

Digital Gæstebog

Mikkel Caira, Tobias Madsen, Mikkel Munck, Sigurd Suhm

15. november 2013

1	Indledning	2
2	Abstract	3
3	Analyse	4
3.1	Aktører	4
3.2	Use cases & scenarier	4
3.2.1	Scenarie for bruger	4
3.2.2	Scenarie for administrator	5
3.3	Objektmodel	5
3.4	Domænemodel	5
4	Design	7
4.1	Use case: Skriv besked	7
4.2	Use case: Skjul besked	12
4.3	Use case: Slet besked	13
4.4	Use case: Skift gæstebog	13
5	Arbejdsproces	15
6	Konklusion	17

KAPITEL 1

INDLEDNING

Jørgen Bjerring driver en mindre virksomhed i Magtenbølle lidt udenfor Odense. Han kombinerer *bed & breakfast* med baner til mountain biking samt kulturelle events for at skabe fokus omkring byen og tiltrække gæster. I opstarten af projektet søger han IT-løsninger til at hjælpe med driften og markedsføringen af konceptet.

Til at hjælpe med branding og markedsføring ønsker vi at udvikle en *digital gæstebog*, som tillader besøgende at give deres mening til kende og beskrive deres oplevelser. Bjerring vil på den måde få nem adgang til feedback fra de besøgende, og positive beskrivelser kan anvendes som anmeldelser på sociale netværk eller egen hjemmeside.

KAPITEL 2

ABSTRACT

This report will describe our work so far with the development of the *digital guestbook*. The main focus will be on the analysis of the problem domain as well as the design and implementation of the solution. The purpose of the report is to give the reader comprehensive insight in the system from a technical standpoint as well as a description of the design process that is the foundation of the solution.

Furthermore we will describe our work process and the innovative and collaborative methods that we have used.

KAPITEL 3

ANALYSE

3.1 Aktører

Der er følgende aktører i systemet:

Administrator: Han har mulighed for at se alle beskeder i gæstebogen, også selvom de er skjulte. Derudover har han mulighed for at administrere gæstebogens forskellige beskeder, ved at slette eller skjulte dem der ikke er relevante for ham og/eller brugerne.

Gæstebogs brugere: Brugeren har mulighed for at skrive en lille anmeldelse af deres ophold på gården, ved at indtaste navn, vurdering og en besked, der vil blive gemt og vist i gæstebogen for andre brugere, medmindre administratoren vælger at skjule/slette disse.

3.2 Use cases & scenarier

- *Brugeren* skriver en besked. (Han tilføjer sit navn, en vurdering og en besked)
- *Administratoren* sletter en besked fra gæstebogen.
- *Administratoren* skjuler en besked I gæstebogen.
- *Administratoren* vælger hvilken gæstebog der er aktuelt vist for brugeren.

3.2.1 Scenarie for bruger

Brugeren vil gerne skrive en anmeldelse af sit ophold. De åbner gæstebogs-applikationen, hvor de har mulighed for at indtaste deres navn, vurdering af opholdet, samt et besked.

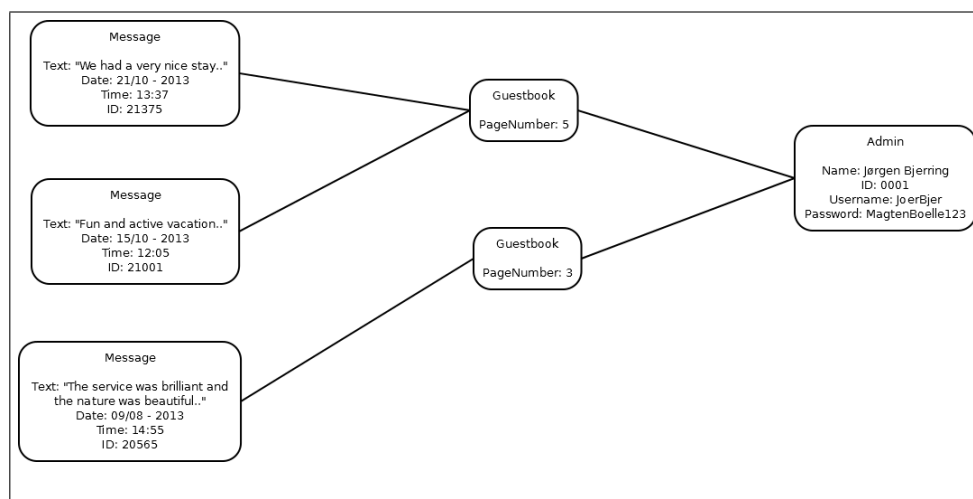
Herefter kan de sende beskeden som bliver oprettet og vist i en ny fane under *Se beskeder*. Her har de mulighed for at se deres besked, sammen med beskeder fra andre brugeres tidlige ophold.

