

Organic Donkey Hostel - Hjemmeside med booking

Projektkursus Systemudvikling 2015

28. maj 2015

Projektgruppe-nummer 1, gruppemedlemmer:

Jeppe Schönemann Skov 240992

Rose Sofie Greve 250493

Frederik Leed Henriksen 110394

Instruktor: Christoffer Wadum

Indhold

1	Abstract	3
2	IT-projektets formål og rammer	4
3	Kravspecifikationer for IT-løsningen	5
3.1	a	5
3.2	b	6
3.3	c	7
3.4	d	8
3.5	e	9
3.6	f	10
4	Systemdesign sammenfatning	12
5	Program- og systemtest	12
6	Brugergrænseflade og interaktionsdesign	12
6.1	eksempler på brugergrænseflade	12
6.2	dynamikken i brugerinteraktion	15
6.3	audio-visual præsentation	16
7	Projektsamarbejdet	16
8	Litteraturreview	17
8.1	Designing for usability	17
8.2	A rational design process	17

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.

Changelog

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	21.05.15	3.1 a	Ændring i tekst
RSG	27.05.15	3.1 b	Nyt diagram

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 sider billeder og information om hostelet.

Application domain: Skal administrere bookninger, herunder til- og afmelde folk samt finde kontaktoplysninger på de folk der har oprettet en booking.

Conditions: Udvikling og endeligt brug af systemet skal være minded på, at dem der skal betjene det, ikke har stor teknisk erfaring. Vi skal derfor udvikle minded på, at de tekniske detaljer ikke er deres fokus.

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 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
- Betaingssystem. Et eksternt system som kunden viderestilles til. Ikke et betalingssystem vi selv koder.

Ikke-funktionelle krav:

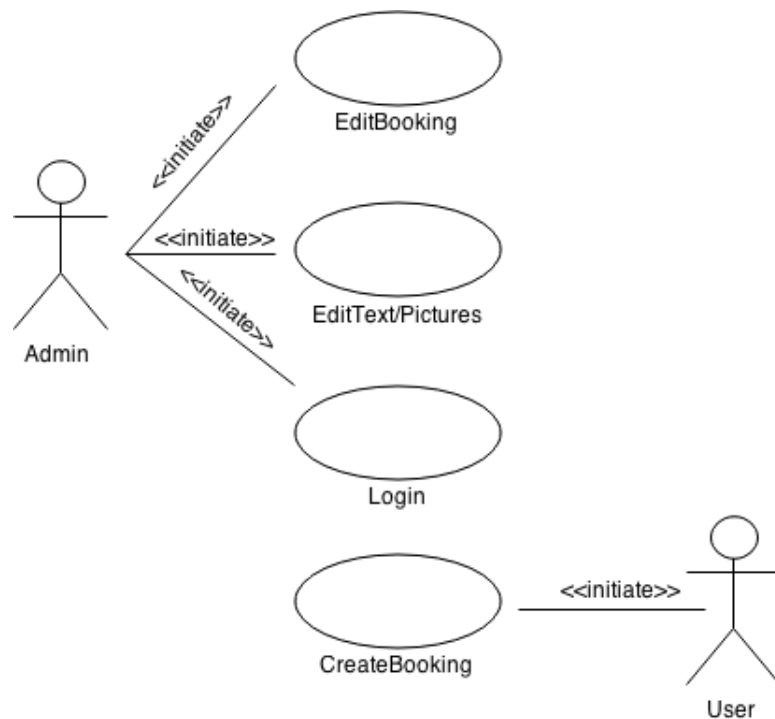
- Simpelt system, som kan benyttes af folk uden særlig teknisk erfaring
- Side med information om stedet
- Side med billedegalleri
- Side med kontaktinformation
- Layout der afspejler hostelets stil og holdning til miljøet.

