

Forprosjektrapport

ELEKTRONISK BESØKSREGISTER FOR NC-SPECTRUM

ERIK LI	S236777
ANDREAS STENSRUD	S236742
JOAKIM F. MØLLER	S196960
EMIL R. NEDREGÅRD	S236767

TABLE OF CONTENTS

1	Presentasjon	3
1.1	Presentasjon.....	3
1.2	Gruppe.....	3
1.3	Oppdragsgiver	4
1.4	Oppgaven	4
1.5	Kontaktpersoner.....	4
2	Sammendrag	5
3	Dagens situasjon	5
4	Mål og Rammebetingelser	6
4.1	Mål	6
4.2	Rammebetingelser	6
5	Løsninger	7
5.1	Utviklingsspråk og rammeverk	7
5.2	Utviklingsverktøy.....	8
6	Analyse av virkninger	9
6.1	Front-end.....	9
6.2	Back-end.....	9
6.3	Oppsummering av analyse	10
7	Arbeids- og Fremdriftsplan	11
7.1	Tabellrepresentasjon.....	11
7.2	Gantt-diagram-representasjon.....	12
8	Vedlegg	13
8.1	Gantt diagram	13

1 PRESENTASJON

1.1 PRESENTASJON

NC-SPECTRUM

Oppdragsgiver	NC-Spectrum	
Prosjekttittel	Elektronisk besøksregister for NC-Spectrum	
Oppgave	Prosjektet går ut på å utvikle et elektronisk besøksregister for NC-Spectrums operasjonssentral som kan erstatte deres nåværende utdaterte fysiske journalbok.	
Bedriftsnettside	https://www.nc-spectrum.no/	
Gruppenummer	19	
Gruppedlemmer	Erik Li	s236777
	Andreas Stensrud	s236742
	Joakim F. Møller	s196960
	Emil R. Nedregård	s236767
Veileder	Terje Gjøsæter	
Gruppenettside	http://student.cs.hioa.no/~s236767/bachelornettside/	
Prosjektperiode	05.01.2017 – 24.05.2017	

1.2 GRUPPE

Gruppe 19 består av Erik Li, Andreas Stensrud, Joakim F. Møller og Emil R. Nedregård. Vi er en blanding av studenter som går på bachelorlinjene, "Anvendt Datateknologi" og "Informasjonsteknologi" ved Høgskolen i Oslo og Akershus, fra Høsten 2014 til Våren 2017.

Emil, Andreas og Joakim har arbeidet sammen gjennom studiet, men det er første gang Erik jobber sammen med dem. Vi har ulike kompetanser med tanke på hva vi kan og ikke kan. Erik er mer erfaren i programmering/back-end, Emil er ganske erfaren på front-end, GIT og serverhåndtering mens Joakim og Andreas er erfarne innenfor testing og dokumentasjon. Vi kommer nok til å fordele oppgavene slik som oppgitt, samtidig som alle kommer til å gjøre noe innenfor alle oppgavene og feltene.

1.3 OPPDRAGSGIVER

NC-Spectrum er et konsultentselskap som ligger i Kviteseid. Selskapet fokuserer på prosjektering, rådgivning, etablering, drift av elektronikk og infrastruktur i bredbåndsnett. NC-Spectrum har over 230 kunder, inkludert mange små og mellomstore e-verk, vannverk og renseanlegg, hvor cybersikkerhet på operasjonssystemene og –nettverkene må ivaretas. NC-Spectrums løsninger er innovative og fleksible, og skreddersys til kundens utfordringer og krav til sikkerhet, oppetid og stabilitet. De er også partner i ”Center for Cyber and Information Security” (CCIS) og arbeider tett med blant annet KraftCERT.

