



SUBMISSION OF WRITTEN WORK

Class code: GRPRO

Name of course: GRUNDLÆGGENDE PROGRAMMERING

Course manager: DAN HANSEN

Course e-portfolio: GRPRO-AUTUMN 2014

Thesis or project title: KAFFEKLUBBEN

Supervisor: JESPER MADSEN

Full Name:

1. MARK ROSTGAARD MORTENSEN 09/04-1993 MROM@itu.dk

2. AMANDA ENGEL THILO 21/03-1993 AENT@itu.dk

3. FREDERIK HOVMAND JØRGENSEN 20/01-1993 FHOV@itu.dk

Birthdate (dd/mm/yyyy): E-mail:

4. _____ @itu.dk

5. _____ @itu.dk

6. _____ @itu.dk

7. _____ @itu.dk



EXAM PROJECT
Biobooking

KAFFEKLUBBEN

Amanda Engel Thilo

Frederik Hovmand Jørgensen

Mark Rostgaard Mortensen

DECEMBER 17, 2014

Abstract

In this report we address our workgroups attempt at designing a digital solution for broadening the audience for both the digital and physical offerings, that Statens Museum for Kunst(SMK) provides. We use a multitude of idea generation techniques, to arrive at a product, and then interviews with SMK. We propose an enhanced web-presence, for image viewing and discussion, based on feedback from SMK and visits to the museum. The report concludes with the proposal of a system in which users can comment and interact with the different artworks on the website.

Indhold

| | | |
|-------|---|----|
| 1 | Forord | 3 |
| 2 | Indledning | 4 |
| 2.1 | <i>Baggrund</i> | 4 |
| 2.2 | <i>Problemstilling</i> | 4 |
| 2.2.1 | Krav til programmet | 4 |
| 2.2.2 | Brugerscenario | 4 |
| 2.2.3 | Systemdesign | 5 |
| 3 | Problemanalyse | 6 |
| 3.1 | <i>Vores løsning</i> | 6 |
| 3.1.1 | Alternative løsninger | 6 |
| 3.2 | <i>Databasedesign</i> | 6 |
| 4 | Brugervejledning | 7 |
| 4.1 | <i>Programmet - hvordan fungere det</i> | 7 |
| 4.1.1 | Begrænsninger | 7 |
| 4.1.2 | Fejlmeddelsesr | 7 |
| 4.2 | <i>Eksempel</i> | 7 |
| 5 | Problemanalyse | 8 |
| 6 | Løsning | 9 |
| 6.1 | <i>Teknisk Analyse</i> | 9 |
| 6.1.1 | Kommentarsystem | 9 |
| 6.1.2 | Crowd-sourcet | 9 |
| 6.1.3 | Single Page Application | 9 |
| 6.1.4 | Backend | 9 |
| 6.1.5 | Programmeringssprog | 9 |
| 6.1.6 | White- og Blackbox testing | 9 |
| 6.1.7 | Application Programming Interface(API) | 9 |
| 6.2 | <i>Design</i> | 10 |
| 6.2.1 | Index side | 10 |
| 6.2.2 | Værkforum | 10 |
| 6.2.3 | Zoomfunktion | 10 |
| 6.3 | <i>Teknisk beskrivelse</i> | 10 |
| 6.3.1 | Flowchart | 10 |

| | | |
|-------|-------------------------|-----------|
| 6.3.2 | Data | 10 |
| 6.3.3 | Arkitektur | 10 |
| 6.3.4 | Brugergrænseflade | 10 |
| 6.3.5 | Begrænsninger | 10 |
| 7 | Afprøvning | 11 |
| 8 | Brugervejledning | 12 |
| 8.1 | <i>Eksempel</i> | 12 |
| 8.2 | <i>Brugsscenarie</i> | 12 |
| 9 | Konklusion | 13 |
| 10 | Litteratur | 14 |

Kapitel 1

Forord

Kapitel 2

Indledning

2.1 Baggrund

Det problemområde vi har arbejdet med i forbindelse med vores projekt, er udviklingen af et softwaresystem til brug for en ekspedient der betjener en billetluge i en biograf. Det område der skulle dækkes var reservation både når kunden stod foran billetlugen, samt når kunden ringede ind på telefon. Vores løsning skulle håndtere ekspedientens arbejdsopgaver i sådanne situationer. Løsningen skal derfor fokusere udelukkende på reservationer, og ikke salg.

