Sørøst-Norge  Fakultet for ØIT  Institutt for IT og informasjonssystemer  –  Prosjektoppgave  **30/04/2019**  **Applikasjonsutvikling for internett/APP2000B-1 18H** |
| Christoffer Johansen (216735), Kristian Olsen(216718), Thomas Hodnebrog(216720) og Andreas Fætten(152058).  **Exizt nettbutikk**  Prosjektoppgave for APP2000B-1 18H Applikasjonsutvikling for internett |
|  |

# Sammendrag

Hensikten med prosjektet var at vi skulle lære oss å lage en fungerende webapplikasjon ved hjelp av verktøy og metoder vi lærte i APP200B-1 kurset. Første semester lærte vi hvordan vi skulle lage applikasjoner ved hjelp av php, javascript, jquery og ajax. Det var først i andre semester vi begynte å lage selve applikasjonen, fra januar frem til april. Vi valgte å lage en nettbutikk for moteklær som vår applikasjon. Grunnen til at vi valgte dette var fordi vi følte det var stort nok omfang, og vi tenkte ut et par funksjoner som kunne gjøre applikasjonen vår litt mer unik i forhold til lignende applikasjoner som allerede finnes. Bortsett fra litt komplikasjoner i starten, følte vi at prosjektet gikk ganske bra, og vi endte opp med å komme i mål med alt bortsett fra handlekurv funksjonen. Den ble litt vrien å få til, så vi endte opp med å hardkode en løsning slik at vi fikk mulighet til å teste resten av funksjonene i applikasjonen. Ellers føler vi at vi har en fungerende applikasjon som vi er ganske fornøyd med selv.

Innholdsfortegnelse

[1 Sammendrag 2](#_Toc7515607)

[2 Administrativ del 5](#_Toc7515608)

[2.1 Beskrivelse av webapplikasjon 5](#_Toc7515609)

[2.2 Prosjektmetode, prosjektorganisering, roller og ansvarsfordeling 5](#_Toc7515610)

[2.2.1 Prosjektmetode 5](#_Toc7515611)

[2.2.2 Prosjektorganisering 5](#_Toc7515612)

[2.2.3 Roller 6](#_Toc7515613)

[2.2.4 Ansvarsfordeling 6](#_Toc7515614)

[2.3 Litteratur og internett 6](#_Toc7515615)

[2.3.1 Litteratur 6](#_Toc7515616)

[2.4 Prosjektplan og prosjektlogg 6](#_Toc7515617)

[2.4.1 Scrum 6](#_Toc7515618)

[2.4.2 Prosjektlogg 17](#_Toc7515619)

[2.5 Refleksjonsnotat 18](#_Toc7515620)

[2.5.1 Hva gikk bra? 18](#_Toc7515621)

[2.5.2 Hva gikk dårlig? 18](#_Toc7515622)

[2.5.3 Hva burde blitt gjort annerledes? 19](#_Toc7515623)

[2.5.4 Veien videre 19](#_Toc7515624)

[3 Faglig del 21](#_Toc7515625)

[3.1 Kravspesifikasjon 21](#_Toc7515626)

[3.1.1 Use cases 21](#_Toc7515627)

[3.2 Systembeskrivelse 23](#_Toc7515628)

[3.2.1 Overordnet beskrivelse 23](#_Toc7515629)

[3.2.2 Brukergrensesnitt 23](#_Toc7515630)

[3.2.3 Klientsiden 24](#_Toc7515631)

[3.2.4 Tjenersiden 25](#_Toc7515632)

[3.2.5 Databasen 26](#_Toc7515633)

[3.3 Installasjon 27](#_Toc7515634)

[3.4 Testrapport 27](#_Toc7515635)

[3.4.1 Testplan 27](#_Toc7515636)

[3.4.2 Funksjonalitet testing 28](#_Toc7515637)

[3.4.3 Brukervennlighet 28](#_Toc7515638)

[3.4.4 Kompatibilitet 28](#_Toc7515639)

[3.4.5 Testrapporter 28](#_Toc7515640)

[3.5 Kilder og referanser 29](#_Toc7515641)

[3.5.1 Bøker 29](#_Toc7515642)

[3.5.2 Nettressurser 30](#_Toc7515643)

[3.5.3 Programbiblioteker 30](#_Toc7515644)

[3.5.4 Ekstra informasjon om kilder 30](#_Toc7515645)

# Administrativ del

## Beskrivelse av webapplikasjon

Applikasjonen vi har valgt å lage er en nettbutikk for klær. Poenget med applikasjonen er at folk flest skal få mulighet til å kjøpe klær ganske billig på nett, og i tillegg få mulighet til å få en del avslag på klær ved hjelp av visse funksjoner vi skal forklare mer senere. Applikasjonen skal også ha et design som gjør at det er lett tilgjengelig og lett for alle å bruke.

* Nye brukere skal kunne opprette en brukerprofil på applikasjonen for å logge inn, eksisterende brukere skal kunne logge inn når de vil.
* Brukere skal kunne endre opplysningene på brukerprofilen sin, f.eks. navn og e-post.
* Brukere skal kunne verve venner som nye brukere av applikasjonen. Hvis den vervede brukeren kjøper noe i butikken, skal den som verver få en rabatt-kupong til deres neste kjøp i butikken.
* Brukere skal kunne ha mulighet til å delta maksimalt en gang på en månedlig quiz for å være med i trekningen av å vinne gratis premier, f.eks. klær.
* Brukere skal kunne søke og filtrere på ønskede klesmerker, farger, størrelser og pris.
* Brukere skal kunne legge til klær de ønsker å kjøpe i en handlekurv som viser pris på enkelte klesplagg og totalpris.
* Applikasjonen skal være designet responsivt slik at den kan brukes på mobil, nettbrett og pc.

## Prosjektmetode, prosjektorganisering, roller og ansvarsfordeling

### Prosjektmetode

Som prosjektmetode brukte vi den agile metoden Scrum. Grunnen til at vi brukte denne metoden er fordi vi er alle kjent med denne metoden da vi brukte denne metoden i vårt første prosjekt. Metoden fungerte bra fordi vi ble enige om hvem som skulle ha hvilke oppgaver, og når deadlines skulle settes. Metoden fikk også folk til å møte sammen for å jobbe sammen på prosjektet. Metoden gjør at vi kan sette opp oppgaver på et kanban brett, på dette kanban brettet kan alle se hva man skal gjøre. Hvis ikke noen av disse oppgavene blir gjort så tar man de videre til neste sprint. Etter hver sprint har vi også retrospektive møter som hjelper å vise oss hva vi kanskje gjør feil, og hva vi kan endre til neste gang.

### Prosjektorganisering

prosjektets startfase jobbet vi mest individuelt, med et møte i uken. Vi så etter hvert at vi måtte omorganisere med oftere oppmøte hvis vi skulle få fremgang på prosjektet. Det ble da 2 møter den første uken, og deretter 3 ganger i uken, onsdag, torsdag og fredag. Vi jobbet i gjennomsnitt 4-5 timer daglig.

### Roller

Rollene vi brukte var frontend/design, backend og databaseansvarlig.

**Frontend:** Kristian, Thomas

**Design/Photoshop:** Thomas

**Backend:** Christoffer, Andreas

**Databaseansvarlig:** Andreas

### Ansvarsfordeling

Først når gruppen ble grunnlagt, så var Markus Scrum Master (Ansvar for scrum teamet). Christoffer var en hjelpende hånd for Scrum Masteren. Etter Markus droppa ut av gruppen ble Christoffer satt som Scrum Masteren. Scrum masteren er personen som setter opp brukerhistorier som skal bli gjort i løpet av en sprint. Når en sprint er ferdig så vil scrum masteren sette opp et retrospektivt møte for å høre hvordan alle syntes det gikk. Scrum masteren tar da ansvaret for å skrive alt dette ned i en sprint rapport. Ellers har alle like mye ansvar for å gjøre oppgavene de skal.

## Litteratur og internett

### Litteratur

Henrik Kniberg & Mattias Skarin. (2010). Kanban and Scrum making the most of both.

