

TRACKABUS

BACHELORPROJEKT

Procesrapport
for
TrackABus

Author:

Gruppe 13038

Supervisor:

Michael Alrøe

18. december 2013

Versionshistorie:

Ver.	Dato	Initialer	Beskrivelse
1.0	12-12-2013	13038	Start på rapportskrivning

Godkendelsesformular:

Forfatter(e):	Christoffer Lousdahl Werge (CLW) Lasse Lindsted Sørensen (LLS)
Godkendes af:	Michael Alrøe.
Projektnr.:	Bachelorprojekt 13038.
Filnavn:	Procesrapport.pdf
Antal sider:	26
Kunde:	Michael Alrøe (MA).

Sted og dato: _____

10832 _____
Christoffer Lousdahl Werge

MA _____
Michael Alrøe

09421 _____
Lasse Lindsted Sørensen

1 Resumé og abstract

1.1 Resumé

I forbindelse med bachelorprojektet, er der blevet udarbejdet et system til at indlæse positionen for en bus, vise denne på et kort, samt underrette brugeren om, hvor lang tid der er til ankomst for en bus ved et givent stoppested. Desuden er der også mulighed for, at ruter kan favoriseres, og hermed gemmes lokalt. Ud over brugerfunktionaliteterne, understøtter systemet også muligheden for, at en administrator kan oprette, vedligeholde og slette busser, busruter og stoppesteder.

Persisteringen af data sker i form af to relationelle databaser; En lokal og en distribueret. Den lokale giver brugeren mulighed for at gemme ruter der bruges ofte, og den distribuerede indeholder alle informationer om ruterne, samt bussernes nuværende og tidligere position. Den distribuerede database indeholder også funktionalitet til at udregne tiden for en bus til et givet stoppested.

Den distribuerede database er opbygget ved hjælp af MySQL, og den lokale er opbygget ved hjælp af SQLite. Brugeren tilgår systemet igennem en Android platform, hvorigennem samtlige brugerfunktionaliteter kan tilgås. Kodesproget brugt til dette er Java. Administrator hjemmesiden er opbygget ved hjælp af HTML, CSS og JavaScript, men da den er bygget på baggrund af ASP.NET er C# (C Sharp) det primære kodesprog i denne sammenhæng. Til systemet er der blevet designet en simulator, som står for at simulere en bus, i alle henseende. Den er blevet udarbejdet i WPF ved brug af .NET frameworket, med C# (C Sharp) som kodesprog.

Til projektstyring er der blevet brugt dele af Scrum, Extreme Programming og Unified Process.

Det færdige produkt er et system, hvori en bruger kan holdes opdateret omkring placering af busser, samt en administrator nemt kan oprette, ændre og fjerne ruter, stoppesteder og busser.

1.2 Abstract

In relation to the bachelor project, a system has been designed, to retrieve the position of a bus, draw it on a map, and notify the user of how much time remains, until the nearest bus, is at the chosen stop. The user system also implements the functionality, where a route can be favoured, and saved locally. Furthermore, the system also supports the possibility, that an administrator can create, maintain and delete buses, routes and stops.

The persistence of data, is done by two relational databases; One local and one distributed. The local database makes it possible for the user to save a route which is used often and the distributed database contains all the information regarding the routes, as well as the previous and current positions of the buses. The distributed database also contains the functionality to calculate the remaining time until a bus reaches a specified stop.

The distributed database is created with MySQL and the local is created with SQLite. The user accesses the system through an Android platform, where all user functionalities can be accessed through. The language used for this is Java. The administrator homepage, is created with HTML, CSS and JavaScript, but since it is created as an ASP.NET application, C# (C Sharp) is the primary language used. A simulator has been designed for the system, with the purpose of simulating the complete functionality of a bus. This has been created as a WPF application, with the .NET framework, and with C# (C Sharp) as a language.

For the purpose of project management, parts of Scrum, Extreme Programming and Unified Process have been used.

The final product is a system, where the user can be kept updated in regard to the the position of a bus, and where an administrator easily can create, manage and remove routes, stops and buses.