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



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

Tabel 1: **Admin log-in**

use-case-name	Admin log-in
participating actors	administratoren
flow of events	<ol style="list-style-type: none">1. administratoren intaster sit log-in på admin log-in siden og trykker på log-in knappen2. admin databasen bliver spurgt om de indsendte oplysninger er rigtige.3. hvis der findes et match, bliver man videreført til admin-siden. Hvis ikke får man en fejlmeddelelse.
Entry conditions	ingen
Exit conditions	ingen

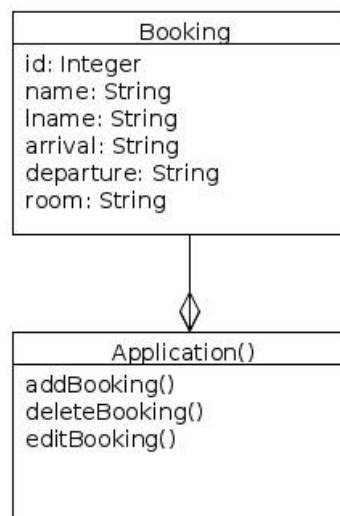
Tabel 2: **Admin slet booking**

use-case-name	Admin slet booking
participating actors	administratoren
flow of events	<ol style="list-style-type: none">1. administratoren trykker på slet knappen i tabellen ud for bookingen2. ændringen bliver sendt til databasen og bookingen slettes3. tabellen opdateres og bookingen er slettet og vises ikke mere
Entry conditions	admin log-in
Exit conditions	ingen

Tabel 3: **kunde booking**

use-case-name	kunde booking
participating actors	kunden
flow of events	<ol style="list-style-type: none"> 1. kunden indtaster sine ønskede booking datoer og kontaktinformationer 2. kunden trykker på booking knappen og booking informationerne bliver tjekket for at verificere om bookingen ikke overlapper med andre bookinger 3. hvis der ikke er nogen overlab, bliver bookingen oprettet og kunden bliver videresendt til "home"siden. Hvis der er overlab får kunden en fejlmeddelelse
Entry conditions	at kunden befinder sig på bookingsiden
Exit conditions	ingen

3.4 d



Figur 2: Booking system - UML

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

3.5 e

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

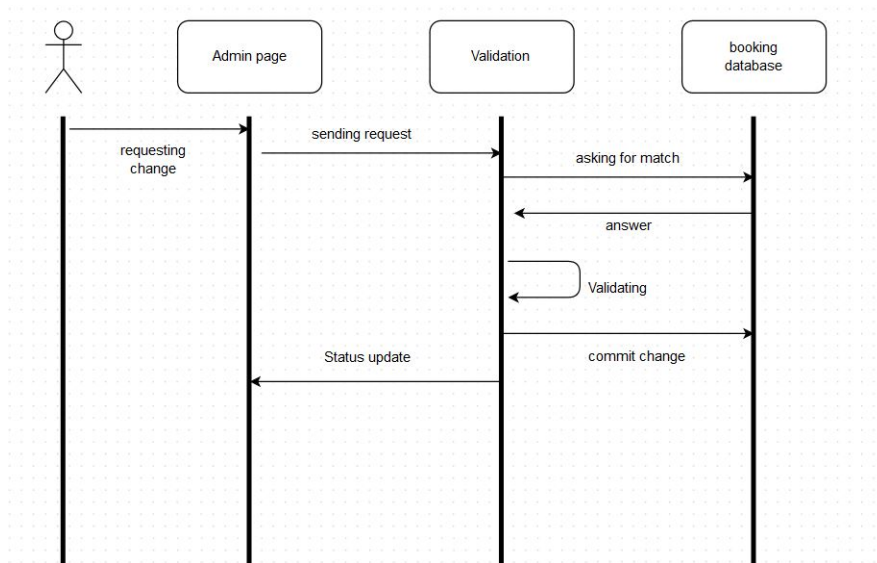
Tabel 5: **Control**

admin log-in verificeringsknappen	facilitere forespørgslen 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
fjern booking knap	fjerner en oprettet booking fra databasen
rediger web-sider	redigere 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**