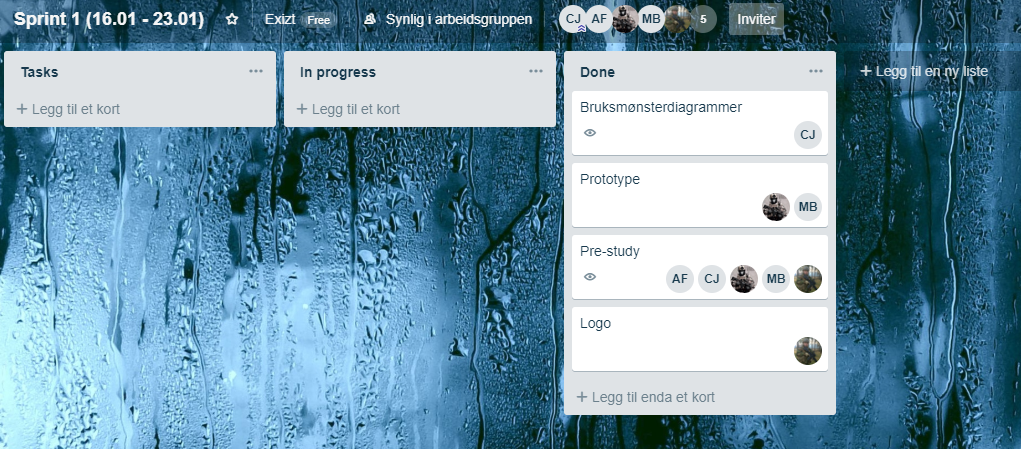
## Prosjektplan og prosjektlogg

### Scrum

Første møtet ble holdt 11.01, da døpte vi butikken til Exizt. Møtets varighet var 45 min.

#### **Sprint 1(16.01 🡪 23.01)**

Onsdag 16.01 satt vi et møte, der scrum master(Markus) og Christoffer satte opp oppgaver som skulle bli gjort i løpet av sprinten. Disse oppgavene var å lese seg opp på relevant kompetanse for å lage applikasjonen, f.eks. php eller bootstrap. Logo, bruksmønsterdiagrammer og første prototype var også noen av disse oppgavene.



Figur 1 Sprint brett 1

**Retrospektivt møte sprint 1 (23.01.19)**

**Kristian** og **Markus**:

Ferdige med prototypen av indeks siden, laget også en midlertidig indeks side. Vi begynte smått å opprette mappe strukturen. Utenom det så ble resten av tiden brukt til å lære seg litt om css grid. Gjennomførte alt på sprinten.

**Christoffer**:

Lærte seg litt om hvordan vi skal implementere og bruke php, om hvordan data skal bli lagret etter bruker registrering, også i forhold til innlogging. Gjennomførte alt på sprinten.

**Thomas**:

Laget forskjellige logoer for en avstemning, hvilken av de som blir valgt skjer i neste sprint.

**Andreas**:

Hadde ingen oppgaver denne sprinten

Første sprinten var kort med få gjøremål, og vi gjennomførte alle oppgavene nevnt ovenfor. I neste sprint som startet 23/01 begynte vi med å bygge siden.



Figur 2 Første prototype av siden på mobil

#### **Sprint 2(23.01 🡪 06.01)**

Onsdag 23.01 hadde vi alle sammen et møte hvor vi ble enige om oppgaver som skulle gjøres i sprint 2. Dette møtet varte i 25 minutter.

**Daglig sprint-møte (29.01)**

Vi hadde et daglig sprint-møte den 29.01 om hvordan sprinten hadde gått og hva vi skulle gjøre videre. Vi tok en gjennomgang av milepæl 3 og milepæl4, etter vi hadde fått tilbakemelding av Bjørn. Vi hadde kommet i gang med forsiden og registrerings-siden og CSS-stilen. Databasen trengte litt små-endringer, og tok det på møtet.

**Oppgavene som var gjort og skulle gjøres:**

**Markus**: Lagt inn alt av html, og skal legge inn CSS-stilen senere i sprinten.

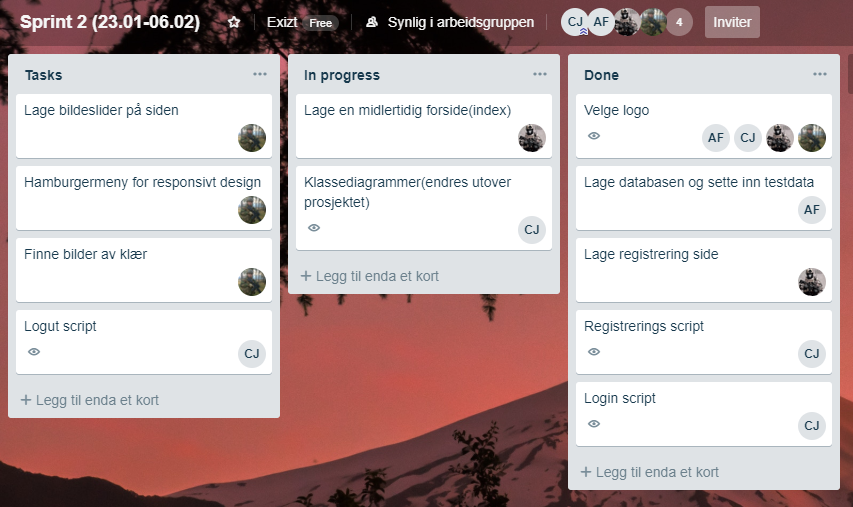
**Andreas**: Endret på databasen og opprettet den.

**Thomas:** Vi ble enige om logoen han hadde laget. Så skulle han legge inn javascript når mer ble lagt til på forsiden.

**Kristian:** Jobbet med protypen av siden, skulle fortsette på header delen av siden og jobbe med forsiden.

**Christoffer:** Jobbet med php scripts på siden, skulle implementeres når formsene ble ferdig.

Trello brettet etter vi var ferdig med sprint 2



Figur 3 Sprint brett 2

**Retrospektivt møte sprint 2(14.02)**

**Andreas:** Ferdig ble ferdig med databasen, og laget noen triggere til databasen. Fremover skulle han se på hva vi trengte av tabeller når det kom til quiz og verve funksjonene.

**Kristian:** Jobbet fortsatt med forsiden, footer og header. På dette tidspunktet hadde vi ikke blitt helt enige om design.

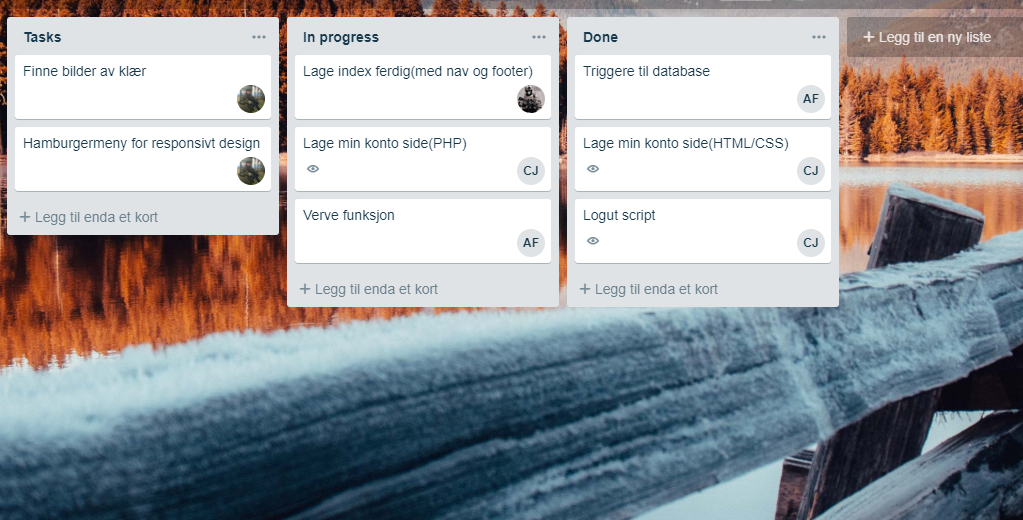
**Christoffer:** Ble ferdig med registreringsscript og log-in script.

Det oppstod litt komplikasjoner innenfor gruppen. En gikk ut av gruppen, det var han som hadde satt seg dypt inn i CSS-grid. Det var det vi brukte for å lage design av siden. Vi vurderte på dette møtet å bytte til bootstrap.

Noe vi fokuserte på til neste sprint var å ha mye mer oversikt over hva vi skulle gjøre. Vi «skal» møtes hver torsdag og fredag fremover for å få gjort så mye sammen som mulig. Vi mente denne løsningen ville gjøre prosjektet mye lettere å gjennomføre når vi ble en mann mindre på gruppen.

#### **Sprint 3(14.02-21.02)**

På denne sprinten ville vi lage noen papir prototyper av sidene for å få mer oversikt over hvilke funksjoner vi skulle implementere.

