

# Organic Donkey Hostel - Hjemmeside med booking

## Projektkursus Systemudvikling 2015

1. juni 2015

Projektgruppe-nummer 1, gruppemedlemmer:

Jeppe Schönemann Skov 240992

Rose Sofie Greve 250493

Frederik Leed Henriksen 110394

Instruktor: Christoffer Wadum

# Indhold

<b>1</b>	<b>Abstract</b>	<b>3</b>
<b>2</b>	<b>IT-projektets formål og rammer</b>	<b>4</b>
<b>3</b>	<b>Kravspecifikationer for IT-løsningen</b>	<b>5</b>
3.1	a - Funktionelle og ikke funktionelle krav . . . . .	5
3.2	b - Use case model . . . . .	6
3.3	c - Use cases . . . . .	7
3.4	d - Klassediagram over problemområdet . . . . .	8
3.5	e - BCE-model . . . . .	9
3.6	f - Sekvensdiagrammer over use cases . . . . .	10
<b>4</b>	<b>Systemdesign sammenfatning</b>	<b>12</b>
<b>5</b>	<b>Program- og systemtest</b>	<b>12</b>
<b>6</b>	<b>Brugergrænseflade og interaktionsdesign</b>	<b>13</b>
6.1	Eksempler på brugergrænseflade . . . . .	13
6.2	Dynamikken i brugerinteraktion . . . . .	15
6.3	Audio-visual præsentation . . . . .	16
<b>7</b>	<b>Projektsamarbejdet</b>	<b>16</b>
<b>8</b>	<b>Litteraturreview</b>	<b>17</b>
8.1	Review af The M.A.D experience, Christensen et al. (1998) . . . . .	17
8.2	No silver bullet . . . . .	17
<b>9</b>	<b>Bilag</b>	<b>19</b>
9.1	Bilag 1: Versionsstyring . . . . .	19
9.2	Bilag 2: Changelog for projektrapporten . . . . .	19
9.3	Bilag 3: Timeline . . . . .	19
9.4	Bilag 4: Test . . . . .	19

# **1 Abstract**

We are making a website for a small Bed and Breakfast on Isla Margarita, Venezuela. The place is working towards becoming a sustainable and self-sufficient mini-hostel, where guests can enjoy organic greens from the garden. The frontpage on the website will contain a description of the place and the surrounding area. The website will contain a simple booking- and online payment system. The website will have a link with pictures of the houses and surrounding facilities, and a link with information on the different activities taking place at the hostel. With courtesy to the place working with ecology and sustainability, the layout of the website will be simple and inspired by the colors of nature. Further, we will make an administration part of the system, from where our costumer can see and edit the bookings. The administrator will also be able to edit the website photos and text descriptions.

## 2 IT-projektets formål og rammer

Her følger en beskrivelse af rammerne for vores IT-projekt, baseret på FACTOR-begrebet.

**Functionality:** Skal indeholde et booking- og betalingssystem, hvor administratoren har mulighed for at redigere i bookingerne. Samt redigere i de billeder og den information der fremvises på hjemmesiden.

**Application domain:** Skal administrere bookninger, herunder redigere og slette bookinger samt finde kontaktoplysninger på de personer der har oprettet en booking.

**Conditions:** Vi skal udvikle et simpelt system, da de der skal betjene systemet har ikke stor teknisk erfaring.

**Technology:** Systemet skal både udvikles og kunne betjenes på en billig pc med internet. Grundet hyppige strømsvigt skal databasen ligge på en webserver.

**Objects:** Personer der ønsker at booke værelse. Administrering af bookinger og websiden.

**Responsibility:** Systemet skal løse administrative problematikker i henhold til at holde styr på kontaktoplysninger af kunder, holde styr på hvilke værelser der er ledige, og hvornår disse er ledige. Systemet skal også reklamere for hostel'et.

## 3 Kravspecifikationer for IT-løsningen

### 3.1 a - Funktionelle og ikke funktionelle krav

Dette afsnit angiver de funktionelle og ikke funktionelle krav, samt de begrænsninger vi har til vores system.