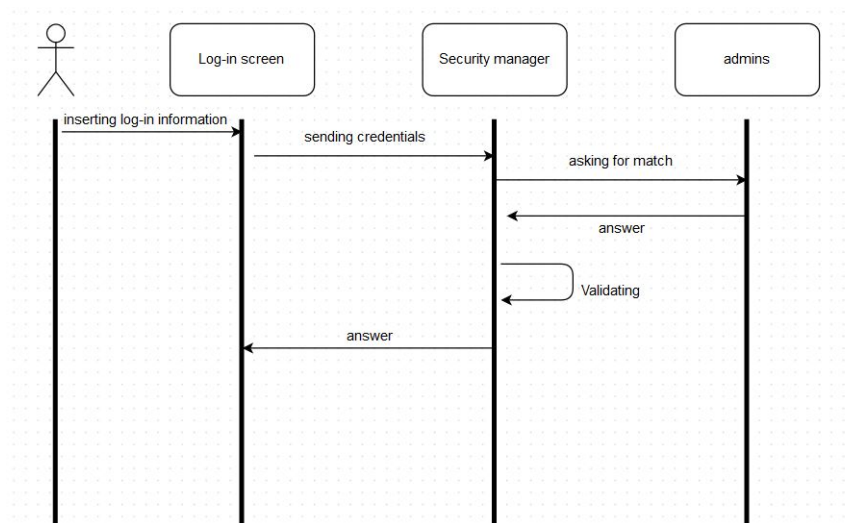
Admin databasen	indeholder oplysninger om hvilke brugere der eksistere i systemet
administrator siden	Her har administrator mulighed for at tilse aktuelle bookinger og rette i databasen
booking database	indeholder alle bookinger på siden

3.6 f



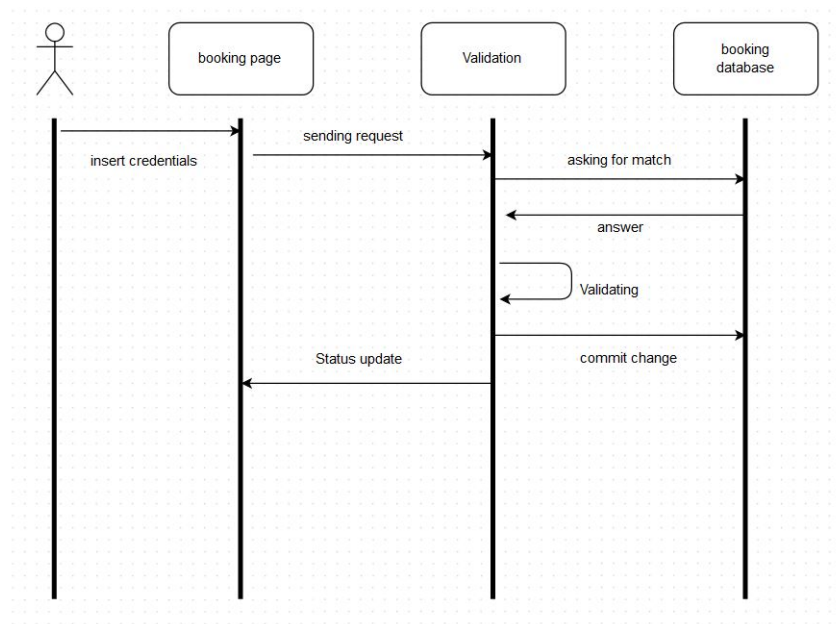
Figur 3: Admin Interaction

Figur 4 illustrerer interaktionen mellem den administrative side og bookingdatabasen.



Figur 4: Admin log-in

Figur 5 illustrerer valideringprocessen ved login på den administrative side.



Figur 5: Customer Interaction

Firgur 6 illustrerer processen når en bruger forsøger at oprette en booking.

4 Systemdesign sammenfatning

Systemet som det er nu: Vi har foreløbigt lavet skabelonen til hjemmesiden, som den endeligt skal se ud. Dette er gjort i html og css kode. Derudover har vi implementeret links, så man kan navigere mellem siderne.

De væsentligste mangler: En administrator side hvor man kan ændre billederne på siden. Derudover skal der også være et bookingsystem, som kunder kan oprette bookninger, og som administratoren via administrator siden kan oprette/slette bookninger og hente information om kunden. Desuden mangler der et betalingssystem på siden.

5 Program- og systemtest

Da vi endnu ikke har fået vores system op at køre, mener vi ikke at det, vi har indtil nu, ville være værd at tage med. I forhold til vores hjemmeside kan vi kun teste den visuelt ved at se, om linksne fører til de korrekte sider, og det gør de.