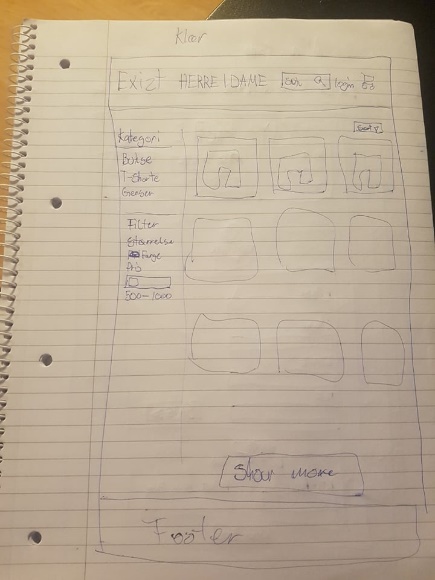
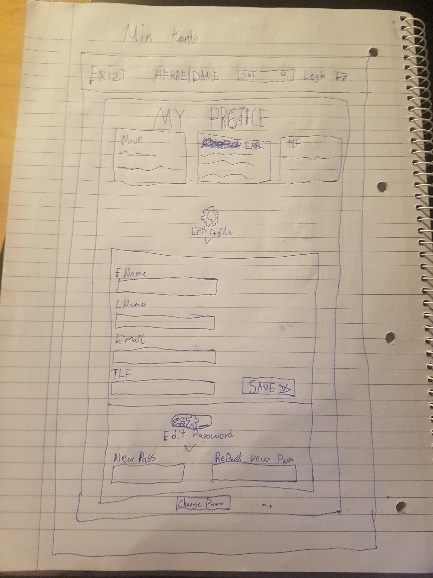


Figur 4 Sprint brett 3

**Retrospektivt møte sprint 3 (06.03)**

Vi hadde en ganske liten sprint fordi vi satte av tid til de obligatoriske oppgavene som skulle bli gjort i andre fag, det var grunnen til at det retrospektive møtet ble så sent som det ble. På denne sprinten ble vi enige om at vi skulle bruke bootstrap istedenfor CSS-grid. Det ble litt mindre oppgaver å gjøre på frontend, og vi kunne da fokusere mer på backend.

Vi skisserte design av sidene litt mer så vi så hva vi trengte av funksjoner. Etter denne sprinten hadde vi mer oversikt over hva vi måtte gjøre på nettsiden enn vi hadde tidligere.



Figur 5 Papir utgaver av profil og klessiden

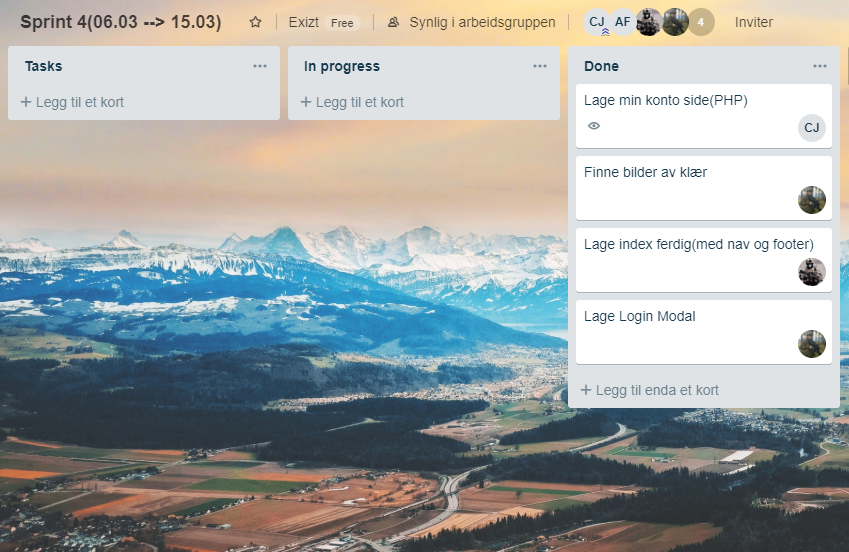
**Christoffer:** Lagde ferdig design på «min profil» siden. Startet litt på backend.

**Kristian:** Gjorde slik at forsiden brukte bootstrap istedenfor CSS-grid. Fortsatt litt småting igjen.

**Andreas:** Tok seg av noen triggere på databasen, begynte også med en verve funksjon.

#### **Sprint 4(14.02 🡪 21.02)**

På denne sprinten satte vi i gang med profilsiden og litt frontend finpussing for å få siden fin.



Figur 6 Sprint brett 4

**Retrospektivt møte sprint 4(21.02)**

Denne sprinten varte i litt over en uke. Vi ville få ferdig mye frontend og få ferdig to viktige backend funksjoner. Det var da snakk om verving og quiz funksjonene. Disse funksjonene var litt mer vanskelig en vi forutså, så vi måtte sette de over til neste sprint også. Siden prosjektfremføringen var rett rundt hjørnet, så sa vi at vi «skulle» få de ferdig neste sprint. Hvis ikke måtte vi ta disse funksjonene ut av nettsiden, ellers fikk vi gjort mye denne sprinten.

**Kristian:** Forsiden ble ferdig, nesten helt ferdig med klessiden(frontend).

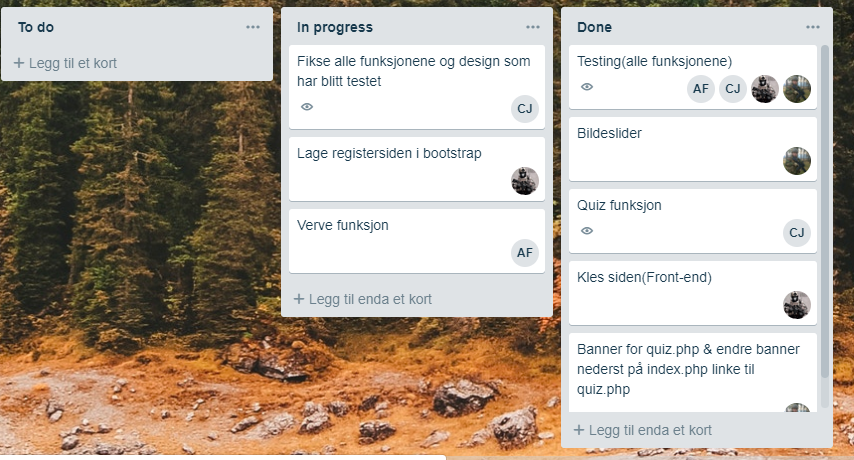
**Andreas:** Satte fortsatt fast på vervingsfunksjonen.

**Thomas:** Fant bilder som vi skulle bruke på nettsiden og laget en log-in modal.

**Christoffer:** Laget profilside (backend funskjoner som: update info og update password). Begynte på quiz funksjonen, og laget ekstra database tabeller.

#### **Sprint 5 (15.03 🡪 22.03)**

Denne sprinten satte vi fokus på å få testet litt. Vi så litt på hva siden manglet ut ifra det vi hadde satt av. Vi så også på det vi i utgangspunktet hadde av brukerhistorier, og hva vi hadde planlagt applikasjonen skulle være. Vi innså at vi kanskje hadde siktet litt for høyt, og vi måtte dessverre ta vekk noen av funksjonene vi i utgangspunktet hadde planlagt. Funksjonene vi fjernet var å legge til favoritter av klesplagg, og betalingsdelen. Så denne sprinten har bare vært masse finpussing på applikasjonen, inkludert funksjonene vi prøvde å få på plass.



Figur 7 Sprint brett 5

**Retrospektivt møte sprint 5(22.03)**

Denne sprinten fikk vi gjort ganske mye. Verve funksjonen som vi hadde veldig lyst til å få på plass var nesten i mål. Men for å få testet den ordentlig trengte vi handlekurv siden til å funke. Siden kles siden kun hadde fått select-spørringer på klær, og ingen funksjoner for å legge dem i handlekurven enda, så var det en del som måtte gjøres. Neste uke skulle vi sette i gang med dette. Ellers var det mye finpuss på applikasjonen, og vi testet funksjoner. Vi testet brukergrensesnittet, og sjekket om det var intuitivt for brukere. Derfor ble det laget litt feilmeldinger. Siden ingen i gruppen hadde satt seg inn i noe jquery/AJAX, så bestemte Christoffer seg for å lage det med PHP.