#### **Funktionelle krav:**

- Side med bookingsystem der giver brugeren mulighed for at booke senge
- Administrativ side der giver vores kunde mulighed for at redigere i bookinger, tekst og billeder på brugersiden
- Loginsystem til den administrative side
- Betalingssystem. Et eksternt system som kunden viderestilles til. Ikke et betalingssystem vi selv koder.

#### **Ikke-funktionelle krav:**

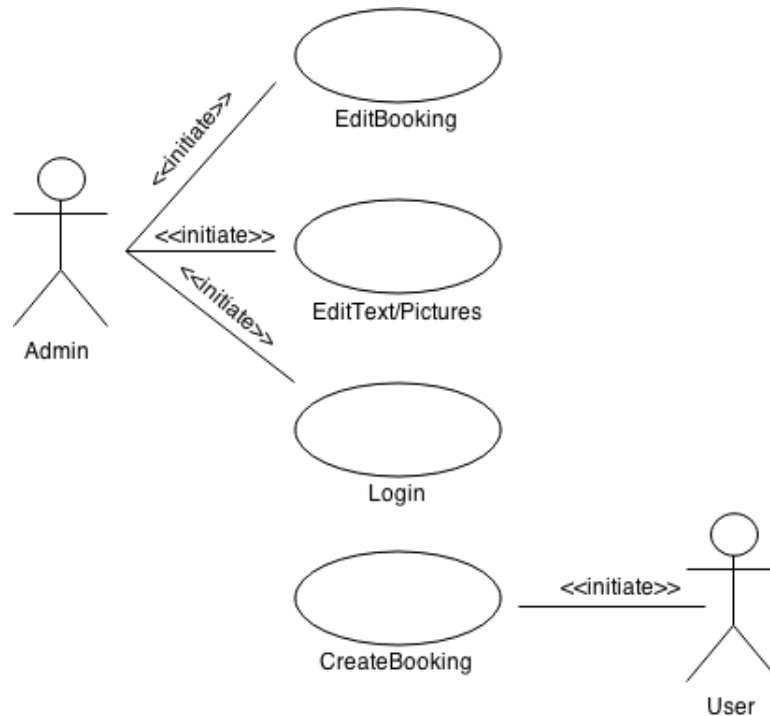
- Simpelt system som kan benyttes af folk uden særlig teknisk erfaring
- Side med information om stedet
- Side med billedegalleri
- Side med kontaktinformation
- Layout der afspejler hostelets stil og holdning til miljøet.

#### **Begrænsninger:**

Siden skal kunne vises på en billig computer og uden hurtigt internet til rådighed. Databasen med bookinger skal ligge på en webserver, da strømsvigt er hyppige i området.

### 3.2 b - Use case model

Dette afsnit beskriver kort det mest essentielle ved systemets use cases, og illustrerer sammenhængen mellem disse og aktørerne i et højniveau-diagram.



Figur 1: Et højniveau-diagram der illustrerer hvilke use cases de forskellige aktører har

**EditBooking** Administratoren kan slette og redigere i værelsesbookingerne. Login er krævet for at kunne redigere i bookinger.

**EditText/Pictures** Administratoren kan slette og redigere i teksten og billederne på hjemmesiden. Login er krævet for at kunne redigere i hjemmesiden.

**Login** Administratoren skal logge ind på Admin siden, for at se og redigere bookinger samt tekst og billeder.

**CreateBooking** Brugeren kan booke et værelse på hostelet gennem hjemmesiden.

### 3.3 c - Use cases

De følgende tre tabeller angiver mere detaljerede beskrivelser af vores tre vigtigste use cases. Diagrammerne illustrerer relationerne mellem aktørerne og systemet.