3.2.2 Scenarie for administrator

Administratoren vælger at gå ind i administrator-panelet, hvor han har nogle forskellige muligheder. Han kan se beskederne i den aktuelt valgte gæstebog, han kan vælge beskeder ud som han vil skjule eller slette og han har derudover mulighed for at ændre hvilken gæstebog der er aktuelt vist for brugeren.

3.3 Objektmodel

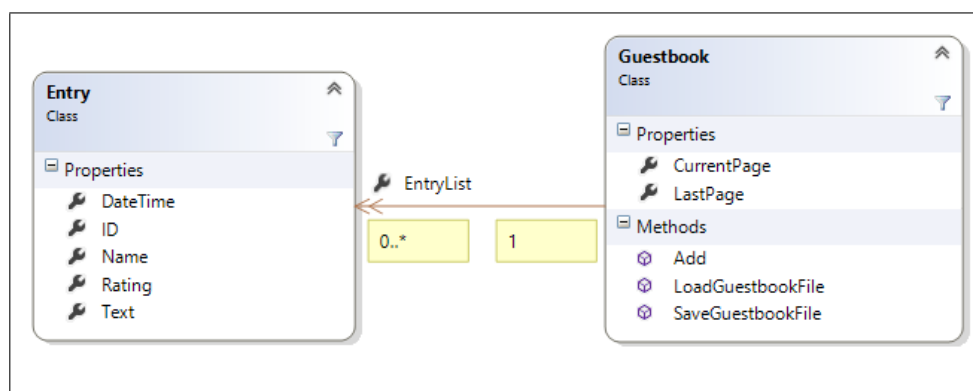
Figur 3.1 viser eksempler på objekter i systemet.



Figur 3.1: Objektmodel for den digitale gæstebog

3.4 Domænemodel

Figur 3.2 viser domænemodellen for systemet.