Indhold

1	Resumé og abstract	2
1.1	Resumé	2
1.2	Abstract	3
2	Forord	6
3	Indledning	6
4	Projektafgrænsning	9
5	Projektgennemførsel	9
5.1	Sprint 1	9
5.2	Sprint 2	10
5.3	Sprint 3	10
5.4	Periode med andet arbejde	10
5.5	Sprint 4	11
5.6	Sprint 5	11
5.7	Sprint 6	12
5.8	Sprint 7	12
5.9	Sprint 8	12
6	Metoder	13
7	Specifikations- og analysearbejdet	14
8	Designprocessen	17
9	Udviklingsværktøjer	19
10	Resultater	20
11	Diskussion af opnåede resultater	22
12	Projektets fortræffeligheder	23

13 Forslag til forbedringer af projektet eller produktet	24
14 Konklusion	25
15 Referencer	26

2 Forord

Dette projekt er udarbejdet af to studerende på Ingeniørhøjskolen i Aarhus. Projektet udgør, i sammenhæng med dette dokument og dokumenterne TrackABus_Kravspecifikation, TrackABus_Accepttestrapport og TrackABus_Systemarkitektur, bachelorprojektet på syvende semester for IKT-Linjen. Dokumentopsætningen er udgjort på baggrund af den projektform, der er blevet arbejdet med i første til fjerde semester.

Bachelorprojektet er lavet uafhængtigt af et firma, og er derfor udarbejdet på egen hånd, af projektgruppens medlemmer.

3 Indledning

En bus følger en ruteplan, men det er ikke altid, at bussen er ved et givent stoppested, præcis på det tidspunkt det forekommer i ruteplanen. Det vil derfor være gavnligt at kunne vide, præcis hvor en bus er, og hvor lang tid der er, til den nærmeste er ved et givet stoppested. Denne viden vil, for det første, give brugeren en større chance for at nå sin bus og for det andet, med sikkerhed vide, om en bus er kørt fra et givet stoppested. Dette dokument beskriver udviklingen af en mobil applikation der kan vise rute, stoppesteder og busser på en rute, samt vise tiden til ankomst for en bus til et stoppested. Desuden beskriver dokumentet også oprettelsen af et administrator værktøj, hvori busser, ruter og stoppesteder kan vedligeholdes. Mobilapplikationen er designet til Android, men kan nemt genskabes på en anden platform, da en stor del af funktionaliteterne ligger på en server. Serveren er dog ikke klargjort til et distribueret system, men kan nemt skiftes ud med en der er, hvis det skulle være nødvendigt.

Kravene er blevet udarbejdet agilt, da projektet ikke har en ekstern kunde og således ikke repræsenterer kundens krav til systemet, men derimod udviklernes. En kunde er blevet simuleret, i form af en vejleder, således at alle krav-ændringer og accepttest udførsel, blev gemmemgået med hjælp fra en ekstern kilde.

TrackABus mobil applikationen og administrator hjemmesiden, er blevet udviklet til at håndtere systemkravene. Systemet fungerer således, at en bruger kan vælge en rute, sådan

samtligte persisterede informationer omkring denne vil blive vist. Dette inkluderer busser, stoppesteder og selvfølgelig ruten. Brugeren kan herefter tilgå tidsudrenings funktionaliteten ved at trykke på et stoppested, hvorefter tiden til ankomst for den nærmeste bus i begge retninger, vil blive vist. De ruter brugeren kan tilgå, skal først oprettes igennem administrations hjemmesiden. Denne del af systemet er derfor den eneste, der kan ændre ruter og stoppesteder. I sammenhæng med persitering bruges der to relationelle databaser; En distribueret, og en lokal til hver mobilapplikation. Den distribuerede håndterer samtlige informationer om ruter, stoppesteder og busser, hvor den lokale bruges i sammenhæng med favorisering af ruter. Mobilapplikationen er baseret på et eksamensprojekt i ITSMAP, lavet af gruppens to medlemmer, og betegnes derfor som legacy code. Hele systemet kan derfor ses som en videreudvikling af dette eksamensprojekt. Der eksisterer ingen yderligere krav, end dem projektgruppen selv har fastsat.

Det har ikke været muligt at tilgå reelle data for busser og ruter. Derfor har det været nødvendigt at bruge en del af arbejdsressourcerne på, at designe og implementere et system til at kunne håndtere disse data. Dette blev i sidste ende, til administrator hjemmesiden og simulatoren.

I arbejdsproces øjemed, er der blevet holdt daglige møder i gruppen, hvor dagens arbejde blev diskuteret, samt det ugentlige mål revideret. Desuden blev der holdt ugentlige møder med ad-hoc kunden, hvori projektes fremskreden blev forklaret og diskuteret.

Begreber og forkortelser

- Scrum: Agil projektstyrings metode.
- UP: Unified Process.
- XP: Extreme Programming.

Læsevejledning

Nedenfor er listet en kort beskrivelse af hvert afsnit i dette dokument:

Afsnit 1 - Abstract og Resumé: Disse afsnit giver en kort beskrivelse af projektet på både dansk og engelsk.