Tabel 1: Højniveau use-case-diagram over use casen Login

Use-case-name	Login
Participating actors	Administratoren
Flow of events	<ol style="list-style-type: none"><li>1. Administratoren indtaster sit brugernavn og kodeord på admin log-in siden og trykker på log-in knappen</li><li>2. Admin-databasen bliver spurgt om de indsendte oplysninger er rigtige.</li><li>3. Hvis der findes et match, bliver man videreført til admin-siden. Hvis ikke får man en fejlmeddelelse.</li></ol>
Entry conditions	At man ikke allerede er logget ind
Exit conditions	At man er logget ind

Tabel 2: Højniveau use-case-diagram over use casen EditBooking

Use-case-name	EditBooking
Participating actors	Administratoren
Flow of events	<ol style="list-style-type: none"><li>1. Administratoren indtaster den ønskede ændring for en booking</li><li>2. Ændringen bliver sendt til databasen og bookingens detaljer redigeres</li><li>3. Tabellen opdateres med de nye oplysninger</li></ol>
Entry conditions	At man er logget ind
Exit conditions	At den ønskede ændring i bookingen er sket

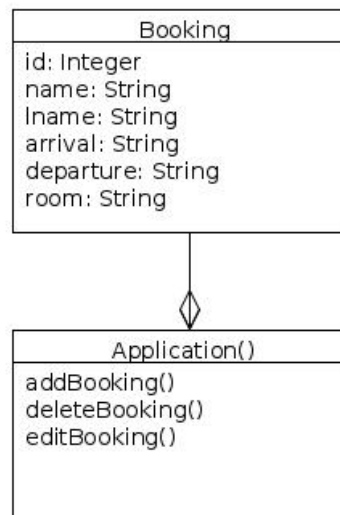
Tabel 3: Højniveau use-case-diagram over use casen CreateBooking

Use-case-name	CreateBooking
Participating actors	Brugeren
Flow of events	<ol style="list-style-type: none"> <li>1. Kunden indtaster sine ønskede booking datoer og kontaktinformationer</li> <li>2. Kunden trykker på booking-knappen og bookinginformationerne bliver tjekket for at verificere at bookingen ikke overlapper med andre bookinger</li> <li>3. Hvis datoerne for bookingen godkendes, bliver bookingen oprettet, og kunden bliver videresendt til "home"siden. Hvis der er overlap får kunden en fejlmeddelelse.</li> </ol>
Entry conditions	At brugeren befinder sig på bookingsiden
Exit conditions	At brugerens booking gemmes i databasen

### 3.4 d - Klassediagram over problemområdet

Dette afsnit illustrerer vores problemområder i et klassediagram.

Figur 2 illustrerer et klassediagram over problemområdet bookingsystemet. Application() tilføjer, redigerer og sletter bookinger fra bookingdatabasen. Booking definerer hvilke parametre en booking skal indeholde.



Figur 2: UML klassediagram over bookingsystemet



### 3.5 e - BCE-model

Dette afsnit indeholder en BCE-model over vores system.

Tabel 4: Boundary

Admin log-in side	Web-siden for administratoren som benyttes til at verificere at personen er administrator
Kunde booking web-side	Her har kunden adgang til at oprette en booking samt navigering til de andre kunde sider
Kunde "home"web-side	Her kan kunden navigere til de andre sider
Kunde galleri web-side	Her kan kunden navigere til de andre sider
Kunde kontakt web-side	Her kan kunden navigere til de andre sider
Kunde aktiviteter web-side	Her kan kunden navigere til de andre sider