1.4 OPPGAVEN

Opgaven vår går ut på å utvikle et elektronisk besøksregister for NC-Spectrums operasjonssentral. De bruker i dag en fysisk bok som besøkende manuelt må skrive seg inn og ut i. De ønsker dermed at vi utvikler dem en god og brukervennlig besøksregistreringsløsning, som også i tillegg ivaretar NC-Spectrums krav til sikkerhet og besøkendes personvern. Systemet må ha søkemulighet på personer og dato, og gjerne synkronisere avtaler med webløsningen vår. Løsningen skal implementeres i form av en prototype og skal brukertestes. Av de mange ulike språkene vi kommer til å bruke for å utvikle systemet, vet vi at språket PHP er et ønske fra oppdragsgiveren. Derfor vil vi bruke PHP som et av hovedspråkene for dette systemet.

Besøkende skal forholde seg et enkelt og oversiktlig brukergrensesnitt der man kan registrere seg med navn, firma og m.m. Besøkende vil deretter bli lagret i databasen og kan enkelt melde seg inn og ut av hvert besøk uten å registrere seg på nytt. Administrator vil ha et brukergrensesnitt som får oversikt over blant annet historikk, databasen, registrerte gjester og rettigheter til å endre på informasjon m.m.

Det ble ikke gitt mange kravspesifikasjoner enn det som allerede har blitt nevnt, dermed står vi relativt fritt til hvordan vi har lyst til å utvikle og utvide mer på systemet.

1.5 KONTAKTPERSONER

Veileder ved HIOA

Terje Gjøsæter

Førsteamanuensis

E-post: Terje.Gjosater@hioa.no

Telefon: +47 917 53 413

Kontaktperson ved NC-Spectrum

Silje Aakre

Prosjektmedarbeider

E-post: Silje.aakre@nc-spectrum.no

Telefon: +47 400 95 209

2 SAMMENDRAG

Vi skal våren 2017 skrive bacheloroppgave i samarbeid med NC-Spectrum. Vi har fått i oppgave å lage et elektronisk besøksregister for operasjonssentralen. Webløsningen skal holde styr på de besøkende på grunn av sikkerhetsmessige årsaker. Besøkende må melde seg inn og ut hver gang de går inn eller forlater operasjonssentralen. De ønsker seg en digital webløsning da dagens løsning er i en fysisk bok. Administratorene vil i tillegg også få sitt eget grensesnitt som holder styr på ulike ting som historikk, oversikt over registrerte gjester og ansatte, registreringsinformasjonen til besøkende, og rettighetene til å endre informasjon og lage nye administratorer.

Tidlig i prosessen var vi på besøk hos NC-Spectrum for å se på dagens løsning, vi gikk også gjennom funksjonalitet som måtte og kunne være med. Samtidig brukte vi tid på å bli enige om hvordan den endelige webløsningen kunne komme til å se ut. Vi står relativt fritt til å lage webløsningen som vi selv vil, så lenge de samsvarer med de få kravene som allerede ble satt av oppdragsgiveren.

Språkene som skal brukes på denne webløsningen er blant annet, HTML, CSS med Bootstrap, JavaScript, JQuery, AJAX, JSON, PHP og ulike utviklingsverktøy som XAMPP (Apacheserver og MySQL database), GitHub, MVC, Netbeans og Sublime Text. Ved bruk av disse språkene og verktøyene skal vi utvikle et brukervennlig, enkelt, raskt, oversiktlig og universelt utformet produkt. Webløsningen skal primært brukes gjennom en nettportal.

Informasjon om veileder og kontaktperson [finner man her](#).

3 DAGENS SITUASJON

NC-Spectrum ønsker som nevnt en elektronisk besøksregistreringsløsning. Per dags dato skriver besøkende seg inn og ut i en fysisk bok med navn, firma og dato m.m. Slik teknologien er i dag er det en utdatert metode for besøksregistrering og ikke minst synes de selv at det er uprofesjonelt. Derfor er det nødvendig og elementært at dette blir elektronisk.