Afsnit 2 - Indledning: Dette afsnit beskriver baggrunden for projektet, kravene til projektet, samt hvilke arbejdsmetoder og processer, der er anvendt. Herefter følger begreber og forkortelser samt en læsevejledning.

Afsnit 3 - Projektafgrænsning: Afsnittet beskriver kort om de begrænsninger, der blev sat, da projektet blev fastlagt.

Afsnit 4 - Projektgennemførelsen: Her præsenteres de forskellige iterationer, hvorigennem projektet er udviklet, samt erfaringerne med disse.

Afsnit 5 - Metoder: Dette afsnit beskriver de forskellige arbejdsmetoder, der er anvendt i udarbejdelsen af det endelige produkt.

Afsnit 6 - Specifikation- og analysearbejdet: Analysearbejdet, der ligger til grund for opbygningen af projektet, præsenteres i dette afsnit. Dette indebærer, hvordan der er kommet fra en idé til et færdigt system.

Afsnit 7 - Designprocessen: Her beskrives selve designprocessen, hvori beslutninger vedrørende design forklares.

Afsnit 8 - Udviklingsværktøjer: Dette afsnit giver en beskrivelse af de mest væsentlige udviklingsværktøjer, der er blevet brugt til udviklingen af projektet.

Afsnit 9 - Resultater: De mest væsentlige resultater præsenteres objektivt i dette afsnit.

Afsnit 10 - Diskussion af opnåede resultater: I dette afsnit diskuteres der på baggrund af de opnåede resultater.

Afsnit 11 - Projektets fortræffeligheder: Her præsenteres de dele af projektet, som er blevet fundet særdeles velfungerende.

Afsnit 12 - Forslag til forbedringer af projektet eller produktet: Afsnittet giver en beskrivelse af, hvad der kunne have været forbedret i selve produktet eller produktudviklingen.

Afsnit 13 - Konklusion: Der laves her en konklusion på baggrund af de opnåede resultater og diskussionen af disse, samt på baggrund af de opnåede erfaringer.

Afsnit 14 - Referencer: Afsnittet lister de forskellige materialer og værker, der refereres til.

4 Projektafgrænsning

Da projektet har været udviklet selvstændigt, uden et projektoplæg eller fastsatte krav fra en kunde, har der ikke været mange afgrænsninger. En afgrænsning der hurtig opstod, var muligheden for at få GPS-koordinator for busserne, fra Midttrafik. Dette gjorde det endnu vigtigere, hurtigt at få lavet en bus simulator, der kunne bruges som alternativ.

5 Projektgennemførelse

Projektet er blevet gennemført gennem ni iterationer med en varighed på 1-2 uger, der har hjulpet med til at dele projektet op i mindre, overskuelige, sprints.

Der har, i hver sprint, været fokus på at få implementeret en lille del af systemet eller få skrevet en del af dokumentationen. Ved at holde det til små sprint, har det haft den fordel, at det har været muligt, at basere de næste sprint på baggrund af opnået erfaring, og feedback fra alle de foregående sprints. Samtidigt har det hjulpet til at sikre projektet færdigørelse til tiden.

På figur 1 ses en graf der viser, antal commits til GitHub. Dette kan bruges som en grov skitse til at vise, hvornår i projektet forløbet det meste af arbejdet har fundet sted. Hertil kan det også ses, hvordan eksamensperioden midlertidigt afbrød arbejdsprocessen.



Figur 1: Antal commits over projektets forløb

5.1 Sprint 1

Sprint 1 fungerede som en typisk inception fase, som kendt fra udviklingsmetoden Unified Process. Sprintet fokuserede på at få udformet et forprojekt. Dette indebærer at finde ud af,

hvad hele projektet skulle bestå af og hvad der skulle laves. Til dette blev der opstillet og beskrevet de forskellige krav til systemet i form af en simple kravspecifikation, der kort beskrev de forskellige Use Cases og aktører i systemet samt et simple Use Case diagram blev lavet.

5.2 Sprint 2

Sprint 2, den første elaboration fase for projektet, løb fra uge 36 til og med uge 38. Dette sprint blev hovedsageligt brugt på at udforme detaljerede Use Cases.

Undervejs blev det opdaget, at der manglede en Use Case, der tog højde for et scenario, der blev overset i det første sprint. For at have et værktøj til at bruge som repository og versions styring blev GitHub sat op, og de første LaTeX filer blev tilføjet. Til at hoste hjemmesiden og databasen, blev domænet www.trackabus.dk købt, og en server blev lejet hos UnoEuro¹.