**Christoffer:** Laget quiz funksjonen. Fikset på det som ble testet på siden. Noen punkter som fortsatt mangler.

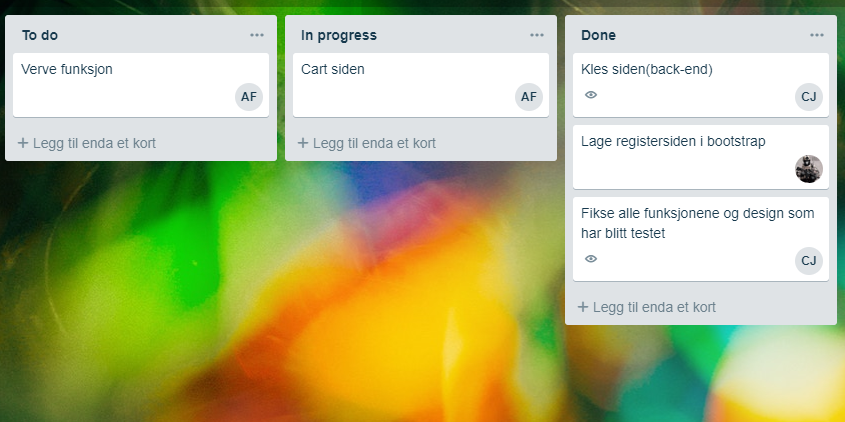
**Kristian:** Gjorde ferdig frontend på klessiden, han fikk også inn noe PHP for å spørre etter klær i databasen. Ble nesten ferdig med registrerings siden i bootstrap, siden den var foreløpig i CSS-grid.

**Andreas:** Fikk testet verve funksjon lokalt på pc, men for å få testet det på siden vår måtte litt mer på plass først, som sagt.

**Thomas:** Laget mange bannerbilder, og laget bilde-slider til forsiden.

#### **Sprint 6(22.03 🡪 29.03)**

Denne uken skulle vi prøve å få handlekurv og verve funksjonene til å funke, eller i det minste komme godt i gang. Det var 3 uker til fremføring, og disse MÅTTE være på plass innen da. Christoffer jobbet med filtrering av klessiden, og Andreas jobbet med handlekurv funksjonen. Når dette kom på plass, kunne Andreas implementere vervefunksjonen som han hadde fått til lokalt. Det var også litt ting som måtte på plass av den listen som vi hadde testet. Ellers var front-end teamet fokusert på å designe siden litt mer. Brukergrensesnittet trengte fortsatt en god organisering før applikasjonen skulle bli fremført.



Figur 8 Sprint brett 6

**Retrospektivt møte sprint 6(29.03)**

Denne sprinten gikk veldig fort. Det var ikke så mye som ble gjort. Det var mye surring på hva vi faktisk skulle få til før fremføringen av applikasjonen. Men ellers var det grei fremgang på backend. Filtrering var så og si ferdig, men det var enda diskusjon om hvordan denne skulle bli implementert på best mulig måte. Christoffer tok da kontakt med Bjørn for å høre litt om hva han syntes om dette. Listen med ting som var testet ble gjort. Som sagt var det ingen som hadde satt seg inn i noe jquery eller AJAX. For å få filteret til å fungere på best mulig måte, måtte Christoffer sette seg ned å lære litt AJAX og jquery. handlekurven ble også nesten ferdig, så til uken ville vi fortsette med å kombinere klessiden med handlekurven. Det var fortsatt masse testing som måtte bli gjort fremover før fremføringen. Også front-enden skulle vi prøve å få fokus på utover de 2 neste ukene.

**Christoffer:** Filtrerings funksjon på klessidene.

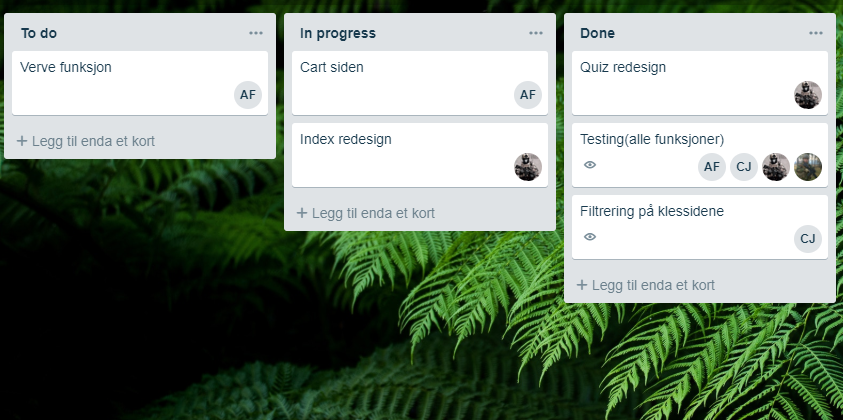
**Andreas:** Laget handlekurv siden og fikk deler av funksjonen til å funke.

**Kristian:** Laget registrerings-siden i bootstrap og fikset på masse små ting i frontenden.

**Thomas:** gjorde enderinger på design med nye bilder.

#### **Sprint 7(29.03 🡪 05.04)**

I denne sprinten skulle vi få ferdig filtrering og handlekurven. Vi måtte fokusere på å få testet alt, og få et veldig godt og gjennomført design. Designet var bra på frontenden, vi måtte bare endre litt på farger. Det var ikke så mye som måtte til på filtreringen, men Christoffer måtte finne ut av hvordan han skulle implementere det for å få det mest mulig dynamisk. Handlekurven begynte å komme seg, men det så ut som Andreas trengte litt hjelp, så vi gjorde vårt beste utover sprinten.



Figur 9 Sprint brett 7

**Retrospektivt møte sprint 7(05.04)**

Vi fikk til en del små ting som vi måtte få på plass. Ting som vi hadde testet f.eks. og design ting. Nye bilder og endring på farger. Filtreringen var på plass, dynamisk på alle de forskjellige klessidene. Vi trengte bare å teste litt for å se om alt funket som det skulle. Handlekurven var nesten ferdig, det var bare noen små funksjoner som manglet for å få den på plass. Nå jobbet vi med å få til å slette varer og endre antall varer på handlekurven. Siden det bare var en uke igjen til fremføring, så jobbet vi intensivt med dette. Ellers gikk det greit denne uken. Flere ting som kunne ha blitt gjort, kanskje.

**Christoffer:** Fikk filter funksjonen til å funke dynamisk på alle klessidene.

**Andreas:** Ble nesten ferdig med handlekurven, manglet sletting og antall varer.

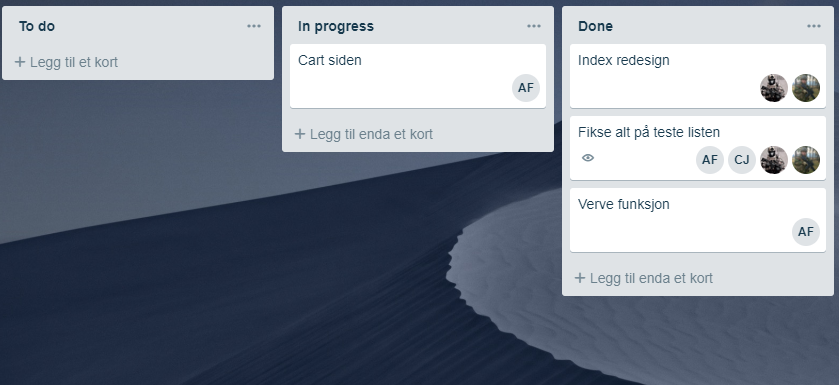
**Kristian og Thomas:** Endinger på design med nye bilder osv. Endret på forsiden, footer, profil, og registrering sidene.

**Alle sammen:** Testet ting.

#### **Sprint 8 Siste sprint (05.04 🡪 11.04)**

Denne sprinten måtte vi få på plass alt vi manglet. Det som ble testet 05.04 skulle på plass. Christoffer sendte melding til Bjørn angående lagring av æ,ø og å. Dette måtte på plass. Siden vi skulle fremføre applikasjonen måtte vi også gjøre klar powerpoint, og være klare for denne dagen. Så ut i fra det vi manglet på siden, var vi på litt tynn tråd. Men vi skulle ha møte sammen hver dag denne sprinten og få «ALT» på plass.

Så kort sagt måtte vi få på plass design, det som var testet.