4 MÅL OG RAMMEBETINGELSER

4.1 MÅL

Målet er å lage et system som registrerer besøkende for operasjonssentralen til NC-Spectrum. Webløsningen gir gjestene et enkelt, oversiktlig, brukervennlig og universelt utformet brukergrensesnitt som de kan forholde seg til. Førstegangsbesøkende registrerer seg inn i systemet og deretter ved et tastetrykk kan man enkelt sjekke seg inn og ut, og alt annet vil bli håndtert av systemet. En administrator vil ha sitt eget brukergrensesnitt som får oversikt over ulike informasjon som historikk, besøkendes registreringsinformasjon, oversikt over alle ansatte og besøkende, statistikk og andre ulike opsjoner. En slik webløsning kan gjøre registrering mer effektivt enn med en fysisk bok.

4.2 RAMMEBETINGELSER

Oppdragsgiver ønsket at vi skulle programmere webløsningen i PHP fordi det var et språk de også følte seg komfortabel med, slik at de kunne legge inn endringer i koden etter prosjektet er avsluttet. Annet enn det sto vi ganske fritt til hvordan vi vil utvikle webløsningen, dermed kom vi frem til at vi ville gjerne bruke følgende språk og verktøy for webløsningen:

Programmeringsspråk og rammeverk

- PHP
- HTML & CSS/Bootstrap
- JavaScript
 - JQuery
- AJAX
 - JSON
- MVC
- MySQL

Verktøy

- Netbeans & Sublime Text
- XAMPP
 - Apache Server
 - MySQL Database
- GitHub

Vi ønsker å lage en webløsning som skal være tilgjengelig gjennom en nettportal. Av teknisk utstyr kommer vi primært å ta utgangspunkt i at webløsningen skal brukes på en iPad, men som nevnt tidligere skal den være universelt utformet, dermed må den også fungere på alle typer smarttelefoner, nettbrett og datamaskiner.

5 LØSNINGER

Dette avsnittet bygger på avsnitt [4 Mål og Rammebetingelser](#). Her går vi igjennom hva de ulike utviklingsspråkene og –verktøyene vi bruker er og gjør.

5.1 UTVIKLINGSSPRÅK OG RAMMEVERK

PHP

PHP er et server-side-tolket scripting-språk som brukes primært for webutvikling, men på grunn av at PHP er veldig dynamisk og løst typet, egnes det også godt for generell programmering. PHP kode er som regel skrevet sammen med HTML kode, men for at serveren skal oppfatte at .html dokumentet inneholder PHP-kode må fil-extension bli forandret til PHP.ⁱ

HTML

HTML (HyperText Markup Language) er et standard markup språk som for å utvikle nettsider. Vi kan se på dette som skjelettet for alle nettsider, hva titler, avsnitter, seksjoner, bilder og metadata skal inneholde.

CSS/Bootstrap

CSS (Cascading Style Sheet) er språket som blir brukt for å designe utseende og innholdet på ulik markup-språk. Primært brukes dette alltid sammen med HTML for å pynte på de ulike HTML-seksjonene. Dermed kan dette sees på som en utvidelse av HTML.

I denne forbindelse har vi valgt å bruke Bootstrap som er et gratis front-end CSS-rammeverk for design av nettsider. Bootstrap inneholder maler på HTML- og CSS design, som blant annet skrifttyper, farger, knapper og inputbokser for å gjøre front-end designen mye enklere for alle brukere. Dette er et av de mest brukte CSS-rammeverkene i verden.

JavaScript & JQuery

JavaScript er et høynivå og dynamisk programmeringsspråk som sammen med HTML og CSS brukes av alle nettsider. Det brukes som regel for å lage mer dynamikk på nettsider og gjøre nettsiden mer brukervennlig og forståelig.

JQuery er et JavaScript-bibliotek som forenkler JavaScript på mange måter. Den forenkler syntaks og gjør JavaScript mer oversiktlig og mindre kronglete. Det er det mest brukte JavaScript-biblioteket i verden.