Efter første udkast til kravspecifikationen var lavet, blev den revideret og omskrevet.

5.3 Sprint 3

Sprint 3, der fungerede som den anden elaboration fase, foregik i uge 39. Fokuset i dette sprint lå på at få lavet første udkast til databasen, med de forskellige tabeller og relationer, og første udkast til hjemmesiden blev lavet, hvor der blev undersøgt hvordan Google Maps skal implementeres. Mobil applikationen, fra et tidligere projekt, blev fundet frem og samlet for at være sikker på den stadigvæk kunne kompileres og tilføjet til GitHub. I denne fase blev det første udkast til databasen implementeret. Herunder blev de første udregninger for tiden til ankomst for en bus ved et stoppested, påbegyndt. Hjemmesidens designet blev etableret, samt det blev det undersøgt, hvordan kortet herpå skulle opsættes.

5.4 Periode med andet arbejde

Denne periode startede i uge 40 og forløb til og med uge 43. Denne periode blev ikke brugt til dette projekt, da gruppen var optaget af andet arbejde. Dette inkluderede

¹www.unoeuro.com

eksamensprojekter, såvel som forberedelse hertil.

5.5 Sprint 4

Sprint 4 bestod af uge 44 og 45 og var den første del af construction fasen.

I dette sprint blev der fokuseret meget på at få udviklet det første fuldt fungerende system. Første udkast til hjemmesiden blev lavet færdig, med mulighed for at oprette simple busruter, stoppesteder og busser, samt gemme disse på den distribuerede database. Herunder blev muligheden for at tilføje busser til en rute også implementeret. Simulatoren blev også designet og implementeret i dette sprint, der gav mulighed for at simulere busser, som kører på deres tilknyttede busrute. Web servicen blev herunder også implementeret og designet. Dette blev gjort, da dette sprint også bestod af en videreudvikling i mobil applikationens funktionaliter. Dette betød, at proceduren for tilgang til den distribuerede database blev opdateret, da en ny database blev taget i brug. Det første udkast til den Stored Procedure, som står for at udregne tiden til ankomst for en bus til et stoppestedet, også blev færdig udviklet.

I slutningen af sprintet var et det første udkast til systemet færdig implementeret. Dog var der visse funktionaliter, som ikke blev færdiggjort. Dette var dog efter planen, da disse funktionaliter var ment til at blive udviklet i sprint 5.

5.6 Sprint 5

Sprint 5, som forløb i ugerne 46 og 47, var anden del construction fasen.

Med et fungerende system færdiggjort i sprint 5, blev fokuset lagt på færdiggørelse af de sidste funktionaliter. Dette indebar oprettelse af komplekse på kryds af alle komponenterne. Dette krævede en vis mængde refaktorering. Det indebar at lave database designet, såvel som procedurene og funktionerne herpå, til at understøtte denne funktionalitet. Hjemmesidens ruteadministrerings værktøj, blev herunder også videreudviklet, så disse typer ruter kunne oprettes. Herudover blev en ny brugergrænseflade udviklet, og funktionerne hertil blev opdateret. Udvidelsen af simulatoren krævede ikke mange ændringer, da dennes implementering af bussens vending ved endestation først blev implementeret i dette sprint. Dog blev database tilgangen for denne lavet fuldstændigt om, da der var

foretaget væsentlige ændringer på den distribuerede database.

De fleste mobil applikations funktionaliteter blev også udviklet i dette sprint. Dette blev gjort da denne var baseret på legacy code, og de fleste implementeringer var udviklet på forhånd. Det var dog nødvendigt at refaktorere store dele af denne, da det ikke levede op til de fastsatte design- og kodestandarder.

5.7 Sprint 6

Sprint 6 var første del af transition fasen for projektet og forgik i uge 48.

I dette sprint blev store dele af implementeringen refaktoreret, og acceptesten blev udformet og lavet færdig.

Under den interne afvikling af acceptesten, blev der fundet en del fejl. Disse blev dog hurtigt udrettet, hvorefter acceptestens afvikling blev fuldført sammen med ad-hoc kunden. Nogle definerings- og stavfejl blev fundet, og disse udbedret. Herefter blev testen godkendt, hvorefter arbejdet på dokumentet vedrørende systemarkitekturen, blev påbegyndt.

5.8 Sprint 7

Sprint 7, anden del af transition fasen forløb over to uger; Uge 49 og 50.