6 Brugergrænseflade og interaktionsdesign

6.1 eksempler på brugergrænseflade

i figur 6 ses vores booking side hvor man kan oprette en booking. Det foreløbige design er således at der udover den oblikatoriske meny-bar også er fire tekstfelter, en "drop-down" meny 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)

[Home](#)
[Activities](#)
[Booking](#)
[Gallery](#)
[Contact](#)

Organic Donkey

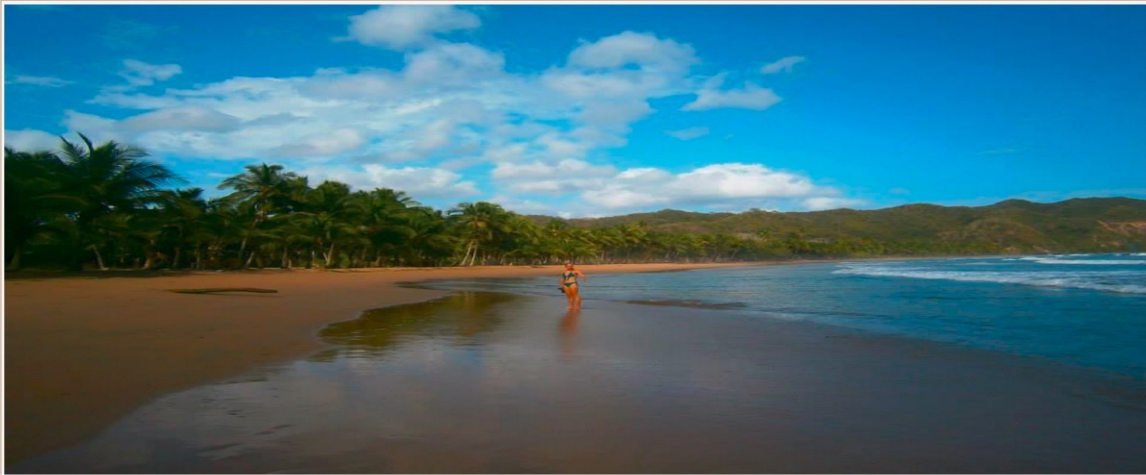
Name
Last name
From to
Room

May 2015							June 2015							July 2015							
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	
					1	2		1	2	3	4	5	6					1	2	3	4
3	4	5	6	7	8	9	7	8	9	10	11	12	13	5	6	7	8	9	10	11	
10	11	12	13	14	15	16	14	15	16	17	18	19	20	12	13	14	15	16	17	18	
17	18	19	20	21	22	23	21	22	23	24	25	26	27	19	20	21	22	23	24	25	
24	25	26	27	28	29	30	28	29	30					26	27	28	29	30	31		
31																					

Figur 6: opretter en booking
her ses en bookning der er ved at blive lavet, hvor brugeren er ved at vælge dato

I fugur 7 ses det layout som går igen på de fleste sider,
med overskrift, meny-bar og teskt.

Organic Donkey



Organic Hostel * Isla Magarita

April 20, 2015

adkwkdakdkwdawkdawkdawkdawkdakdkwdakdkwadakkdwak.

adkskdaskdksakdk. adkskdssakksak.

daowoakodkpwakpdao. dksadkaskdksadksakdskl.