Figur 10 Sprint brett 8

**Retrospektivt møte sprint 8(11.04)**

Vi fikk gjort så mye som mulig før presentasjonen. Den siste uken var lange dager for å prøve å få de siste oppgavene vi hadde igjen på plass. Til slutt var det handlekurven vi hadde problemer med. handlekurven ble ikke ferdig, og vi var ikke sikker på om vi i det hele tatt kom til å få den ferdig heller. Ellers ble resten av oppgavene gjort, og alt kom på plass unntatt handlekurven før presentasjonen.

**Christoffer:** Kombinerte PHP og Jquery for å få fine og gode inndatavalidering for bruker på registrerings-siden og profilsiden. Salg-siden ble laget, backend. Søkefunksjon i filtreringen ble laget. Flere errormeldinger ble laget. Æ,ø, og å på registeringen fungerte.

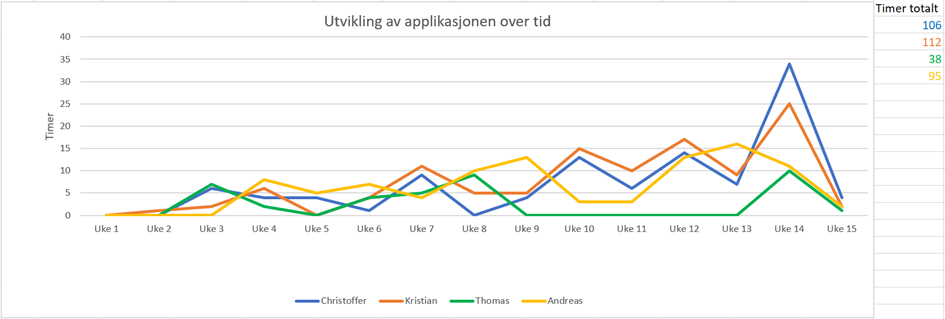
**Kristian:** Laget frontend på salgsiden. Gjorde ferdig forsidedesignet.

**Thomas:** Var med på å gjøre ferdig forsidedesignet.

**Andreas:** Satt fast med handlekurven, selve siden var på plass, men funksjonen ville ikke funke.

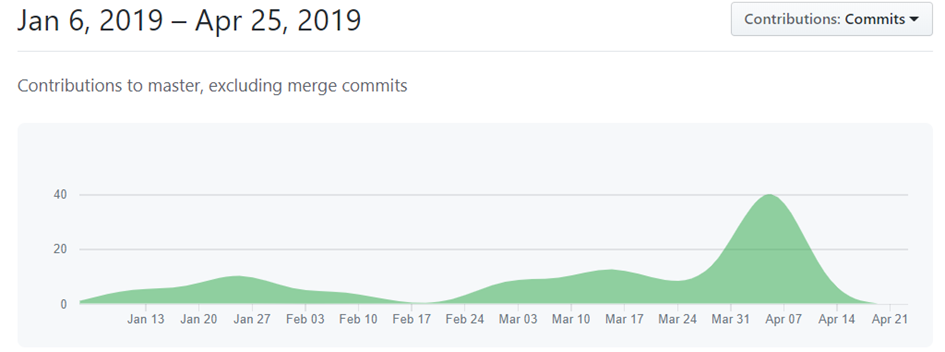
### Prosjektlogg

Vi har to grafer for å vise hvordan utviklingen av applikasjonen har vært gjennom perioden vi har jobbet med den. Den første grafen under viser tidsforbruket per person.



Figur 11 Viser periode vi har jobber i antall timer

Den andre grafen under viser når, og hvor mye kode som ble implementert på applikasjonen.



Figur 12 Viser når vi har gjort flest commits på applikasjonen

## Refleksjonsnotat

### Hva gikk bra?

Vi jobbet til slutt bra som en gruppe. Vi hadde god kommunikasjon i gruppa, og hjalp hverandre hvis det var nødvendig. Vi ble enig om å møtes på skolen onsdag, torsdag og fredag for å jobbe med applikasjonen, noe som fikk skikkelig fremgang på prosjektet. Til tross for litt komplikasjoner i starten, føler vi at vi kom ganske i mål med prosjektet. Vi har fått til de viktigste funksjonene, altså verving, quiz, og filter, som gjør applikasjonen vår litt unik.

### Hva gikk dårlig?

Vi hadde en vanskelig start på prosjektet. Vi slet litt med å bestemme oss for hvilken type applikasjon vi ville gå for. Vi valgte først en applikasjon for å sette opp ligaer på fifa, men ombestemte oss rett før milepæl 4 skulle leveres inn. Dette gjorde at vi ikke fikk så god tid som vi burde hatt på prosjektplanen.  
Vi hadde også en på gruppa som hoppet av gruppa rett etter vi hadde kommet litt i gang med applikasjonen. Vi skulle da egentlig bruke CSS grid system for å gjøre alt responsivt, men siden han hoppet av valgte vi heller å gå for bootstrap sitt grid system. Vi fikk da et lite tilbakeslag siden vi måtte lære oss bootstrap og gjøre all html som allerede var blitt utviklet om til bootstrap.   
Scopet for prosjektet var litt stort til å begynne med. Vi måtte gjøre scopet litt mindre ved å fjerne funksjoner som ikke var alt for relevante, slik at vi fikk tid til de viktigste funksjonene til applikasjonen.  
På grunn av forskjellige komplikasjoner kom vi derfor litt sent i gang med prosjektet. Vi burde egentlig kommet ordentlig i gang med applikasjonen litt kjappere enn det vi gjorde, da vi ikke innså hvor stort det endte opp med å bli.

### Hva burde blitt gjort annerledes?

Vi burde valgt riktig type applikasjon fra starten av, og i tillegg hatt en klar prosjektplan litt tidligere. Vi skulle ha begynt tidligere med utviklingen av applikasjonen, og sett på forskjellige rammeverk før vi startet på applikasjonen slik at vi slapp å bruke tid på å bytte. Vi burde ha innsett tidligere hvor stort prosjektet var, og tilpasset scopet deretter.

### Veien videre

#### Hva er gjort?

Etter vi valgte å ta litt ned på brukerhistoriene etter milepæl 4 innlevering, så er løsningen nesten helt ferdig ut ifra de nye brukerhistoriene. Et gjennomført design er utført med bootsrap rammeverk, javascript og CSS. Backend er utført med PHP der funksjonene til siden er laget. Databasen er også komplett med mange tabeller og assosiasjoner. Vi har også utført sikring av siden, ved å sikre mot SQL injections.

#### Hva gjenstår?

Fra den første milepælen så gjenstår det litt:

* «Som en kunde vil jeg ha en ønskeliste, så jeg kan finne fram til varer jeg ønsker meg til en vær tid.».
* «Som kunde vil jeg ha brukerhistorikk slik at alt jeg har lagt i handlekurv ikke går tapt»
* «Som kunde vil jeg ha flere måter å utføre betaling på»

Etter at vi fikk en veldig dårlig start med et stort scope, så måtte vi trakke ned på disse. Med de nye brukerhistorie-diagrammene, så har vi prøvd å lage alle brukerhistoriene. Kun en av disse ble ikke helt utført og dette er da handlekurven.

#### Er det utløste problemer?

Vårt uløste problem er handlekurven. Denne handlekurven ble dessverre problematisk mot slutten av innleveringen. Hadde vi vist at denne skulle bli så vanskelig som det ble, hadde vi satt mer tid på den.   
Grunnen til at vi har problemer med den er fordi vi prøver å lage en tabell som SESSION variabel. Dette ble problematisk fordi vi ikke har greid å få tabellen assosiativ. Så når kunden legger en ny ting i tabellen, så overskriver den bare det som allerede ligger i tabellen. Det som skjer er at vi har ikke noe nøkkel til å holde styr på de forskjellige verdiene i tabellen.

Så for testingens skyld så har vi laget en slags hardkodet handlekurv bare for å teste at verve funksjonen funker.

Her vil du da kunne teste og legge inn en ordre. Men det er alltid samme klesplagg.

Hvis vi skulle ha gjort handlekurven på nytt, så ville vi lagt inn et skjema som hadde tatt imot adresse til kunden. Vi ville da tatt våres «ideale» databaseløsning og lagt inn adresse, når kunden la inn ordren og når ordren hadde blitt sendt til kunde i «Orders» tabellen. Vi ville da også lagt inn hvilke plagg kunden bestilte inn i «OrderLine» tabellen.