AJAX/JSON

AJAX (Asynkront JavaScript og XML) er et sett av webutviklingsteknikker på klient-siden som brukes til å lage asynkrone webapplikasjoner. Det vil si at AJAX sørger for at webapplikasjoner kan sende og motta data fra enten server eller klientsiden i bakgrunnen uten å skape forstyrrelser for siden man er på. ⁱⁱ

JSON (JavaScript Object Notation) er en *open-standard* format som sender dataobjekter i tekstformat i stedet for XML-format som benyttes i AJAX. Det vil da si at JSON brukes som regel sammen med AJAX for å erstatte dens XML-format. ⁱⁱⁱ

MVC

MVC (Model-View-Controller) er et kjent og enkelt designmønster som deler programvaren inn i tre deler. Dette er for å skille hvordan den interne informasjonen er representert før de blir representert til brukere. Model behandler data, logikken og reglene i programvaren. View representerer informasjonen som brukere ser, nemlig brukergrensesnittet. Controlleren behandler informasjon, som dataene og kommunikasjonen mellom View(brukere) og Model. ^{iv}

5.2 UTVIKLINGSVERKTØY

Netbeans/Sublime Text

Netbeans IDE (Integrated Development Environment) er et utviklingsverktøy som brukes hovedsakelig for utvikling i språk som HTML5, PHP og C/C++. Den har en omfattende debugger som er nyttig under utvikling for å finne feil og analysere koden. Samt har den god støtte til XAMPP.

Sublime Text er en teksteditor som støtter mange ulike programmeringsspråk og plugins. Er mye brukt av profesjonelle brukere på grunn av dens frihet med språk og plug-ins.

XAMPP

XAMPP (Cross-Platform (X) – Apache (A) – MariaDB (M) – PHP (P) – Pearl (P)) er en cross-plattform versjon av LAMP (Linux) og WAMP (Windows). Dette utviklingsverktøyet kjører ulike typer services som brukes hovedsakelig av utviklere for å kjøre brukbare webservere lokalt for utvikling og testing av løsninger. ^v

GitHub

GitHub er et av verdens mest brukte nettbaserte repository hosting-services. Utviklere kan laste opp et prosjekt i en repository (lagringssted) og gi andre utviklere tilgang til denne slik at flere utviklere kan jobbe med samme prosjekt samtidig.

GitHub er basert på noe som kalles et distribuert system. Det vil si det finnes en master-repository som inneholder den originale versjonen av et prosjekt. Hver utvikler må dermed laste ned sin egen versjon av masteren og arbeide på deres egne lokale versjon. Dermed bruker man kommandoer som pull (last ned master-repository), commit (send endringene gjort i lokale versjon), push (laste opp den lokale versjonen til master), merge (dersom flere endringer har blitt gjort av ulike utviklere må de ulike versjonene slås sammen) for å vedlikeholde og interagere med prosjektet. GitHub har også en GUI (Graphical User Interface) man kan forholde seg til.

6 ANALYSE AV VIRKNINGER

Denne delen vil forklare de punktene nevnt i avsnitt [5 Løsninger](#). Her vil vi forklare hvorfor vi bruker de ulike utviklingspråkene og -verktøyene, og ikke minst hvilke virkninger det har på sluttproduktet.

6.1 FRONT-END

Valget av front-end-språkene HTML og CSS er kjernen i hver nettside og vil dermed være påkrevd å bruke for å utvikle en webbløsning. Valget ved bruken av CSS-rammeverket, Bootstrap, er for at nettsiden skal få en ryddig og komfortabel følelse. Bootstrap sine maler er universelt utformet noe som gjør at webbløsningen vil være automatisk tilpasset mot alle typer enheter.

JavaScript/JQuery brukes for å gjøre webbløsningen mer dynamisk som å få til overganger og animasjoner som gir en mer naturlig og behagelig flyt ved interaksjon mellom bruker og webbløsningen. Disse språkene er brukt på alle typer webbløsninger og er godt mulig at det ikke finnes en webbløsning som ikke bruker disse språkene.

AJAX brukes for å gi bedre flyt for kommunikasjonen mellom front-end og back-end. AJAX oppdaterer deler av websiden uten at hele websiden må lastes inn på nytt, dette vil da lette på belastningen av systemet. Et konkret eksempel er hvis man registrer en bruker, vil kun delen som er relevant for registreringen oppdateres med en gang uten at hele siden må lastes inn på nytt for at seksjonen skal oppdateres.