Copyright © 2015

Figur 7: index siden
her ses vores sides forside

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

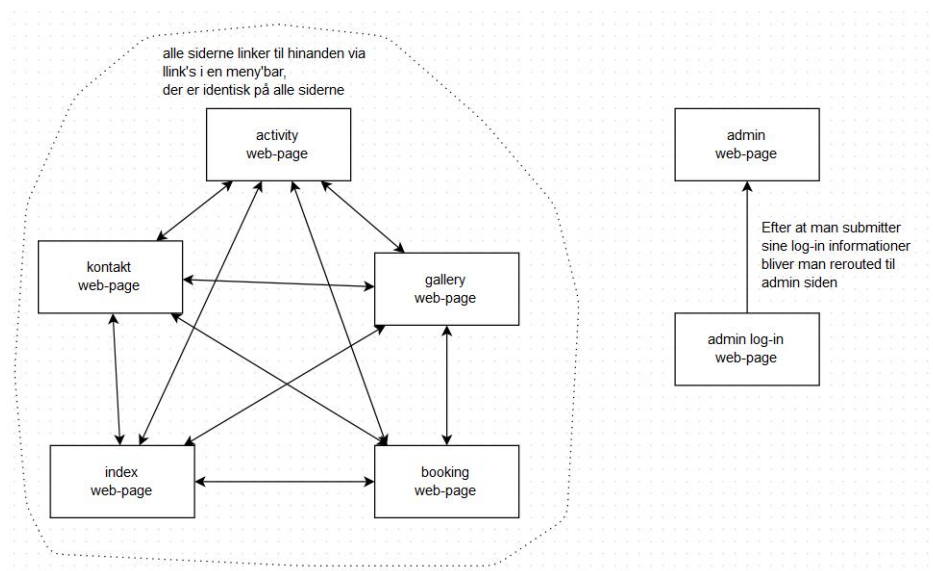
Next

Search:

Figur 8: oprettede bookinger
her ses admin siden hvor admin kan se og slette bookinger

6.2 dynamikken i brugerinteraktion

i figur 9 ses rutediagrammet over hvordan brugere af siden kan bevæge sig imellem de forskellige views. Dette er for kundens vedkommende gjort ved at bruge meny-baren. Det er for admin gjort ved at logge ind.



Figur 9: flowchart over siden
her ses rutediagrammet som siden fungerer på nuværende tidspunkt

6.3 audio-visual præsentation

7 Projektsamarbejdet

Internt i gruppen arbejder vi godt sammen. Vi syntes dog at vi begynder at få travlt, hvis vi gerne vil ende ud med et mere eller mindre færdigt produkt, og har derfor aftalt at mødes hver mandag og torsdag og arbejde sammen på projektet. Inden vi går hjem uddeler vi arbejdsopgaver, og aftaler hvad vi skal have nået hver især, inden vi mødes næste gang.

Vi har indtil videre haft tre 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.

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.

Vi har endnu ikke en fast dato for næste møde, da kunden som sagt er fleksibel, og vi derfor kan kontakte hende, når vi har brug for at diskutere projektet.

Vi har indset, at vi generelt set skal prioritere projektet højere, og vil derfor lave en tidsplan med deadlines til os selv. Første step bliver at lave en så detaljeret som muligt projektplan. Vi er motiverede af og glæder os til at gå igang med at kode, og lære hvordan et sådant system er bygget op.

8 Litteraturreview

8.1 Designing for usability

Denne artikel beskriver tre design principper, som Gould og Lewis mener er vigtige, men selv om de måske virker indlysende, bliver de ofte forsømt af udviklerne af flere årsager. Udviklerne undervurderer eller misforstår simpelthen nogle gange disse ”simple” principper. De tre principper: Fokus på målgruppe og opgaver. Hvor designeren må vide hvem målgruppen er og være i direkte kontakt med udviklerne, for at de kan analysere og studere målgruppens adfærd i stedet for at bare læse eller høre om dem.

Simulation og prototype. Målgruppen skal afprøve prototypen, hvor deres reaktion og ydeevne på systemet analyseres og observeres.

Test og omstruktur. Det sidste princip dækker over test af systemet og de fejl, som bliver fundet, skal rettes, og hvis det bliver nødvendigt, skal programmet omstruktureres og testes forfra.

Derefter uddyber de principperne, og forklarer, hvordan de kan anvendes i starten af design fasen efterfulgt af en test fase. De slutter deres diskussion ved at præsentere et case studie, hvor de anvender disse principper i et lyd besked system fra IBM.

Det er svært at argumentere mod vigtigheden af disse principper, som måske virker meget basale, men det kan være meget let undervurdere vigtigheden af disse principper. Object-Oriented Software Engineering Using UML, Patterns, and Java bogen beskriver seks faser, når man udvikler et system.