Hele dette sprint blev brugt på at færdiggøre dokumentet vedrørende systemarkitekturen, samt dette dokument. Samtidig blev fundet nogle fejl i systemet, som blev udrettet. Herunder blev der også tilføjet enkelte fejlbeskeder til mobil applikationen og hjemmesiden, vedrørende tab af internet forbindelse.

5.9 Sprint 8

Sprint 8, det sidste sprint, samt den sidste del af transition fasen forgik i uge 51, frem til og med onsdag.

Da projektet skulle afleveres i slutningen af sprintet, blev den resterende tid brugt til at gennemrette dokumenterne, samt gøre hele projektet klar til aflevering. Dette indbar udskrivning, indbinding samt oprettelse af bilag. Den sidste dag af sprintet bestod udelukkende af aflevering.

6 Metoder

Som nævnt i indledningen, er der blevet arbejdet under det agile process framework *Scrum*. Metoden er dog modificeret en smule, da den er designet til en større projektgruppe. De daglige Scrum møder bruges i denne sammenhæng, som en mulighed for at diskutere dagens opgaver. Da gruppemedlemmerne konstant har arbejdet i samme lokale, har præsenteringer af den foregående dags arbejds, ikke været essentiel under Scrum mødet.

Produktet er opbygget via iterationer, hvori der først og fremmest blev fokuseret på funktionalitet, dog ikke uden at sørge for, at de designmæssige krav blev opfyldt. Hver iteration har haft et mål, som var nødvendigt at nå, før næste iteration kunne påbegyndes. Grunden til dette er, at arbejdsformen har været meget lagdelt, idet en iteration byggede funktionalitet ovenpå den forrige iterations mål. Ved hver fuldstændt iteration blev et kort møde afviklet i gruppen, hvor der blev diskuteret, hvad det næste mål skulle være og hvorvidt ny information fundet under denne iteration, skulle implementeres i den næste eller, om det krævede en reimplementering af tidligere kode. Dette har medført en yderst agil arbejdsmetode, hvor resultater konstant blev revideret, bygget ovenpå eller fjernet fuldstændigt. Dette resulterede i et produkt, som kun indeholder implementering, der er blevet grundigt undersøgt.

Hver uge er der blevet afholdt et møde med ad-hoc kunden, hvor en produkt inkrementering blev præsenteret, hvis dette var reelt. Hvis ingen nævneværdige produktændringer var blevet foretaget, blev der givet beskrivelse af, hvad den forløbne uges arbejde indebar, samt hvilket arbejde der var planlagt for den kommende.

Under projektudførslen er der blevet gjort brug af Unified Process (UP), hvori specielt inception og elaboration faserne har været essentielle. Inception fasen fandt sted før projektet egentligt gik i gang. Dette er blevet udformet som et forberedelses projekt, hvori ideen blev grundlagt og et udkast til kravene og målene blev givet. Elaboration fasen foregik som udformningen af den første kravspecifikation, hvori projektets grænser blev optegnet. Construction fasen har udelukkende været implementering, samt opdatering opdateringer af kravene, hvis dette blev nødvendigt. I transition fasen blev systemarkitekturen, accept-testen og processrapporten udformet, samt der blev gjort klar til aflevering. UP har været

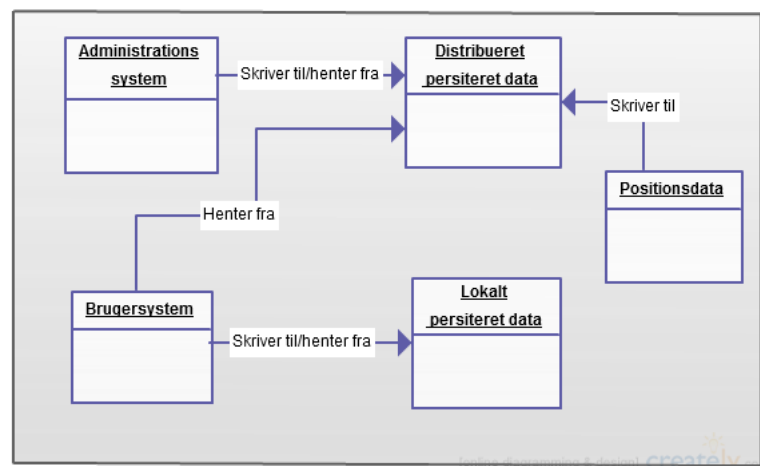
en ganske brugbar udviklingsprocess, da den supplerer Scrum med et iterativt overblik.