Tabel 5: Control

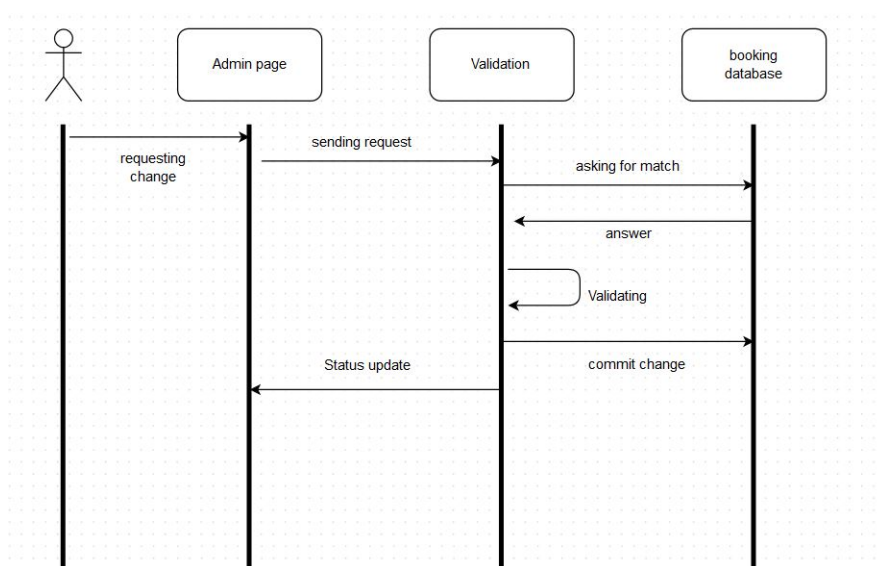
Admin log-in verificeringsknappen	Facilliterer forespørgelsen mellem admin log-in siden og admin-databasen. Hvis personen er en admin, bliver vedkommende sendt videre til admin siden. Hvis ikke, sker der ikke noget
Opret booking knap	Søger databasen for overlap i dato. Hvis der ikke er sådanne konflikter skabes en booking med de givne informationer i databasen. Hvis der er en konflikt vises en passende fejlmeddelelse
Rediger booking knap	Redigerer i en allerede oprettet booking i databasen
Rediger web-sider	Redigerer kundesidernes indhold i forhold til billeder og tekst (Vi ved endnu ikke hvordan dette skal implementeres)
"Home"link	Sender kunden til "home"web-page
Booking link	Sender kunden til booking web-page
Galleri link	Sender kunden til galleri web-page
Kontakt link	Sender kunden til kontakt web-page
Aktiviteter link	Sender kunden til aktiviteter web-page

Tabel 6: Entity

Admin databasen	Indeholder oplysninger om hvilke brugere der eksisterer i systemet
Administrator siden	Her har administrator mulighed for at tilse aktuelle bookinger og rette i databasen
Booking database	Indeholder alle bookinger på siden

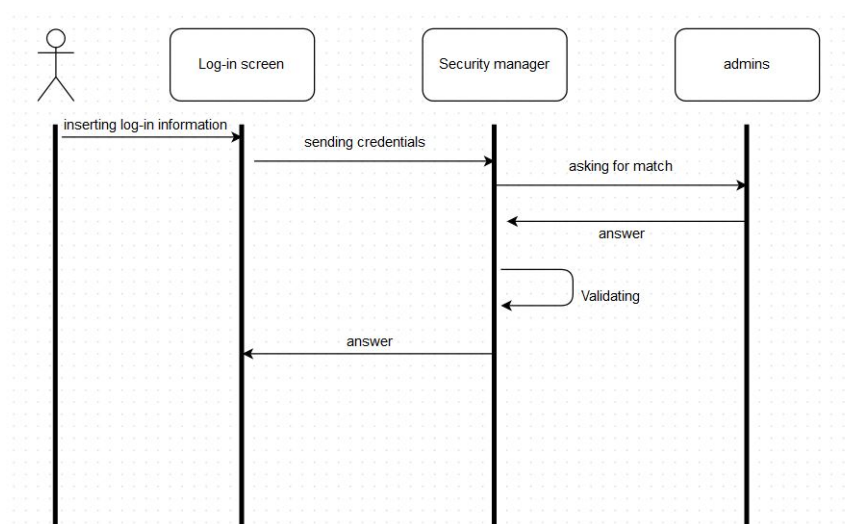
### 3.6 f - Sekvensdiagrammer over use cases

De følgende sekvensdiagrammer illustrerer interaktionen mellem aktøren og systemet i en use case.



