**Kravspesifikasjon**

***Vi kan begynne med å lage en innholdsliste som viser hvert fall alle underoverskriftene vi har med. Det trenger ikke å være så mange av de (jeg tenker min 5), vi må klare å skrive en helt tekst istedenfor å dele det opp med spm/svar.***

***Deretter kan vi ha en side hvor vi kommer med en liten innlending/forord. Det kan være greit å forklare litt hvem vi lager programmet til, hvem vi har laget det for, og eventuelt nevne hvem vi ser på som mottakere av dette dokumentet og kan si da at vi skriver det her med hensyn på at den som leser det har erfaring med ditt og datt. Her kan vi også nevne de som kommer til å lage programmet, og kanskje hvem/hva vi får hjelp av.***

***Så kan vi begynne med hovedteksten på en ny side. Her kan vi begynne å svare på enkelte av spm som muligens kan flettes inn i hverandre. Vi kan begynne med «generelt»:***

* **Hva skal programmet utføre? Hva slags funksjonalitet skal det ha? (use-case)**
* **Hvordan skal programmet bygges opp? Hvilke moduler kan det deles inn i?**

***Og da kanskje gå videre til hvordan vi vil at det skal se ut, og hvorfor vi vil ha det sånn. Både det brukeren ser, og hvordan selve kodingen vil se ut:***

* **Skisse over brukergrensesnittet:**
* **Hvilke klasser (egendefinerte) skal brukes/programmeres?**
* **Hvordan er klassene relatert til hverandre? Super-/subklasser?**
* **Hva slags datastruktur er valgt? (NB! Dere skal bruke Collection-rammeverket)**
* **Beskrivelse av datastrukturen hjelp av figurer**

***Deretter kan vi legge frem en plan, hva som må med og hva som kan bli med hvis tid. Her kan det være greit med noen gode begrunnelser til hvorfor:***

* **Prioriteringsrekkefølge: Hva må kunne utføres, og hva kan gjøres hvis dere får tid?**

***Helt til slutt kan vi ta med oversikt over utviklingsverktøy:***

* **Hvilket utviklingsverktøy skal brukes? (TextPad, Eclipse, Netbeans?)**

***Jeg tenker det kan være greit å ha en forside (Trenger ikke å være noe fancy), og sidetall er alltid viktig. Helst da av typen side x av y. Det kan også være greit å tenke på at vi skal selge produktet når vi skriver det her, så ikke bruke et muntlig språk, og få det til å se bra ut. Derfor får vi tegne så bra vi kan (evn tegne på pc hvis det er mulig). Det er ikke noe endelig løsning, bare et enkelt forslag!***

Vi valgte oppgaven KULTURHUS fordi vi synes den virket mest interessant og så ut som en morsom oppgave.

Dette skal vi ha med:

* Registrering
  + Registrering av lokale
  + Registrering av kontaktperson
  + Registrering av arrangement
  + Registrering av billettsalg
* Billettinntekter
* Kalender
* Sletting
  + Skrive ut en fil i tillegg til lista på skjermen
* Lagring
  + Til og fra fil
* Søkemuligheter og informasjon
  + All data om et valgt arrangement, lokale og en gitt person
  + Alle som skal delta

Dette skal vi ha med hvis vi har tid:

* Promotering
* Søkemuligheter og informasjon
  + Alle arrangementer med: navn, type, sted og tidspunkt
  + Alle arrangementer innen en valgt type
  + Alle kontaktpersoner innen en valgt type
  + Alle arrangementer i en gitt tidsperiode
* Statistikk
  + Totalt antall arrangementer (tidligere og fremtidige) arrangert i hvert lokale
  + Liste over de 10 mest inntektsbringende arrangementene

**Ukeplan**

**Prioriteringsrekkefølge: Hva må kunne utføres, og hva kan gjøres hvis dere får tid?:**

Uke 16:

* Ferdig med kravspesifikasjon

Uke 18:

* Ferdig med metoder osv. sånn at programmet er kjørbart
* Begynner med design og GUI

Uke 20:

* Bli ferdig med prosjektet og begynner på dokumentasjonen

**Hva skal programmet utføre? Hva slags funksjonalitet skal det ha?(use-case):**

Vi skal lage et program for et kulturhus. Det programmet skal utføre et hemmelig ! Tegner inn UML diagram !

**Skisse over brukergrensesnittet:**

(Kommer skisser her)

**Hvordan skal programmet bygges opp? Hvilke moduler kan det deles inn i?:**

* En hovedside(forside) som kundene bruker og ser
  + Et eget vindu for påmelding til arrangementer
  + Et eget vindu med info om et bestem arrangement
* En intern hovedside for ansatte
  + Et vindu for registrering av arrangement
  + Et vindu for registrering av lokaler
  + Et vindu for registrering av kontaktperson

**Hvilke klasser (egendefinerte) skal brukes/programmeres?:**

* En klasse for lokaler
* En abstrakt superklasse for arrangementer
  + Subklasse for kinoer
  + Subklasse for teater
  + Subklasse for konserter
* En klasse for kontaktpersoner
* En klasse for billetter
* En klasse for lokalregister
* En klasse for kontaktpersonregister
* Vindus-klasser for:
  + Hovedside(bruker)
  + Påmelding til arrangement
  + Info om et bestemt arrangement
  + Intern hovedside(ansatte)
  + Registrering av arrangement
  + Registrering av kontaktperson
  + Registrering av lokaler

**Hvordan er klassene relatert til hverandre? Super-/subklasser?:**

Arrangement er en abstrakt superklasse til subklassene: kino, teater og konsert. Disse klassene har igjen kontaktperson. Og så har lokalene en liste som inneholder alle arrangementer i det lokalet. Utenfor det igjen er det et lokalregister som inneholder en liste over alle lokalene. (tegner en tegning).

**Beskrivelse av klassehierarkiet ved hjelp av figurer:**

Slå sammen med den over

**Hva slags datastruktur er valgt?(NB! Dere skal bruke Collection-rammeverket):**

**Beskrivelse av datastrukturen hjelp av figurer:**

Sammen med forrige

**Hvilket utviklingsverktøy skal brukes?(TextPad, Eclipse, Netbeans?):**

Netbeans:

Vi har valgt å bruke Netbeans fordi vi syntes den var grei å bruke, ganske oversiktlig og et hjelpemiddel til prosjektet.

GitHub:

Vi har valgt å bruke GitHub fordi dette var den mest kjente git-tjenesten og begge har brukt det før og er kjent med det. Grunnen til at vi valgte å bruke en git-tjeneste er at det er mye mer oversiktlig en f.eks Dropbox. Med git har du en detaljert

Systemkrav til maskinen programmet skal kjøres på

Opplysninger om Java-versjon (NB må bruke java 1.8)

Fremdriftsplan med milepæler

Planlagt arbeidsfordeling. (NB! Programmeringsoppgavene skal fordeles tilnærmet likt mellom gruppens medlemmer.)