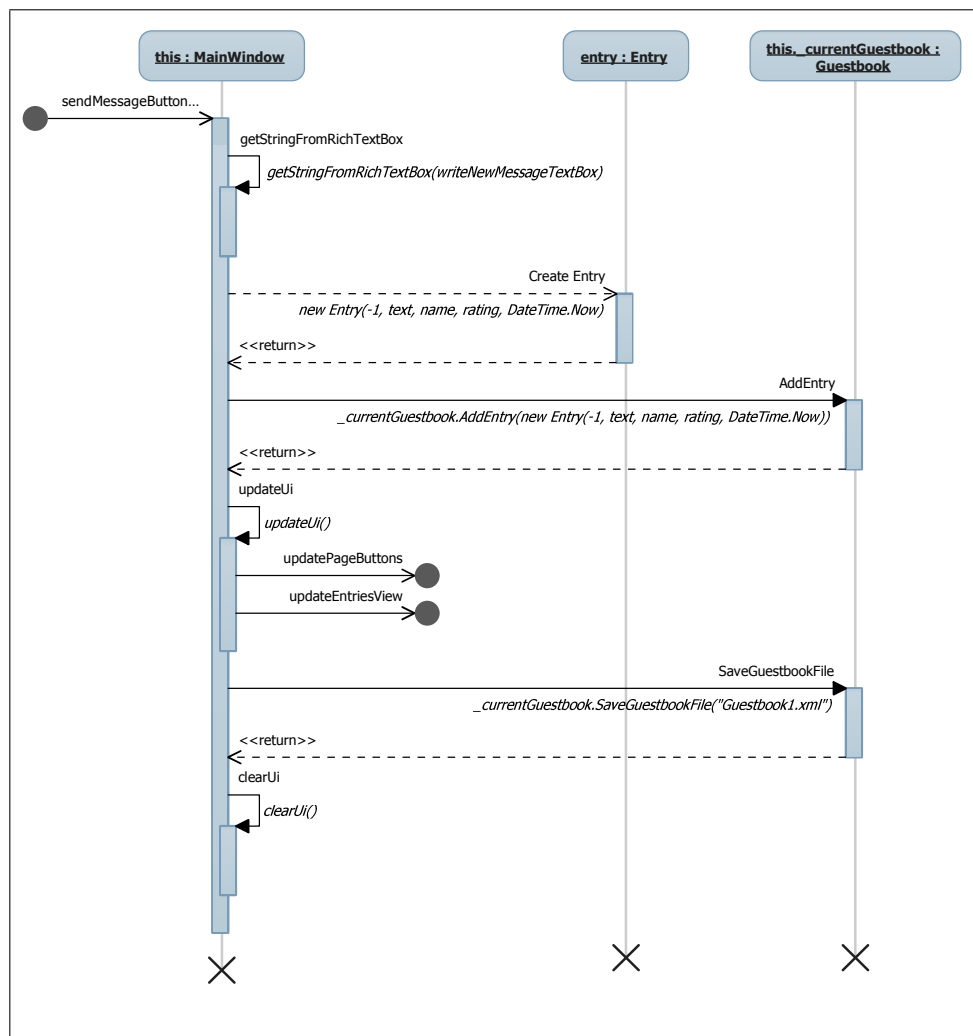


Figur 3.2: Domænemodel for den digitale gæstebog

KAPITEL 4 DESIGN

4.1 Use case: Skriv besked

Fokus har i dette sprint været på funktionaliteten, som tillader brugeren at skrive en besked i gæstebogen. Beskeden skal efterfølgende kunne vises i brugergrænsefladen og gemmes til en fil, som kan indlæses, når programmet åbnes næste gang.



Figur 4.1: Systemsekvensdiagram for AddEntry

Figur 4.1 viser systemsekvensdiagrammet for AddEntry-metoden.

De 3 centrale skridt i operationen, som er knyttet til domænelogikken, er:

- Opret Entry-objektet
- Tilføj Entry-objektet til gæstebogen
- Gem gæstebogsfilen

Det første skridt er et simpelt *constructor*-kald, og vi vil derfor ikke gå i dybden med det.

Det næste skridt er at tilføje beskeden til gæstebogen.

Use-case: Skriv en besked til gæstebogen

Operation: AddEntry(navn, besked: String, vurdering: Int, dato: Date)