Figur 3: Sekvens-diagram over use-casen EditBooking

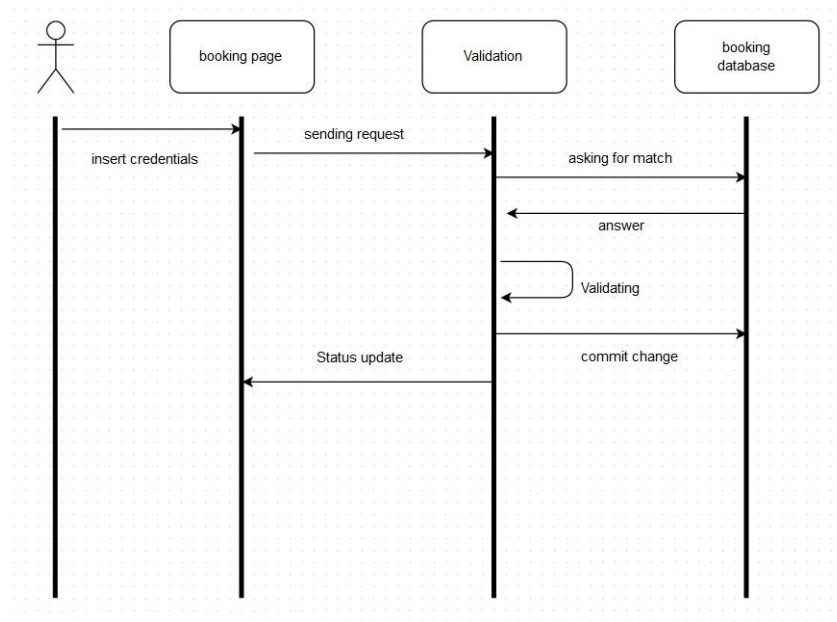
Figur 3: Brugeren forespørger om en ændring fra admin siden. Fra admin siden sendes en forespørgsel til en funktion, der validerer denne. Funktionen sender en forespørgsel til databasen, som svarer. Hvis valideringsfunktionen evaluerer til "sand", bliver ændringen i databasen gennemført. Hvis ikke bliver det skridt sprunget over. Man får en statusopdatering, der beskriver, om ændringen bliver udført eller ej.



Figur 4: Sekvens-diagram over use-casen Login

Figur 4: Brugeren indtaster sine informationer på login skærmen. Informationerne sendes til en sikkerhedsvalideringsfunktion. Funktionen forespørger om et match i admin-databasen. Denne sender

svar tilbage. Sikkerhedsvalideringsfunktionen validerer forespørgelsen. Svaret bliver sendt til login skærmen.



Figur 5: Sekvens-diagram over use-casen CreateBooking

Figur 5: Brugeren indtaster information på bookingsiden. Informationen bliver sendt til en valideringsfunktion. Denne forespørger, om der er konflikter i databasen. Databasen svarer, og valideringsfunktionen validerer forespørgelsen. Status på forespørgsel vises på bookingsiden.

## 4 Systemdesign sammenfatning

**Systemet som det er nu:** Vi har benyttet Play! frameworket til at lave en meget fin prototype af systemet. Det er nu muligt at navigere mellem alle de forskellige sider på hjemmesiden. Vi har fået implementeret en billede-slider på siden *Gallery*, som præsenterer billeder af stedet i et diasshow. På siden *Contact* har vi implementeret et google-maps kort, der viser hostelets beliggenhed. Bookingsystemet er også blevet konstrueret, og det er nu muligt at tilføje en booking fra bruger-siden, samt tilføje og slette bookinger fra admin-siden. Bookingerne gemmes i en database.

**De væsentligste mangler:** Vi mangler en funktion, der gør det muligt for administratoren at ændre en booking. Desuden mangler vi stadig at lave et login-system til admin-siden.

Vi skal også gøre det muligt for kunden, selv at bestemme hvilke billeder der fremvises samt redigere i teksten på de forskellige sider. Endeligt mangler vi at koble bookingsystemet til et betalingssystem.

## 5 Program- og systemtest

Resultetet af vores tests er som følgende:

- Vi kan navigere rundt imellem vores forskellige sider.
- Vi kan oprette og slette bookninger.
- Vi kan sortere i bookningernes representation.

## 6 Brugergrænseflade og interaktionsdesign

Dette afsnit omhandler brugergrænsefladen og interaktionsdesignet, for vores system.

### 6.1 Eksempler på brugergrænseflade

Følgende billeder viser brugergrænsefladen på hjemmesiden.