Krav Analyse System Design Objekt Design Implementation Test

Hvor den ligger meget vægt på samarbejde med kunden i form af modeller og grafiske illustration i stedet for at arbejde sammen med målgruppen, som jeg syntes artiklen ligger op til.

I vores projekt hvor vi har til opgave at designe og lave et online sommerhus booking system, forsøger vi at holde en løbende dialog med vores kunde. Vi fik først og fremmest nogle krav fra kunden, som vi skal overholde, men da de fleste var kommet på plads, og vi havde en nogenlunde idé om, hvad vi egentlig skulle lave i vores projekt. Dernæst har vi planer om at lave vores prototype af hjemmeside færdig, og derefter have et interview med vores kunde, for at se om de er tilfredse, eller om de har nogle ændringer. Efter det har vi planer om at nå ud til målgruppen med prototypen og høre deres mening om systemet, og om det er brugervenligt.

8.2 A rational design process

A Rational Design Process forklarer om hvad den rationelle design process er, og hvorfor det er så svært at lave en rationel struktur på sit projekt.

Kort sagt går et rationelt design ud på, at man stiller nogle helt præcise krav op for sit projekt. At man opdeler projektet i mindre og mindre detaljerede dele, indtil man har en fuldstændig plan for, hvordan man vil udføre sit projekt. Og at man dokumenterer alle beslutninger og ændringer, man foretager sig løbende.

Det er dog sjældent, at programmørens valg i den initielle designfase er særligt rationelle. Dette er der flere årsager til. Som vi selv har kunne mærke i vores projekt, vil der i starten ofte være uklarhed omkring, præcis hvad systemet skal kunne. Mange detaljer dukker altid op løbende under

design- og programmeringsprocessen, og det er derfor ikke muligt fra start af at lave en nøjagtig plan. Selv hvis det var muligt at kende til alle projektets detaljer på en gang, vil det simpelthen være for stor en mundfuld information, til at det er muligt for den menneskelige hjerne at lave et endeligt softwaredesign. Der sker ofte ændringer i, hvad systemet skal indebære. Disse ændringer har ofte betydning for designet, og man ville altså alligevel ende med et andet design end det første "rationelle design" man havde konstrueret. Endeligt sker det tit, at man skal viderearbejde allerede eksisterende software, som allerede har et design. Et rationelt design kan også udelukke muligheden for nye og anderledes ideer til design, inspireret af andre projekter, hvis design ikke nødvendigvis er særligt rationelt, men som på en anden måde virker optimalt for ens projekt.

På trods af alle disse stopklodser for en rationel design process er der også mange positive sider ved at forsøge at lave et rationelt design. Der researches derfor en del i en struktureret "opskrift" på software design. Ligger man tid og kræfter i at få lavet et godt solidt design, før man går i gang med at kode, bliver man tvunget til at reflektere over hele projektet, og tage stilling til hvilke udfordringer man måske støder på. Dette vil spare én tid i den anden ende.

Vi er opsatte på at forsøge os med at lave en detaljeret projektplan, da vi efterhånden begynder at forstå nødvendigheden af at have et godt overblik over det system, vi arbejder på. Vi vil også få lavet noget ordentlig dokumentation, for de valg vi tager, så vi på senere tidspunkter kan finde svar på, hvorfor vi valgte at gøre som vi gjorde. Dog er der mange ting, vi har svært ved at forestille os på dette tidlige stadie, især fordi at ingen af os har nogen programmeringserfaring. Det gør det f.eks. svært for os at vide, hvordan vi egentlig skal inddele projektet i moduler/subsystems, samt hvor lang tid hver del af projektet vil tage.

Vi anser opskriften på hvordan man laver et rationelt design som en slags guide og inspiration til, hvordan man kan gribe et projekt an. I forsøget på at følge opskriften på det rationelle design vil man eksempelvis lægge et større arbejde i at finde de små detaljer og risici ved projektet så tidligt som muligt og dermed undgå for meget back-tracking. Ved efterfølgende at skrive sin dokumentation og projektplan som hvis man havde fulgt det perfekte design, altså "fake" den ideelle process, bliver man stadig tvunget til at være mere reflekteret og forberedt, og gør dermed projektet og ens egen arbejde mere struktureret.