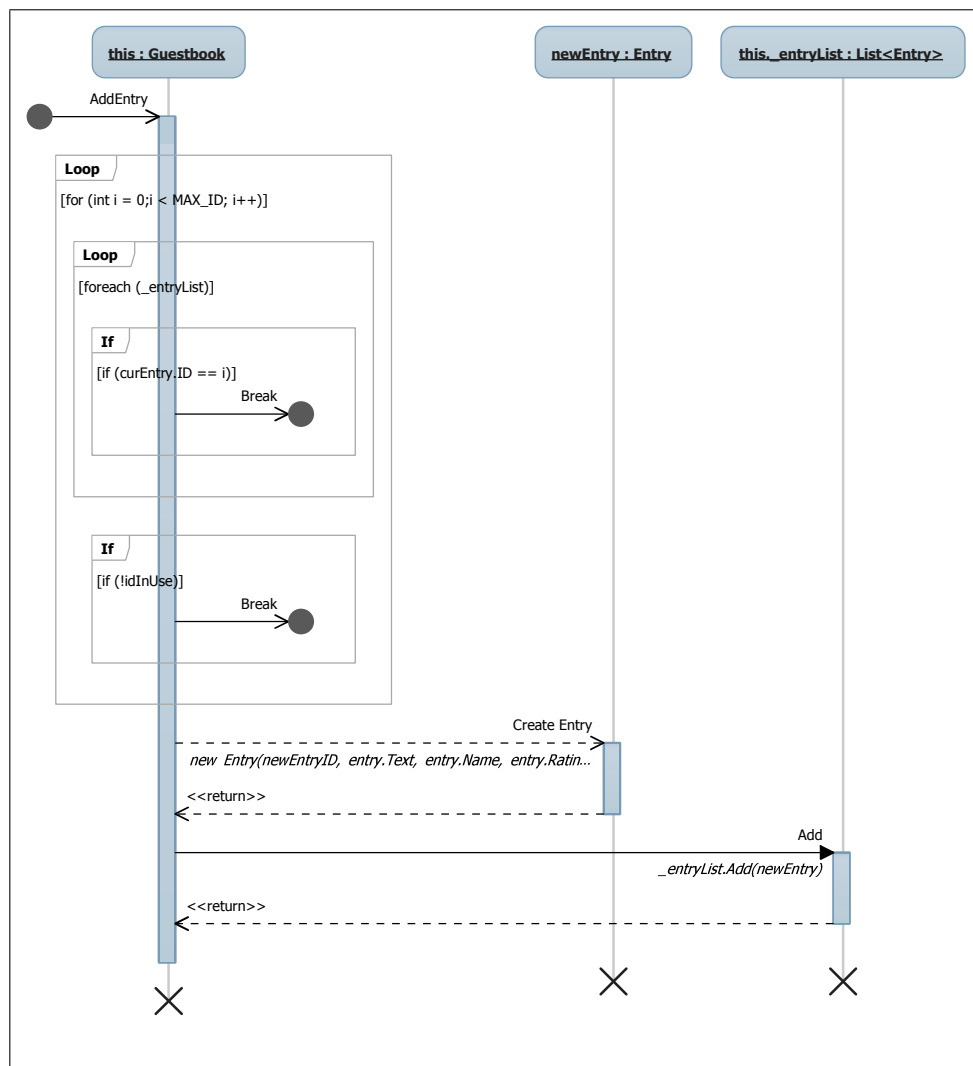
Pre: Navn, vurdering og besked er udfyldt

Post: Et unikt ID er genereret og beskeden er tilføjet til gæstebogen

Output: Brugeren gøres opmærksom på at beskeden er tilføjet

Figur 4.2: Operationskontrakt for AddEntry

Figur 4.2 viser operationskontrakten for denne metode.



Figur 4.3: Sekvensdiagram for AddEntry

Den centrale funktionalitet i denne metode er at generere et unikt ID til beskeden. Figur 4.3 viser sekvensdiagrammet for denne metode.

Use-case: Skriv en besked til gæstebogen

Operation: SaveGuestbookFile(filnavn : String)