I figur 6 ses vores bookingside hvorfra man kan oprette en booking. Det forløbige design er således, at der udover den oblikatoriske menu-bar også er fire tekstfelter, en "drop-down" menu og en knap. Disse felter skal indsamle det data, der bliver til en oprettet booking (Indholdet af en booking er ikke færdigt endnu, der vil altså være andre/anderledes parametre i slutproduktet).

The screenshot shows the 'Organic Donkey' booking interface. At the top right is a navigation bar with links: Home, Activities, Booking, Gallery, and Contact. Below the navigation bar is the 'Organic Donkey' logo. The main form contains the following elements:

- Name: Fredrik
- Last name: Henriksen
- From: 05/19/2015
- to: (empty)
- Room: Room 1
- Submit Query button

Below the form is a calendar view for May, June, and July 2015. The calendar shows dates from Sunday to Saturday. The date 19th of May is highlighted in orange, indicating the selected date for the booking.

Figur 6: Opretter en booking. Her ses en booking der er ved at blive lavet, hvor brugeren er ved at vælge dato

I figur 7 ses det layout som går igen på de fleste sider med overskrift, menu-bar og tekst.



Figur 7: Index siden. Her ses vores sides forside

I figur 8 ses den forløbige admin side hvor administrator har overblikket over bookinger. Administratoren kan søge og slette i bookinger.

Home

Activities

Booking

Gallery

Contact

Showing 1 to 10 of 30 entries

Name	Last Name	Arrival	Departure	Room	Delete
asdf	asdf	05/13/2015	05/30/2015	101>	X
asdfsdf	asdrrenw	05/22/2015	05/26/2015	103>	X
asdfsdf	asdrrenw	05/22/2015	05/26/2015	103>	X
asdfsdf	asdrrenw	05/22/2015	05/26/2015	103>	X
asdfsdf	asdrrenw	05/22/2015	05/26/2015	103>	X
asdfsdf	asdrrenw	05/22/2015	05/26/2015	103>	X
asdfsdf	asdrrenw	05/22/2015	05/26/2015	103>	X
asdfsdf	asdrrenw	05/22/2015	05/26/2015	103>	X
asdfsdf	asdrrenw	05/22/2015	05/26/2015	103>	X
asdfsdf	asdrrenw	05/22/2015	05/26/2015	103>	X

Name

Last Name

Arrival

Departure

Room

Delete

Show 10 entries

Previous

1

2

3

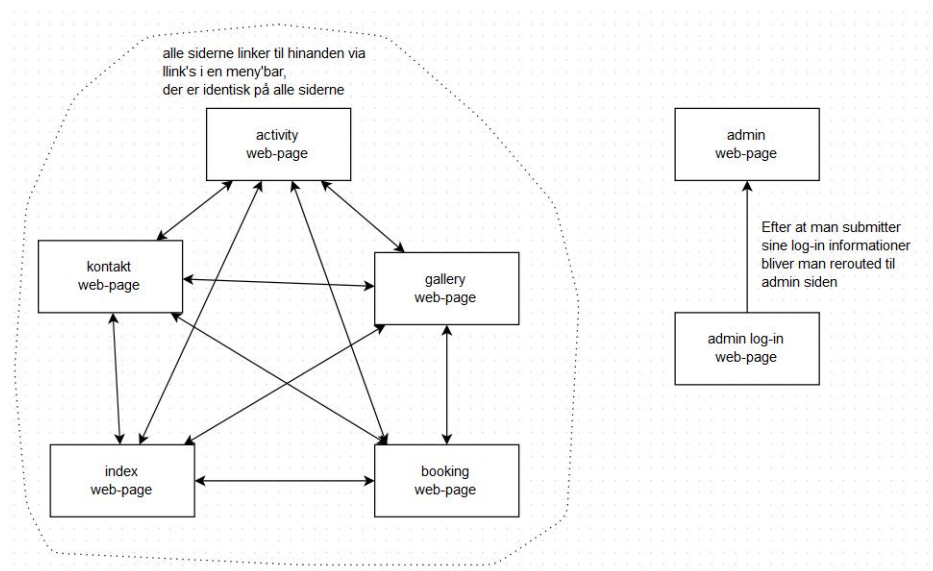
Next

Search:

Figur 8: Oprettede bookinger. Her ses admin-siden hvor administratoren kan se og redigere bookinger

## 6.2 Dynamikken i brugerinteraktion

I figur 9 ses rutediagrammet over, hvordan brugere af siden kan bevæge sig imellem de forskellige views. Kunden betjener siderne gennem menuen. Administratoren skal logge ind for at administrere bookingerne.