Implementeringenprocessen er udført ved hjælp af elementer fra Extreme Programming (XP). Planlægning og kommunikation har været arbejdsprocessens grundelementer, hvori der ofte blev planlagt mange uger frem, imens fokuset kun var på den givne opgave. Den ubrudte kommunikation i projektgruppen bidrog til denne arbejdsform, da det ene gruppemedlem til hver en tid vidste, hvad det andet gruppemedlem lavede. Hertil blev der ofte arbejdet med komponenter, der interagerer direkte, så der konstant blev holdt integrationstests. Udover acceptesten, har disse tests ikke været formelle og er udelukkende afviklet internt i gruppe.

Resultatet af denne arbejdsform er, at ansvarsområder hurtigt bliver uddelegeret, og vedligeholdt. Dette betyder også, at der var klare linjer for, hvem der skulle refaktorere hvilke komponenter, ved overgangen til en ny iteration. Gruppemedlemmet der havde ansvaret for en given komponent havde derfor det største overblik over denne. Dette betød dog ikke, at det andet gruppemedlem ikke vidste hvordan komponenten var udviklet. Dette overblik blev sikret af den konstante kommunikation.

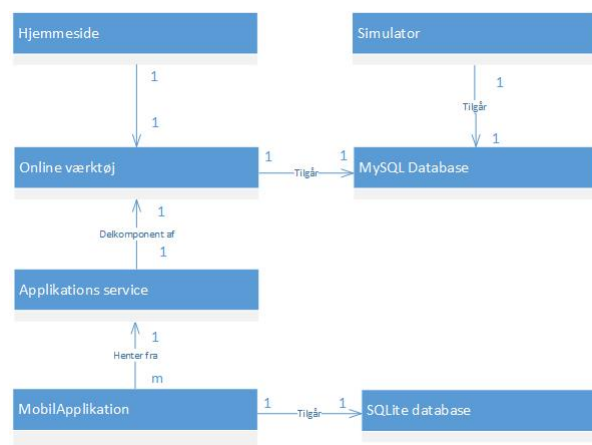
7 Specifikations- og analysearbejdet

Ved påbegyndelse af projektet blev der undersøgt, hvilke domæner projektet skulle bestå af. Dette var i særdeleshed vigtigt, da systemet består af flere dele, som udelukkende interagerer over en distribueret database. På figur 2, kan det første udkast til domænerne ses.



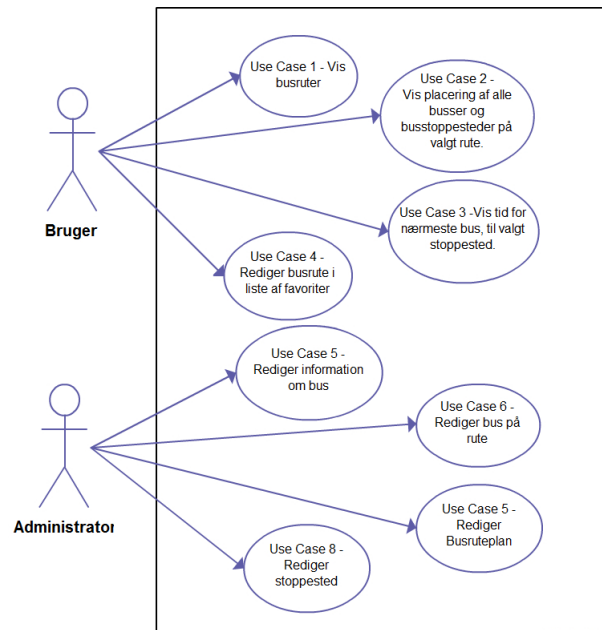
Figur 2: Første udkast af domænemodellen

Under udviklingen af systemet, blev der foretaget ændringer i domænemodellen. De fleste komponenter forblev de samme, men mobil applikationens tilgang til den distribuerede database, blev ændret til at være indirekte. Det vil sige den tilgår en web service, som igen tilgår databasen. Dette sikrer en lav kobling mellem databasen og applikationen, samt en høj samhørighed, da database tilgangen bliver samlet i en komponent, der ikke direkte er knyttet til applikationen. På figur 3 kan den færdige domænemodel ses.



Figur 3: Færdig domænemodel

Efter oprettelse af den første domænemodel, blev der udtænkt de brugssituationer, som et færdigt system skal kunne håndtere. Simulatoren blev ikke set som en komponent i denne sammenhæng, da dette blot var et værktøj til udviklingen. Det var udelukkende hjemmesiden og mobilapplikationen, der blev set som komponenter i en brugssituations sammenhæng. På figur 4, kan det Use Case diagram, systemet er bygget op omkring, ses.