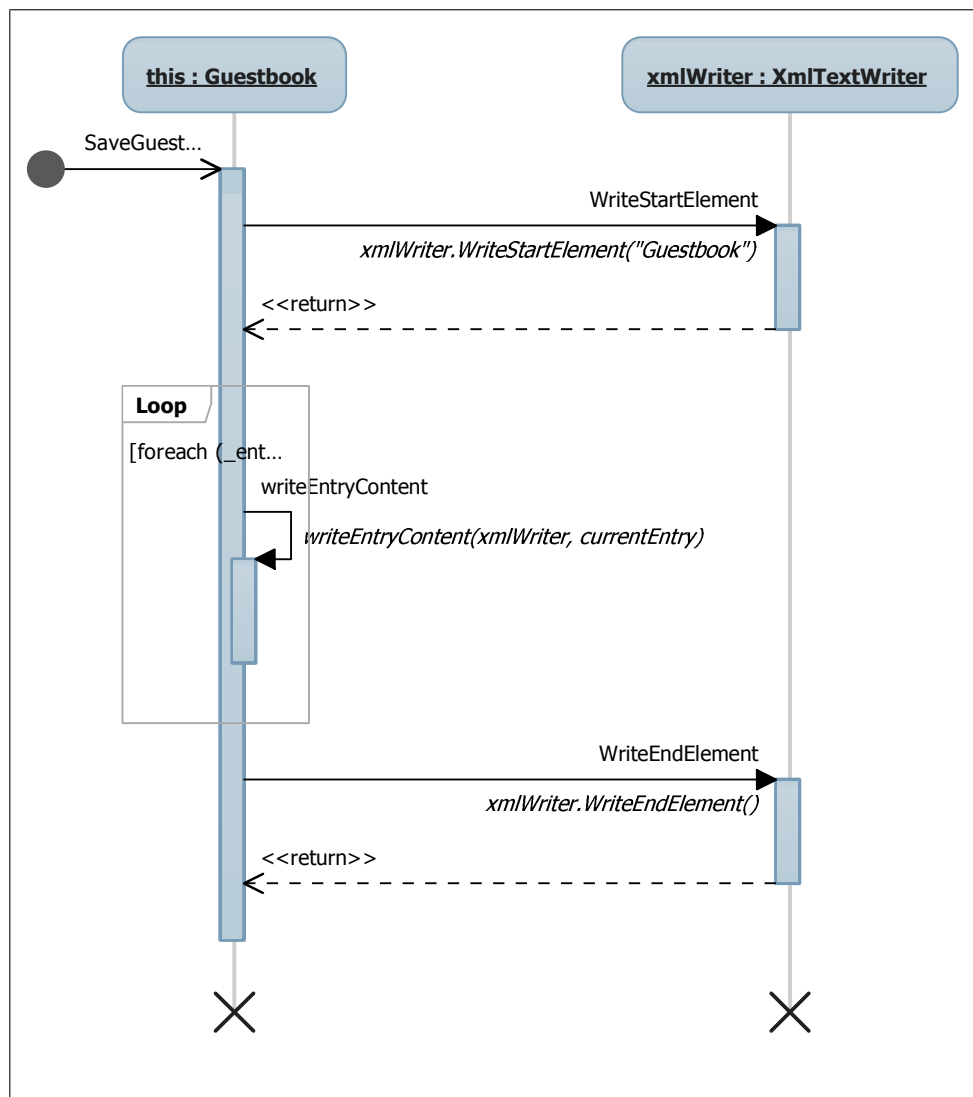
Pre: Et filnavn er valgt for den nuværende gæstebog

Post: Gæstebogen er gemt i Xml-format til harddisken

Output: Intet

Figur 4.4: Operationskontrakt for SaveGuestbookFile

Det sidste skridt er at gemme gæstebogsfilen til harddisken. Figur 4.4 viser operationskontrakten for denne metode.



Figur 4.5: Sekvensdiagram for SaveGuestbookFile

Figur 4.5 viser sekvensdiagrammet for denne metode.

4.2 Use case: Skjul besked

Funktionaliteten for administration er endnu ikke implementeret i systemet, men vi har udarbejdet operationskontrakter for de centrale metoder. Figur 4.6 viser operationskontrakten for funktionaliteten som tillader at skjule en besked i gæstebogen.

Use-case: Administratoren skjuler en besked i gæstebogen

Operation: HideEntry(navn, besked: String, vurdering: Int, dato: Date)

Pre: Beskeden er valgt

Post: Beskeden er skjult

Output: Beskeden er nu blevet skjult

Figur 4.6: Operationskontrakt for HideEntry

4.3 Use case: Slet besked

Figur 4.7 viser operationskontrakten for funktionaliteten som tillader at slette en besked fra gæstebogen.

Use-case: Administrator sletter en besked fra gæstebogen

Operation: DeleteEntry(navn, besked: String, id, vurdering: Int, dato: Date)

Pre: Der er valgt en besked

Post: Den valgte besked bliver slettet fra gæstebogen

Output: "Beskeden er slettet"

Figur 4.7: Operationskontrakt for DeleteEntry

4.4 Use case: Skift gæstebog

Figur 4.8 viser operationskontrakten for funktionaliteten som tillader at ændre, hvilken gæstebog, som skal være aktiv.

Use-case: Administratoren vælger hvilken gæstebog der er aktuelt vist for brugeren

Operation: ChangeGuestbook(Gæstebog: Guestbook)

Pre: Den aktuelle gæstebog er valgt

Post: Gæstenbogen er blevet valgt

Output: Du har nu valgt en ny gæstebog

Figur 4.8: Operationskontrakt for ChangeGuestbook

KAPITEL 5

ARBEJDSPROCES

I dette projekt har vi anvendt et online værktøj: Scrumwise¹, til at holde styr på vores arbejdsproces. Vi har mulighed for at holde styr på alle arbejdsopgaver løbende vha. værktøjer i Scrumwise.