Figur 9: Flowchart over siden. Her ses rutediagrammet over siden, som den fungerer på nuværende tidspunkt

### **6.3 Audio-visual præsentation**

Følg linket for at se en audio-visual præsentation af brugergrænsefladen af vores seneste prototype:  
<https://youtu.be/CF55ggqy9M8>

## **7 Projektsamarbejdet**

Internt i gruppen arbejder vi godt sammen. Cirka halvvejs inde i projektet aftalte vi at sætte mere tid af til projektet. Siden da er vi nået rigtig langt med systemet.

Vi har indtil videre haft fire møder med kunden. Møderne foregår over Skype, da kunden bor i Venezuela. På trods af at afstanden kan være upraktisk, er kunden fleksibel med dato for møder, da vi blot mødes over Skype.

Under møderne tager vi referat i form af stikord, og dokumenterer de aftalte ændringer i projektaftalen og projektplanen, i rapporten.



## 8 Litteraturreview

### 8.1 Review af The M.A.D experience, Christensen et al. (1998)

The M.A.D experience er en artikel omhandlende en research gruppe fra Århus universitet, der ingår et samarbejde med et stort globalt shippingfirma om at lave en prototype til et kundeservicesystem. Artiklen handler, om hvordan gruppen får designet og implementeret en rigtig god prototype, der endda overgår shipping firmaets forventninger. Dette uden at gruppen havde nogen som helst form for ekspertise inden for shippingområdet. Artiklen diskuterer, hvilke delelementer der spillede en rolle for denne succesfulde udførelse af projektet. Gruppemedlemmerne havde tre forskellige roller: en etnograf, en partecipatorisk designer og seks objektorienterede designere.

Etnografen står for at udforme et design, der sørger for, at systemet relaterer til det miljø, projektet omhandler. Han identificerer problemløsningsområdet men ikke selve designløsningerne.

Den partecipatoriske designer arbejder med vigheden i at involvere potentielle brugere af systemet i den kreative designprocess. Dette gøres både ud fra et moralsk og et praktisk grundlag, da det er rimeligt at lade de, som i sidste ende skal bruge systemet, være med til at tage beslutninger, og at disse personer samtidig er de mest praktiske forsøgspersoner at have med ind over projektet. Den partecipatoriske designer går op i at afholde workshops for forskellige mennesker, der har en eller anden tilknytning til projektet, for derefter at indrette systemdesignet udfra den konstruktive kritik de modtager under disse workshops.

De objektorienterede designere integrerer analyserne og designet fra etnografen og den partecipatoriske designer i det endelige design og implementering af systemet.

Udover disse tre elementer kommenterer artiklen på, at det er vigtigt, at man kommer igang med programmeringen så tidligt som muligt, bl.a. så man har noget at vise frem og få feedback på fra kunden og brugerne. Men også fordi at det i deres projekt, som også diskuteret i A Rational Design Process <sup>1</sup>, ikke stod klart fra start af, præcis hvordan systemet skulle udformes. De fandt, at man altid ville ende op med et andet design end den initiale design struktur, blandt andet på grund af den feedback de fik fra kunden og brugerne.

Det konkluderes, at et succesfuldt projekt kan opnås ved at have fokus på analyse af det miljø og det praktiske, som systemet omhandler, og skal henvende sig til. Disse elementer skal optræde som en integreret del af implementeringen af projektet.

### 8.2 No silver bullet

No Silver Bullet – Essence And Accidents in Software Engineering handler om software udvikling og dets vanskeligheder (essentiell og utilsigtet) og løsninger på problemerne.