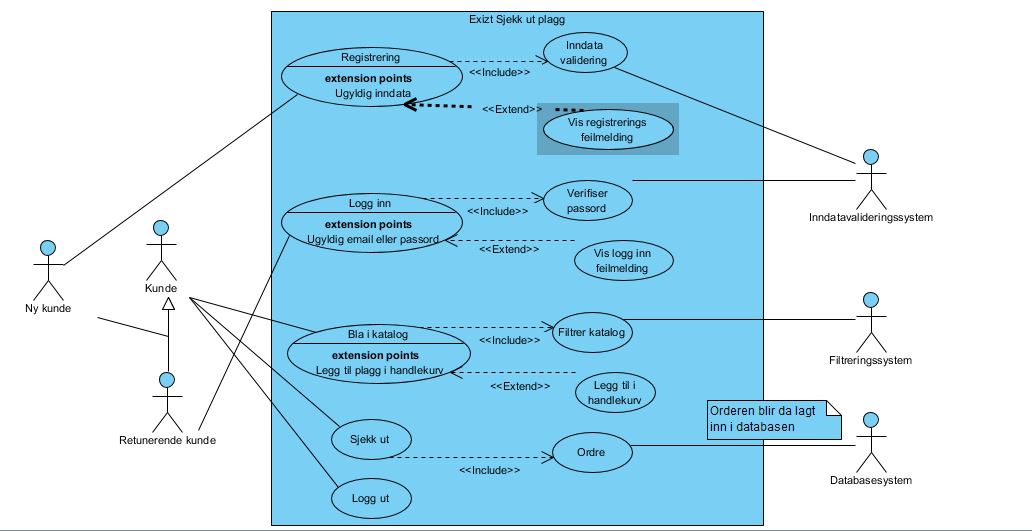
# Faglig del

## Kravspesifikasjon

### Use cases

**Use case diagram «Sjekk ut plagg»:**

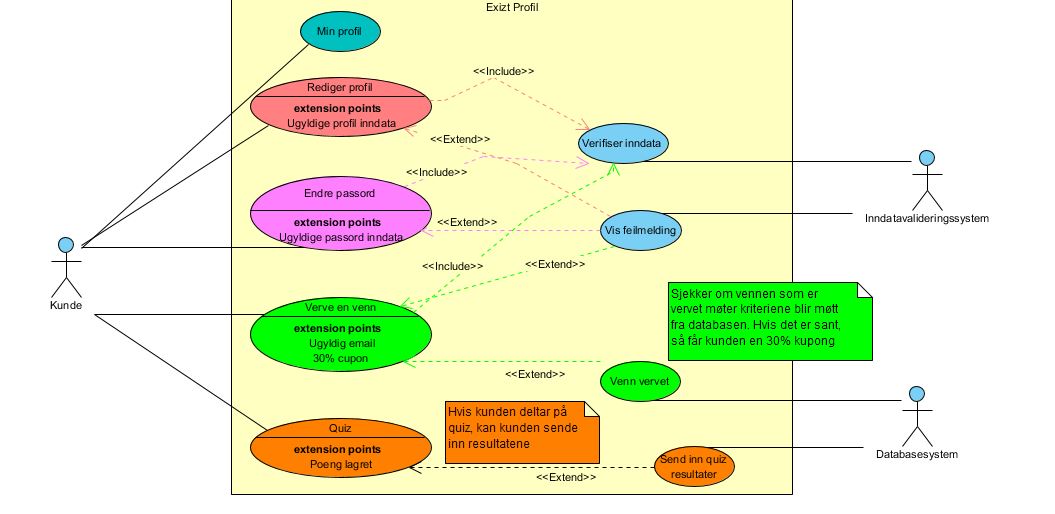
* Først så må det være en kunde som kan kjøre i gang use casene. Det kan da enten være en «Ny kunde» eller en «Returnerende kunde». Hvis det er en ny kunde så, må kunden registrere seg først.
* Registreringen har inndata valideringer som skjer vær gang en kunde registrerer seg. Ellers så sender den en feilmelding bare hvis kunden har skrevet noe feil.
* Hvis det er en returnerende kunde så går kunden rett til «Logg Inn» use-casen. Login verifiserer om passord er riktig vær gang. Ellers så sender en en feilmelding om kunden har skrevet inn feil passord.
* Når da kunden er logget inn kan kunden gå til katalogen. Når kunden er i katalogen så vil katalogen bli filtrert ut ifra hva kunden trykket inn på. Ellers så står kunden fritt til å filtrere på det den ønsker. Kunden kan da velge å legge inn klesplagg inn i handlekurven sin.
* Hvis kunden sjekker ut vil det bli lagt inn en ordre i tabellen. Siden vi valgte å ta bort noen use-cases fra vår første brukerhistorie rapport, så ser use-casesen slik ut nå.



Figur 13 Use-case diagram som viser hvordan man kan sjekke ut ett eller flere plagg fra klessiden

**Use case diagram «Profil»:**

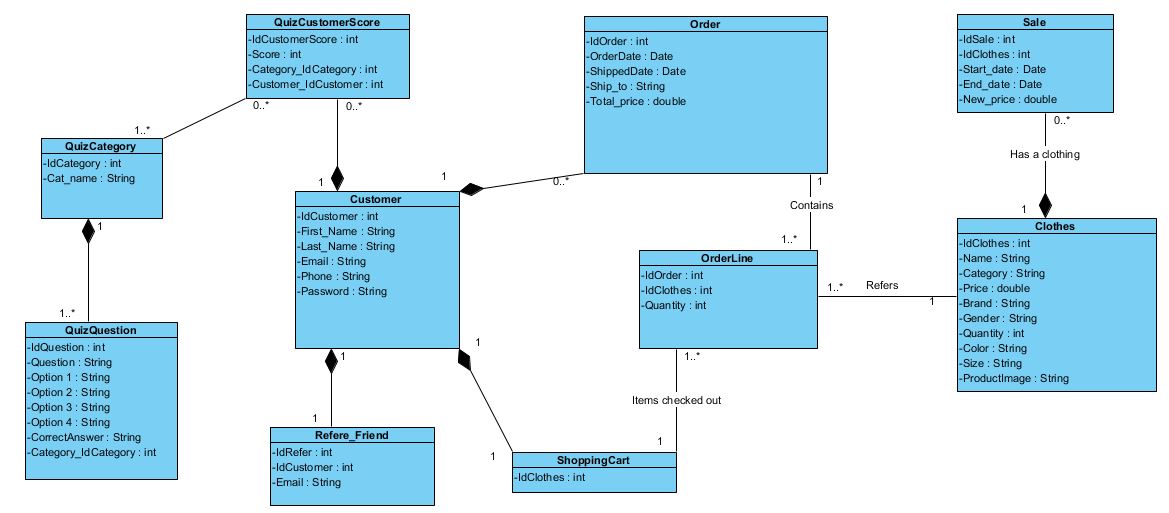
* Kunde kan velge mellom ett par ulike funksjoner når kunden er inne på profilen sin.
* Hvis kunden velger «Rediger profil», så kan kunden velge mellom å endre på navn, etternavn, tlf og email. Det vil bli gjort inndata verifiseringer og feilmeldinger vil bli vist om kunden ikke har gjort riktig.
* Samme går for «Endre passord». Inndata vil bli sjekket og feilmelding vil bli vist dersom det er gjort feil.
* Velger bruker «Verve en venn» så vil inndatavalideringssystemet sjekke om det er en ny bruker og om det er en email kunden skriver inn. Hvis alle kravene blir godkjent for å verve en venn så vil kunden få en kupong gode på 30%.
* Velger kunden «Quiz» casen så kan kunden være med på en trekning av premier. Kunden kan da velge å sende inn resultatene sine, og disse resultatene vil bli lagret i databasen.



Figur 14 Use-case diagram som viser hvordan de forskjellige funksjonene inne på profilen fungerer

## Systembeskrivelse

### Overordnet beskrivelse



Figur 15 Et klassediagram som viser en domenemodell for applikasjonen

Vi har designet siden sånn at kunden lett kan finne frem til det den ønsker. Ikke for mange ting på siden som kan få kunden til å gå i surr. Vi har prøvd å holde det så enkelt som mulig. Av og til så er det enkle ofte det beste. Kunden får tilbakemeldinger på ting man har gjort/gjort feil. Footer med god informasjon om siden. Profil side der man kan se litt informasjon om seg selv, og kan oppdatere sin informasjon til databasen.

Koden har vi også prøvd å designe sånn at det er lett å finn fram til ting for programmereren. Det kan bli veldig styr etter hvert som kodebasen blir veldig stor.