JSON har som hensikt å sende back-end-informasjon til front-end på tekstformat i stedet for dataobjekter. Dette vil akkurat som AJAX lette på belastningen av systemet og kommunikasjonen mellom partene og ikke minst være mer forståelig for mennesker fordi formatet er basert på menneske-forståelige tekst enn AJAX sitt originale format, XML.

6.2 BACK-END

Hovedgrunnen til at vi bruker PHP som det grunnleggende hovedspråket er fordi det var et ønske fra oppdragsgiveren, men samtidig er det et veldig åpent, lett og fleksibelt språk. PHP er velintegret for dynamiske webbløsninger, det gjør at kombinasjonen og samarbeidet med front-end språkene blir enda bedre og ikke minst optimal for en webbløsning.

Designmønsteret MVC blir brukt for å fordele webbløsningen i tre oversiktlige deler, slik at vi vet konkret hvor hver eneste del tilhører og deres ansvarsområder. Ved bruk av et slikt designmønster vil man kunne holde god orden og minske sannsynligheten for at endringen av den del vil påvirke flere ulike deler av systemet.

Utviklingsverktøyene Netbeans og Sublime Text har sine fordeler og ulemper. Vi er en gruppe på 4 og har utviklerverktoy som vi foretrekker. Netbeans er en IDE som gir masse støtte til utviklere, mens Sublime Text er en teksteditor som er ren og oversiktlig. På grunn av våre ulike preferanser vil vi bruke våre egne foretrukne utviklingsverktøy for en optimal personlig programutvikling.

I denne webbløsningen bruker vi XAMPP for å kjøre en lokal Apacheserver og MySQL database. Apacheserveren brukes til å teste og kjøre webbløsningen vår under utvikling og MySQL databasen brukes til å lage en lokal database som vi kan bruke for å teste sikkerheten og lagring av informasjon under utviklingen.

Prosjektet blir da lagret i skyen, nemlig på GitHub. Vi bruker GitHub til å synkronisere vår individuelle utviklingsprosess av webbløsningen slik at vi til enhver tid alltid har de nyeste og oppdaterte versjonene av webbløsningen og vi kan utvikle hver for oss uten at det oppstår kollisjoner mellom ulike versjoner og vedlikeholdsproblemer.

6.3 OPPSUMMERING AV ANALYSE

Alle front- og back-end-språkene og -verktøyene som er nevnt ovenfor har vi brukt og eksperimentert med i tidligere prosjekter, noe som gjør at vi er mer eller mindre kjente og komfortable med dem. Noe som er grunnen til at vi valgte å bruke dem til å begynne med. Det at vi bruker kun kjente verktøy og språk resulterer i at vi ikke nødvendigvis får lært og utforsket nye språk og verktøy, men det vi får gjort er å fordype oss i de vi allerede kjenner til. Det er like bra med tanke på hvor sentralt de språkene og verktøyene vi bruker er i IT-verdenen.

Den ferdige webbløsningen vil resultere i at brukeren får den beste brukeropplevelsen. Webbløsningen skal være lett, oversiktlig, komfortabel å bruke, responsivt, rask og lite ressurskrevende å bruke. Dette gjelder for både sluttbrukere og administratorer som skal vedlikeholde og bruke dette via deres egen administrator brukergrensesnitt.