Figur 4: Færdig domænemodellen

Da Use Case diagrammet udelukkende afspejler mobilapplikationen og hjemmesiden, blev det valgt, at to primære aktører ville være repræsentanter for hele systemet; Brugeren, som tilgår mobilapplikationen og Administratoren, som tilgår hjemmesiden. Databasen og simulatoren kunne ses som sekundære system aktører, men dette er dog blevet fravalgt, da det i en brugssituation vil være brugeren fuldstændigt irrelevant, hvorfra dataen kommer. Ud fra dette diagram blev der udviklet en kravspecifikation, som beskriver de funktionelle krav, systemet skal opfylde. Dette dokument er blevet set som agilt, da der ikke var nogen ekstern kunde, som kunne fastsætte krav og derved var det op til gruppen, hvordan systemet skulle udvikles. Dette har haft den betydning for systemet, at det under processen har været nødvendigt, at foretage visse ændringer i dokumentet, dog ikke uden, at det er blevet grundigt diskuteret i gruppen. Disse udviklinger blev samtidigt afspejlet i Use Case diagrammet, som blev ændret i sammenhæng med kravændringer. Dette har medført, at Use Case diagrammet og kravspecifikationen er blevet samlet til et dokument, der følger systemets udvikling.

Til kravene sat i kravspecifikationen, er der blevet udviklet en accepttestspecifikation, som beskriver, hvordan en eventuel kunde kan teste, at de stillede krav er opfyldt. Dette dokument har også været agilt, i og med det har ændret sig sammen med kravspecifikationen. Ved fuldent implementering af systemet, blev testene, specificeret i dette dokument, fulgt

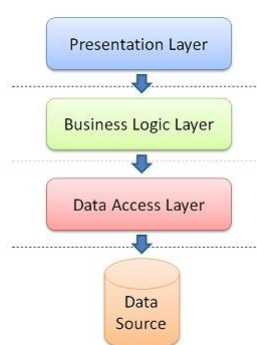
af gruppens ad-hoc kunde.

Under implementeringen blev det fundet meget brugbart at bruge simple sekvensdiagrammer, til at analysere kodens forløb. Disse har dog kun været midlertidige, da de er blevet tegnet på en tavle. Dette har givet det gruppemedlem, som ikke implementerede denne del af systemet, mulighed for at følge det andet gruppemedlems tankeforløb. Til de mere komplekse dele af systemet er der dog blevet udviklet mere detaljerede sekvensdiagrammer.²

Analyse- og specifikations arbejdet forløb efter hensigten, da den konstante kommunikation mellem gruppens medlemmer har sikret, at der var et bibeholdt overblik over processen. Dette sikrede desuden også, at funktionaliteter aldrig blev overset.

Opdelingen af systemet i domænemodellen, samt underopdelingen i Use Cases, gjorde det nemt for gruppens medlemmer at vælge en opgave og fuldføre den. I denne sammenhæng er der aldrig blevet gjort redundant arbejde, da grænsefladen mellem domænerne var klare. Da implementeringen dog blev udviklet iterativt, var det ofte nødvendigt at refaktorere store dele af koden. Dette ses dog ikke som redundans, da refaktoring er en naturlig følge af en agil arbejdsprocess.

8 Designprocessen



Figur 5: Tre-lags model

²Sekvensdiagrammer kan ses i kontekst i systemarkitektur dokumentet under 5: *Logisk view*, 6: *Process/Task view* og 8: *Implementerings view*

Da første udkast til kravspecifikationen var skrevet, blev der fastslået visse designregler, der skulle opretholdes igennem systemet. Den første var vedrørende systemopbygningen af de forskellige komponenter. Det blev hurtigt vedtaget, at for at skabe en god kode-standard og en overskuelige implementering, var tre-lags modellen den bedste metode. (se figur 5)

Tre-lags modellen er simpel og solid og kan nemt implementeres, hvis der arbejdes imod det. Grundidéen i tre-lags modellen er, at opdele systemet i ufahængige moduler, som ikke har behov for, at kommunikere komplekst imellem sig. Dette vil skabe høj samhørighed, samt lav afhængighed, hvilket er bestræbelser der forekommer i ethvert IT-system.³

Også i sammenhæng med høj samhørighed og lav afhængighed, blev det vedtaget, at mobil applikationen aldrig måtte kommunikere direkte med databasen, samt at alle udregninger skulle ske server-side. Det resulterede i en mobil applikation, som ikke udfører noget datamanipulations arbejde, men kun henter, gemmer og bruger data. Desuden kan mobil applikationen nemt skiftes ud, hvis den, for eksempel, skal supporteres på en anden platform. Dette betyder lav afhængighed mellem denne og de andre system komponenter. Af sikkerhedsmæssige årsager, blev det også vedtaget, at forbindelsesbeskrivelsen til databasen skulle gemmes væk.