### Brukergrensesnitt

På brukergrensesnittet har vi hatt Kristian og Thomas til å designe. Vi har brukt en farge kode på siden som vi bestemte oss for. Lysgrå, svart og Orange er sidens fargekode. Dette gjør at siden blir bedre å se på og vi kan også designe lettere når vi har en fargekode å gå etter.

Brukergrensesnittet består av HTML, CSS og Bootstrap som rammeverk. Bildene som vi har brukt på siden har Thomas redigert med Photoshop. For bildene så er det brukt en klasse i Bootstrap som kalles «img-fluid». Det gjør at bildene blir responsive i forhold til mobil og nettbrett. For headeren og footeren så har vi også brukt Bootstrap klasser for å få dem resposive. Disse klassene er også veldig greie og ta i bruk istedenfor å måtte gjøre alt resposivt selv.

### Klientsiden

For å lage klientsiden har vi brukt bootstrap, css, html, ajax, jquery og javascript.  
Vi bygde det meste opp med div tagger i html, som inneholder bootstrap klasser for å style elementene og gjøre dem responsive. Vi brukte også css for å overwrite bootstrap sin css slik at vi kunne style ting som vi selv ville. Vi brukte bootstrap sitt grid system for å gjøre ting enklere responsivt. Måten dette grid systemet fungerer på er at det er maksimalt 12 kolonner, og man kan velge selv hvor stor plass forskjellige elementer skal ta ved å skrive hvor mange kolonner de skal bruke på forskjellig skjermstørrelse. På pc-skjermer brukte vi f.eks. størrelse 3 på hver kolonne, slik at 4 kolonner blir vist. Hvis man tar skjermstørrelsen ned til nettbrett, brukte vi f.eks. størrelse 6 på hver kolonne slik at 2 kolonner blir vist, og størrelse 12 på mobil slik at bare 1 kolonne blir vist.  
Vi lagde navbaren og footeren i separate filer som vi hadde i en require på hver side, slik at vi slapp å skrive samme kode på hver side.

Jquery sine bruksområder på websiden er hovedsakelig for feilhåndtering av bruker og for filtreringsfunksjonen på katalogsidene. Jquery har vi bare satt med script tagger på bunnen av sidene vi tar dette i bruk. Kanskje det hadde vært litt mer ryddig om vi hadde hatt disse scriptene i separate filer. Sidene som tar dette i bruk er registrerings side, katalog sidene og profil siden. Grunnen til dette er fordi feilhåndteringen oppstår i registreringen og ved oppdatering på profil siden. Måten vi håndterer dette på er med en kombinasjon av PHP og Jquery. Vi setter PHP variabler i nettadressen ut ifra hva bruker har gjort feil. Henter disse ut i en Jquery variabel. Ut ifra hva denne Jquery variabelen inneholder, så skal scriptet gjøre noe. For eksempel si at passordene ikke stemmer med hverandre.

Vi har også jquery på katalogsidene, men dette er kombinert med AJAX. Først så hadde vi tenkt å lage katalogen med kun backend. Men så at dette ble dårlig design. Siden da må siden oppdatere seg hver gang en kunde gjør en endring i katalog filteret. Så da valgte vi en i gruppen som skulle sette seg ned å lære mer om jquery og AJAX, for å få katalog sidene til å oppdatere deler av siden uten å måtte oppdatere hele siden på nytt.

Måten vi har gjort dette på er å sende med variabler med alle de forskjellige verdiene som bruker kan filtrere på. Disse variablene blir da sendt til fetch\_data.inc scriptet. Ut ifra hva bruker har filtrert på, så vil det da dannes en SQL spørring som danner en output som blir sendt tilbake som html kode til katalog siden.

Vi brukte også javascript for å lage f.eks. slider med bilder, i tillegg til at bootstrap har en del innebygd javascript.

### Tjenersiden

På tjenersiden så er det da brukt PHP. Vi har prøvd å holde PHP skripts inne i includes mappen. Hva dette vil si er at ting som kun inneholder PHP vil bli lagt inn her. Disse PHP skripta som er inne i includes mappen har en jobb å gjøre og kun denne jobben. Så for eksempel om en bruker registrerer seg og trykker på submit knappen, da vil denne formen bli sent til skriptet kalt «registration.inc.php». Dette skriptet har da en jobb, det er da å ta av seg registreringen. Grunnen til at det er .inc bak dette skriptet er bare for å se at dette da er et «includes» script. Denne måten å håndtere PHP skripts ble vi enige om på forhånd før vi begynte å kode. Vi syntes denne måten gjorde det lettere å se hvilke skripts som gjør hva.

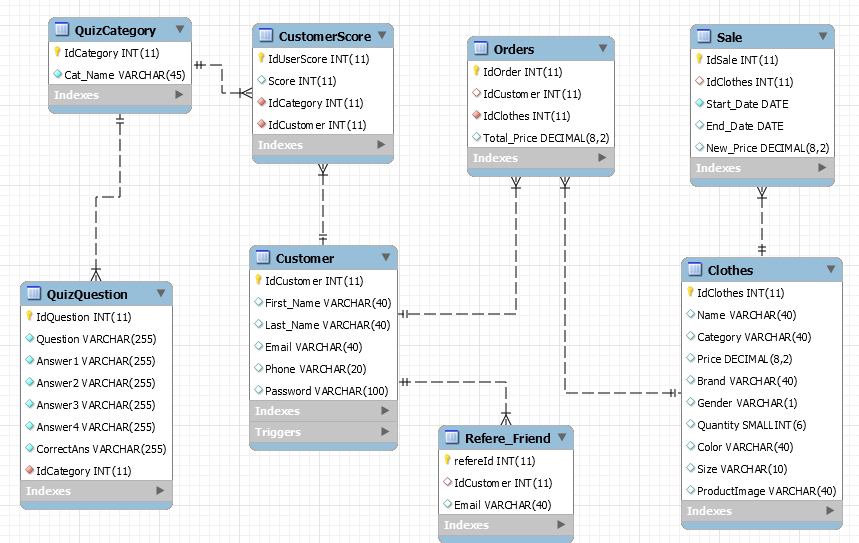
Vi har et script i includes mappen som heter «dbh.inc.php» som står for «database handler». Dette scriptet tar seg av tilkoblingen til databasen. Vi bruker «require» i toppen av scripts der vi trenger databasetilkobling. Vi har da en variable som vi kaller for «$conn» for å tilkoble oss til databasen.

Vi har også prøvd å gjøre en så god jobb som mulig med å hindre SQL-injections og andre sikkerhets brudd. Ting vi har da gjort for å hindre dette er da å ha gode inndatavalideringer. Vi har sjekker da med regular expressions på inndatafelt. For eksempel på registeringen og redigering av profil er dette brukt på navn, email, tlf og passord.

Vi har også brukt prepared statments til databasen der vi kan, dette bruker vi for å hindre SQL-injections. Dette gjør at databasen analyserer, kompilerer og utfører spørreoptimalisering på SQL-setningen, og lagrer da resultatet uten å utføre den. Da på et senere tidspunkt binder applikasjonen verdiene til parametere, og databasen utfører setningen.

For å få til feilmeldinger så har vi satt PHP variabler i nettadressen. For å vite hva vi skal sende som feilmelding, så blir det satt variabler ut ifra hva brukeren har gjort feil. Verdiene til disse variablene blir da hentet ut i en jquery variabel, og vi kan da sette farge og en melding til brukeren. Vi har da kombinert jquery for frontend og PHP for backend i inndatavalideringen.