7 ARBEIDS- OG FREMDRIFTSPLAN

7.1 TABELLREPRESENTASJON

Oppgave	Varighet	Beskrivelse
Planlegging	05.01.17 - 31.01.17	I den første hovedfasen av bachelorprosjektet er det planleggingen som skal skje. Her er det viktig at gruppen planlegger det de kan i den minste detalj slik at man ikke overser viktige elementer. I selve planleggingen ferdiggjør vi dokumenter og krav som kreves for at vi skal kunne gå over til hovedfasen, som er utviklingen og sluttokumentasjonen.
Rapportering	05.01.17 - 24.05.17	Gjennom hele prosjektperioden skal det dokumenteres begivenheter. Det skal skrives styringsdokumenter og sluttokumentasjon. Det er derfor viktig at det blir dokumentert jevnlig slik at man ikke sitter igjen med mye skrivning, men i stedet fordeler det jevnt utover semesteret.
Forprosjekt	05.01.17 - 20.01.17	Før man skal sette seg ned og utvikle selve webløsningen skal forprosjektet gjennomføres. Dette er et innledende arbeid som forteller hva prosjektet er og hvordan vi tenker å gjennomføre det.
Kravspesifikasjon	20.01.17 - 15.02.17	Vi vil jobbe med kravspesifikasjonen i denne perioden, da vi vil være sikre på at vi har et klart bilde over hvordan sluttproduktet vil se ut i samsvar med kravene til NC-Spectrum.
Design	09.01.17 - 01.03.17	I denne fasen skal kravspesifikasjonen være klar og vi kan videre planlegge design for webløsningen. Her lages skisser i begynnelsen frem til det oppstår enighet om et konkret design som skal brukes. Deretter kan vi basere webløsningen på et design som alle har blitt enige om.
Systemarkitektur	20.01.17 - 31.01.17	I denne delen vil vi beskrive hvordan vi tenker å sette sammen arkitekturen mellom komponentene i applikasjonen i form av et klassediagram.
Programmering/Utvikling	06.02.17 - 01.05.17	I denne fasen utvikler vi selve webløsningen. Hensikten er at alt av programmering skal gjøres i denne fasen. Dette inkluderer både front-end og back-end. Front-end baserer seg blant annet på designet og annet som er synlig for brukeren, mens back-end blant annet inkluderer datahåndtering, database og bakgrunnsprosesser.

Testing av webløsning	21.04.17 - 01.05.17	Etter at vi er ferdig med utviklingen skal testing gjennomføres ved bruk av ulike testmetoder. Testing benyttes for å være sikker på at sluttproduktet er optimalt og at det ikke inneholder feil og/eller mangler. Dette betyr at hele webløsningen skal fungere sammenhengende og at ulike funksjoner og elementer i webløsningen skal fungere uavhengig av hverandre.
Presentasjon	06.06.17 - 09.06.17	Når hele webløsningen inkludert all dokumentasjon er komplett og prosjektet er levert skal gruppen presentere prosjektet for sensor. Dette er det siste som gjøres før komplett fullføring av bachelorprosjektet.