2.2 Problemstilling

2.2.1 Krav til programmet

Af krav til programmet kan nævnes at der i forbindelse med reservationerne skal oplyses navn eller anden identitet på kunden. Vi har i vores program valgt at den centrale information omkring brugerne er dennes telefonnummer. Det har vi gjort af den grund at flere kunder godt kan have samme navn, men at ens telefonnummer er unikt.

Derudover skal biografen indeholde flere sale, og sende flere forskellige forestillinger i løber af dagen. Salenes størrelse, antal pladser mm, skal fremgå af databesen frem for at være indkodet i selve programteksten. Derfor har vi istædet for at skrive data ind i programteksten, valgt at lave en speciel klasse til at hente data fra databasen.

Det samme er gældende for de enkelte forestillinger, der derfor også bliver gemt i databasen.

2.2.2 Brugerscenario

Når ekspedienten åbner vores biobooking-system vil der på venstre side af vinduet vare opelistet de film der spilledes i biografen i øjeblikket. Klikker ekspedienten på en af disse film vil der på højre side af vinduet vise sig de tilhørende spilletider til den

valgte film. Klikker ekspedienten derimod ikke på noget, vil der til stå følgene tekst *Klik på en film til venstre for at vise tidspunkter*. Øverst oppe i vinduet vil man se tre faner: *Forestillinger*, *Reservation* og *Ret reservation*. Man kan godt navigere mellem fanerne, men har man ikke først valgt film og tidspunkt vil *Reservation* være ubrugelige.

Efter at ekspedienten har valgt en film og en spilletid vil systemet automatisk skifte videre til den næste fane: *Reservation*. Her kan ekspedienten vælge fri mellem de ledige sæder. Dette gøres ved at ekspedienten enten klikker enkeltvis på de grønne sæder, eller ved at trække musen over de ønskede sæder. De valgte sæder vil nu blive blå, så ekspedienten kan se hvilke der er valgt. Er sæderne tilfredsstillende skal ekspedienten skive navn samt telefonnummer på kunden ind i et informationsfelt nedernst til højre på siden. Nederst på siden ses i venstre hjørne antallet af sæder i alt, samt det ledige antal sæder. Dette kan ekspedienten også se på det grafiske billede over biografsalen, da de allerede reserverede sæder er røde, samt umulige at vælge for reservation. Efter kundens information er indtastet og ekspedienten har klikket på *Fuldfør reservation* knappen kommer et pop-up vindue op, hvor der står skrevet: *Bestillingen er gennemført*.

Ønsker den pågældende kunde derimod at ændre eller slette en reservation, skal ekspedienten klikke sig ind på den sidste fane: *Ret reservation*. På denne side finder ekspedienten øverst oppe et indput felt hvor kundens telefonnummer skal indtastes. Når ekspedienten har klikket enter vil kundens reservation(er) dukke op nedenfor i en pæn liste. Ekspedienten kan så klikke på den reservation som kunden ønsker ændret. Nu føres ekspedienten tilbage til *Reservation* fanen, men de sæder tilknyttet den pågældende reservation ses nu med blåt frem for rødt. Ekspedienten kan således tilføje eller fjerne sæder ved at klikke på sæder med musem og derefter klikke på *Ret reservation* knappen. Ønsker kunden helt at slette reservationen kan dette lade sig gøre ved at klikke på *Slet reservation* knappen.

2.2.3 Systemdesign

Kapitel 3

Problemanalyse

3.1 Vores løsning

3.1.1 Alternative løsninger

3.2 Databasedesign

Kapitel 4

Brugervejledning

4.1 Programmet - hvordan fungere det

4.1.1 Begrænsninger

4.1.2 Fejlmeddelsesr

4.2 Eksempel

Kapitel 5

Problemanalyse

Kapitel 6

Løsning

6.1 Teknisk Analyse

6.1.1 Kommentarsystem

6.1.2 Crowd-sourcet

6.1.3 Single Page Application

6.1.4 Backend

6.1.5 Programmeringssprog

-
-
-
-
-

6.1.6 White- og Blackbox testing

6.1.7 Application Programming Interface(API)

6.2 Design

6.2.1 Index side

6.2.2 Værkforum

6.2.3 Zoomfunktion

6.3 Teknisk beskrivelse

6.3.1 Flowchart

6.3.2 Data

-
-
- –
-

6.3.3 Arkitektur

-
-

6.3.4 Brugergrænseflade

Lyttere

Komponenter

6.3.5 Begrænsninger

Kapitel 7

Afprøvning

- 1.
- 2.
- 3.
- 4.

Kapitel 8

Brugervejledning

8.1 Eksempel

8.2 Brugsscenarie

Kapitel 9

Konklusion

Kapitel 10

Litteratur