Han undersøger arten af programfejl, og egenskaberne af de løsninger som han selv kalder silver bullets.

Brooks mener, at den sværeste del af software udvikling er de arkitektoniske opgaver som specifikation, design og test af konstruktionen.

Vanskelighederne er delt op i essentielle og utilsigtet typer. De essentielle problemer er medfødte til software. En kompleks og præcis abstrakt konstruktion afledt af datasæt, dataelementer, algoritmer

---

<sup>1</sup>Parnas and Clements 1986: A Rational Design Process: How and Why to Fake It

og funktioner. De utilsigtede problemer er de problemer der opstår, når en udvikler er i gang med at udvikle softwaren men er ikke medfødte i selve softwaren.

Brooks opstiller de essentielle vanskeligheder som kompleksitet, overensstemmelse, foranderlig og usynlighed. De uvæsentlige problemer er ikke opstillet så klart.

Tre løsninger på utilsigtede problemer bliver diskuteret. Det er høj niveau programmerings sprog, multiprogrammering og forenet udviklings miljøer. Det største gennembrud for produktiviteten, pålideligheden og enkelhed af de tre løsningsforslag er høj niveau sprog. Høj niveau sprog giver stærke fælles konstruktioner til programmøren og reducere et niveau af software kompleksitet.

I vores projekt hvor vi skal udvikle en hjemmeside med et booking system til nogle sommerhuse i Venezuela. Har vi også benyttet os af et høj niveau sprog som gør udvikling for vores projekt en del lettere og er med til at optimerer vores program. OOSE bogen anbefaler også brug af høj niveau sprog til udvikling, da der er mange tilgængelige pakker der kan hjælpe til udviklingen.

For at samle vores projekt et sted der gør det let tilgængeligt for os alle og gør det muligt for os at udvikle på dele af programmet er github.

## 9 Bilag

### 9.1 Bilag 1: Versionsstyring

Link til vores commit-log på GitHub:

<https://github.com/fumpen/pksu/commits/master>

### 9.2 Bilag 2: Changelog for projektrapporten

Jeppe Schönemann Skov - JSS

Frederik Leed Henriksen - FLH

Rose Sofie Greve - RSG

Initialer	Dato	Afsnit	Ændring
JSS, FLH, RSG	23.04.15	Alle	Release edition 1.0
RSG	07.05.15	1, 2	Stavefejl, mere præcis beskrivelse
RSG	07.05.15	3.a	Krav i stikord, mere uddybende
FLH	07.05.15	3.b, 3.c, 3.e	Omskrivning og ændring i model
JSS	07.05.15	3.d	Opdateret klassediagrammet
RSG, JSS	07.05.15	3.f	Kommentarer til figurer
RSG, JSS	07.05.15	6, 7	Ændring i tekst
RSG	11.05.15	4, 5, 7	Ændring i tekst
FLH	11.05.15	6	Flowchart, skærbilleder, video
RSG	21.05.15	3.1 a	Ændring i tekst
RSG	27.05.15	3.1 b	Nyt diagram
RSG	27.05.15	3.1 c	Ændring af entry og exit conditions
FLH	27.05.15	3.1 f	Beskrivende tekst til diagrammerne

### 9.3 Bilag 3: Timeline

- Under første møde d. 19. februar 2015 fik vi lavet en projektaftale med kunden.
- Under andet møde d. 10. marts 2015 snakkede vi detaljer ift. bookingsystemet.
- Under tredje møde d. 15. april 2015 snakkede vi om valg af betalingssystem, samt om hvilke dele af projektet kunden helst så os nedprioritere i tilfælde af tidspres.
- Den 11 maj 2015 havde vi den første prototype. Denne var i stand til at oprette bookinger, slette bookinger, sortere bookinger og den havde næste alle de sider som der skal være i det endelige produkt.

### 9.4 Bilag 4: Test

Vi har manuelt testet:

- Navigationen mellem siderne

- Opretning/sletning af bookinger
- Sortering i bookninger.

Da der endnu ikke er implementeret yderligere funktionalitet, har vi endnu ikke testet mere.