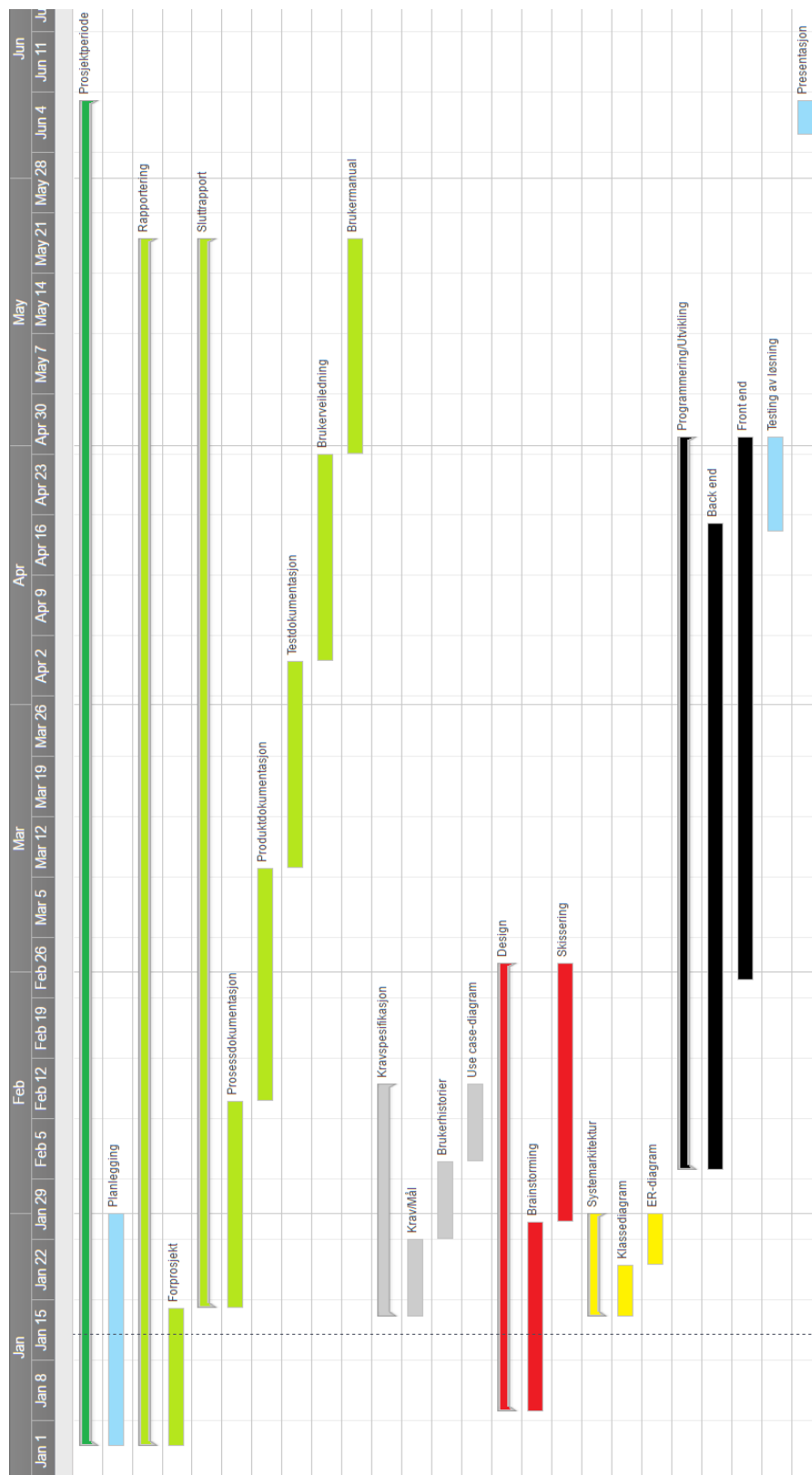
7.2 GANTT-DIAGRAM-REPRESENTASJON

Dette er et Gantt-diagram som viser en grafisk representasjon av tabellen ovenfor. Gantt-diagrammet skal illustrere tidsplanen til prosjektet. Det er verdt å legge til at dette er estimater og at endringer mest sannsynlig vil skje. I Gantt diagrammet har vi valgt å skille de ulike kategoriene i farger for å gjøre det mer oversiktlig. Vi har også hoved- og underpunkter. Søylen som har streker nedover på kantene er hovedpunkter. De som har "standardfargen blå", er ikke under en bestemt kategori. Gantt-diagrammet er utviklet ved hjelp av et verktøy på smartsheet.com.

[Se vedlegg for Gantt Diagram.](#)

8 VEDLEGG

8.1 GANTT DIAGRAM



ⁱ PHP Manual. (2017, January 18). Retrieved from <http://php.net/manual/en/preface.php>

ⁱⁱ Ajax: A New Approach to Web Applications. (2017, January 18). Retrieved from <http://adaptivepath.org/ideas/ajax-new-approach-web-applications/>

ⁱⁱⁱ Introducing JSON. (2017, January 18). Retrieved from <http://www.json.org/>

^{iv} Trygve Reenskaug & James O. Coplien. (2017, January 17). The DCI Architecture: A New Vision of Object-Oriented Programming. Retrieved from http://www.artima.com/articles/dci_vision.html

^v XAMPP. (2017, January 18). Retrieved from <https://www.apachefriends.org/index.html>