### Databasen

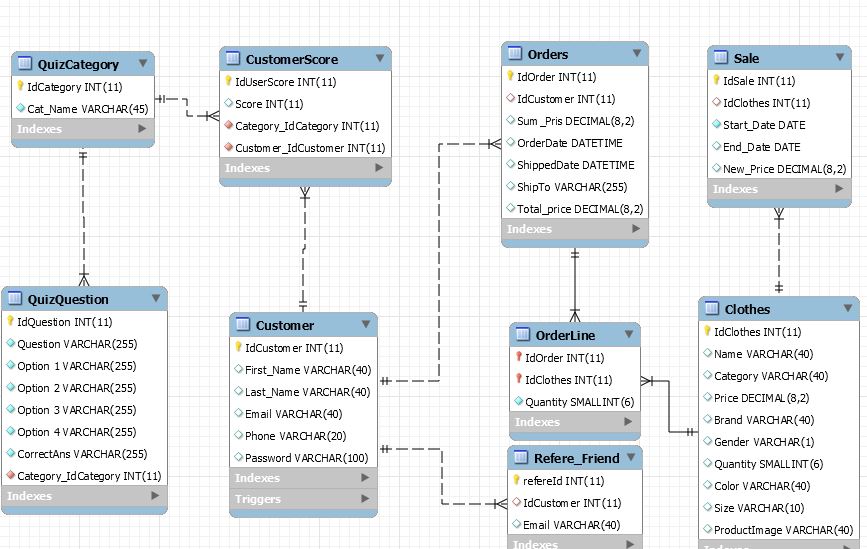


Figur 16 Vår endelige løsning på databasen

Databasen vår er satt opp av 8 tabeller med forskjellige assosiasjoner til hverandre. Vi bruker da MySQL som databasesystem. Vi bruker h18gr2 bruker på itfag sin server. Vi tenkte det var greit at alle på gruppen har tilgang til samme tilstand av databasen til en hver tid. Istedenfor ville vi ha hatt det lokalt, og alles inndata i databasen ville ha vært annerledes. Også hver gang den databaseansvarlige skulle ha gjort noen endringer, så måtte alle ha gjort disse lokalt. Derfor så valgte vi å gjøre det slikt.

I databasen så lagrer vi informasjon om kunder, klær, ordre og quiz funksjoner. I kunde tabellen så er passordene hashet, så hvis noen som ikke skulle hatt tilgang hadde fått tilgang til databasen, så kan man ikke gjøre så mye med disse hashete passordene.

Bildene vi har på klærne er lagret lokalt på maskin med link i klær tabellen til bildene. Det vil realistisk sett vært bedre å lagre bildene i databasen.

Databasen våres har vært i gjennom en del tilstander. Vi har måttet underveis hele tiden endre litt på tabeller, legge til nye tabeller og slette tabeller. ER-diagrammet som forklarer databasen ville vi ikke ha gått videre med. Vi fant ut mot slutten at logisk sett så ville ikke denne databasen fungert med ordre. Men siden vi hadde så liten tid til å re-designe koden vår opp mot den nye database løsningen, så ble slutt databasen den du ser over. Her ser du hvordan vi ville hatt databasen: 

Figur 17 Er-diagram som en bedre løsning på databasen

Vi har tatt med scriptet til databasen løsningen på figur 17 hvis du vil se på den.

Vi har også noen funksjoner og triggere inne i databasen. Vi har en funksjon som tar seg av å få store bokstaver. Vi har da brukt to triggere på oppdatering på navn og etternavn. Hver gang en bruker skriver liten bokstav som forbokstav på navn eller etternavn, så vil denne triggeren rette dette opp.

## Installasjon

For å teste ut løsningen så ta zip filen og pakk den ut i htdocs mappen inne i xampp. Du vil da få en mappe som heter APPGRUPPE2. Inne i denne mappen ligger den en mappe som heter SQL. Her ligger det 4 scripts. «LageDatabasen» er den du skal bruke for å opprette databasen. Hvis det skulle skje noe feil med databasetilkobling, sørg for å sjekke at koden inne i «dbh.inc.php» tilsvarer navnet og plasseringen til databasen din.

«h18grdb2V» Er vår andre løsning på databasen, hvis du vil se på denne. «capitalize\_function» og «tr\_UpperName» er en trigger løsning på navn sånn at det alltid blir stor forbokstav på fornavn og etternavn.

Når du har fått databasen på plass så er det bare å starte XAMMP og teste i vei. For å komme til index siden så skriv dette i URLen: «<http://localhost/APPGRUPPE2/index.php>».

Vi har også laget en test bruker som du først kan logge inn med. Med ellers så er det bare å registrere nye bruker for å logge inn med.

Testbruker: [testbruker@gmail.com](mailto:testbruker@gmail.com)

Passord: Test1234

## Testrapport

### Testplan

I sprint 5 så begynte vi ordentlig med testing. Da hadde vi fått i gang mange ting på siden og tenkte det måtte testes en del. Ellers så hadde vi testet at ting fungerte ordentlig før det ble pushet til github. Testingen gjorde vi i plenum. Da testet vi funksjonalitet, brukervennlighet og kompatibilitet for nettbrett og mobil.

### Funksjonalitet testing

For å teste funksjonaliteten på applikasjonen, så testet vi for eksempel websidelinker, databasetilkobling, former og inndata. Vi sjekker om alle linker funker som de skal. Hopper linker til riktig side osv.

Testing av former som får informasjon fra brukere. Da sjekket vi:

* Inndatavalideringer i feltene
* Om det blir lagret riktig i databasen
* Prøver med forskjellige inputs i feltene

### Brukervennlighet

Her har vi testet:

* Om det er enkelt å lære siden
* Navigasjonen
* Brukertilfredshet
* Generelt utsende

Med navigasjon har vi testet om brukeren lett kan navigere rundt på siden med knapper og linker.

Vi har testet om innholdet er lett å forstå. Sjekket etter stavefeil. Bruken av forskjellige farger som kan irritere bruker.

### Kompatibilitet

Når vi testet kompatibilitet, så sjekket vi om alt som var på pc versjonen også var på nettbrett og mobil. Vi testet om ting ikke ble for smått og ikke for zoomet inn. Vi testet at mobil brukere også hadde like lett tilgang til ting sånn som pc brukere. Om bilder ble riktige størrelse. Vi testet om grid systemet tilpasset seg riktig ut ifra størrelse på skjerm osv.

### Testrapporter

Vi hadde to rapporter som vi laget og fikset på. Noen ting som vi testet ble ikke rapportert så det er sagt. Siden det kunne være veldig mange små ting som vi bare fikset på der og da.

Rapport 1:

* Fikse æ,ø,å validering for registrasjon
* Fikse sånn at bruker ikke kan registrere en ny bruker mens man er logget inn.
* Validering på tlf.
* Forskjellige vanskelighetsgrader på passord valideringen
* Sikre mot sqlinjection på profil
* Få inn ordentlige feilmeldinger på siden (registrering, profil).
* Fikse samkjør mellom css’en mellom top.php og profil.php
* Fikse på brukergrensesnittet.
* Logut funksjon

Rapport 2:

* Nytt indexdesign
* Returmelding på quiz hvis bruker allerede har deltatt
* Skrive inn sitt gamlepassord når man endrer passord.
* Ordentlige meldinger på hva man gjør feil
* Quiz knapp på profil siden
* Vervefunksjon inne på profil siden

## Kilder og referanser

### Bøker

* Henrik Kniberg & Mattias Skarin. (2010). Kanban and Scrum making the most of both.
* Svend Andreas Horgen: Webprogrammering i PHP, 3. utgave 2009
* Laura Lemay, Rafe Colburn, Jennifer Kyrnin: Sams Teach Yourself HTML, CSS & JavaScript Web Publishing 7. Edition 2016

### Nettressurser

* <https://stackoverflow.com/>
* <https://www.php.net/manual/en/index.php>
* <https://getbootstrap.com/>
* <https://api.jquery.com/>
* <https://www.w3schools.com/>
* <https://www.youtube.com/user/TheCharmefis> <https://usn.instructure.com/courses/14599/modules>

### Programbiblioteker

### Ekstra informasjon om kilder

For PHP så har vi lært mye av leksjonene på høst semesteret. Før vi startet i gang med prosjektet så satt backend med litt selv studie på PHP. For å legge en grunnmur for PHP språket så ble PHP boka og en youtuber som heter mmtuts(linket over) brukt.

Etter hvert som vi da begynte med PHP funksjoner så ble stackoverflow og php manualen veldig aktuelle. Brukte da som regel stackoverflow for å søke på et problem som hadde oppstått. Og hvis det for eksempel var noen funksjoner i PHP som ble brukt i disse svarene på stackoverflow, så brukte vi da manualen for å forstå disse funksjonen.

Når det kom til HTML og CSS så kunne alle grunnleggende, så vi trengte ikke akkurat en oppfriskning på dette. Hvis det var noe vi trengte oppfriskning på, så ble det mest brukt w3schools.

For å lære bootstrap, så ble for det meste bootsrap sin hjemme side brukt og w3schools som ble brukt. Og det samme ble det for å søke opp problemer på stackoverflow, og analysere klasser som har blitt brukt i disse svarene på stackoverflow.

Med jquery og AJAX så ble jquery API, w3schools og mmtuts(youtube kanal) brukt. Siden Vi hadde liten tid på å lære jquery og AJAX på, så ble youtube tutorials veldig aktuelt for dette.