Vi har lavet en backlog til vores projekt, som indeholder alle de overordnede opgaver der skal laves over hele projektforsløbet. Disse opgaver er derefter delt ud på nogle forskellige sprints, hvor de brugere der er tilmeldt det givne projekt har mulighed for at "tage en opgave", så man undgår at flere personer arbejder på det samme modul. Det har været en kæmpe fordel at anvende værktøjer som dette, da det giver os et overblik over hele projektet, sprints og konkrete kode-moduler.

I de fire første sprints har vi afsluttet tre backlog-opgaver, som det ses nedenfor på figuren. Det er herefter muligt at tilføje en eller flere backlog-opgaver til fremtidige sprints, hvor det er planen at afslutte de tre sidste opgaver over den sidste tid der er til rådighed inden den endelige applikation skal stå færdig.

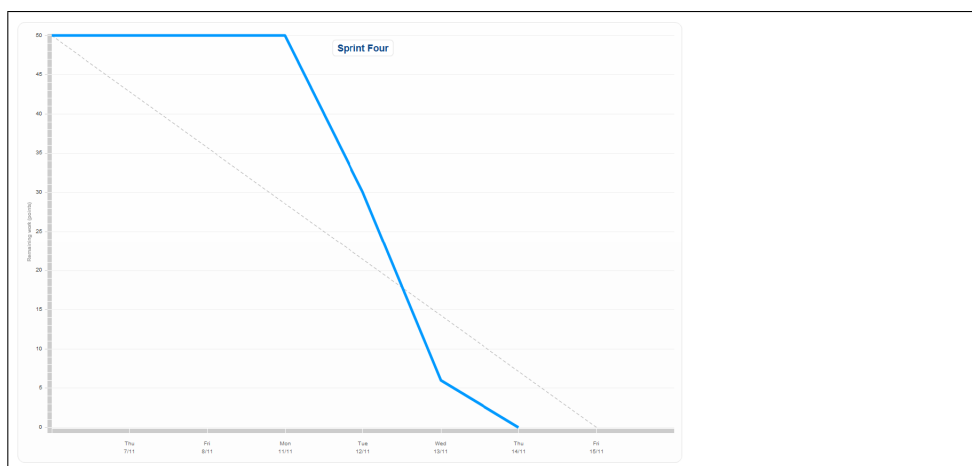
¹www.scrumwise.com



Figur 5.1: Backlog for projektet

Sprint fire gik ud på at optimere funktionaliteten af vores forhenværende applikation. Heriblandt er det nu muligt at tilføje en vurdering i form af stjerner, på en skala fra 1-5, som bliver tilknyttet en gæstebogsbesked. Derudover er fanebladet hvor man kan se gæstebogsbeskeder blevet optimeret i form af en mere dynamisk løsning. Der bliver nu tilføjet/fjernet sider i gæstebogen alt efter hvor mange beskeder der er tilknyttet hver side. Hvis der er mere end fem beskeder vil der automatisk blive oprettet en ny side, og en knap der gør det muligt at gå videre til næste side blive aktiveret, samt en tilbage-knap hvis man er på en anden side end den første. Til sidst er der blevet ændret nogle små-ting i koden og UI'et, så det hele er optimeret.

Arbejdsmorale har i dette sprint været eksponentielt, grundet lidt svingende fravær og helbred. Som det fremgår af burndown-figuren tog det os lang tid at planlægge hvilke arbejdsopgaver der skulle konstrueres i dette sprint. Dog var det muligt for os at nå at lave alle opgaver færdig til deadline fredag, da vi tog en tårn midt på ugen.



Figur 5.2: Burndown for sprint fire

KAPITEL 6

KONKLUSION

Den digitale gæstebog vil være i stand til at fremføre et billede af hvordan de besøgende opfatter Magtenbølle Enggård. Det er en It-løsning hvor der er fokus på de besøgendes syn på Magtenbølle Enggård. Jørgen vil være i stand til at kunne se de enkelte besøgendes opslag i gæstebogen og vil kunne uddrage konstruktiv kritik, samt gode oplevelser. Dette vil hjælpe ham til at gøre Magtenbølle Enggård mere attraktiv, i det han kan sprede budskabet på en nemmere måde, via eventuelle sociale medier.