I datamæssigt sammenhæng, blev det først undersøgt, hvorvidt det var muligt at tilgå reelle informationer om busser. Det blev dog hurtigt konstateret, at dette ikke var en mulighed. Dette var hovedårsagen til, at simulatoren blev udviklet. Simulatoren gav gruppen mulighed for at teste de dele af systemet, hvor informationer om en bus var en nødvendighed, uden en reel bus skulle tilgås. Dette gjorde altså projektet uafhængig af, hvilken kilde informationen kommer fra og derfor også med til at skabe lav kobling.

Rutemæssigt blev der gjort en del overvejelser. Projektet blev i første omgang specificeret til, at samtlige stoppesteder på en rute (dvs. stoppesteder i begge retninger), skulle

³En detaljeret model af systemerne kan følges i systemarkitekturen, under *5: Logisk View* og *8: Implementerings View*

kunne vælges, således at brugeren kunne tilgå den enkelte retnings stoppested. Dette blev hurtigt genovervejet, da det ville resultere i en uoverskuelig rute, med to gange så mange stoppesteder som der, reelt set, var behov for. Derfor blev det vedtaget, at der kun skulle vælges ét stoppested, og tiden for den nærmeste bus, i begge retninger, ville blive vist. I denne sammenhæng blev det også overvejet, at en rute sagtens kunne have mere end en endestation, og det var derfor nødvendigt også at tage højde for disse typer ruter.⁴

Under hele processen er designløsninger og -beslutninger blevet diskuteret imellem medlemmerne af gruppen, hvilket har resulteret i et yderst fleksibelt produkt.

9 Udviklingsværktøjer

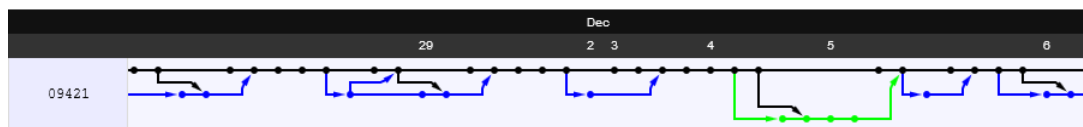
Under udførelsen af projektet er der blevet gjort brug af dele af Scrum, men da gruppens medlemsantal kun er to, blev det ikke set som nødvendigt at tage et Scrum værktøj i brug. Der har været konstant kommunikation mellem gruppemedlemmerne, da der udelukkende er blevet arbejdet i samme lokale. Under skriveprocessen af kravspecifikationen, accepttesten, system arkitekturen og dette dokument, er samtlige underafsnit blevet skrevet på et Scrum-board, hvorpå samtlige gruppemedlemmer kunne melde sig på en opgave.

Under udviklingen af systemet, blev der gjort brug af ansvarsområder, hvori der var klare linjer for, hvem der skulle udvikle hvilke dele.

Til versionsstyring og som repository er der blevet gjort brug af Git. For at tilgå dette er Git værktøjet GitHub blevet brugt. Dette er et standard repository system med versionstyring. Herigennem kan medlemmer tilføje, fjerne eller ændre filer, hvorefter andre medlemmer kan tilgå ændringerne. Når et system af denne størrelse implementeres, genereres der ofte en del unødvendige filer. Git implementerer muligheden for, at kunne ignorere disse filer igennem en ignore protocol. Dette har specielt været nyttigt under udviklingen af mobil applikationen, samt ved skrivningen af projektes dokumenter, da begge auto-genererer mange unødvendige filer, ved bygning. GitHub, og dermed repositoryet, tilgås igennem et commandline værktøj, kaldet Git Shell, som bygger ovenpå Windows

⁴Se afsnit 9: *Data View* i systemarkitekturen, for mere information om dette.

PowerShell. Heri kan der navigeres til den mappe, hvor repositoryet er oprettet og push, pull, add, commit, og merge funktionerne kan her eksekveres. Repositoryet kan også tilgås online, hvori versions historikken kan tilgås. Dette har været nyttigt i den sammenhæng, at et tidligere commit nemt kunne hentes ned, hvis to versioner skulle sammenlignes. Commit og merge træet kan også følges. På figur 6, kan en del af dette træ ses. Den sorte linje repræsenterer masteren, eller trunken. De blå og grønne linjer repræsenterer en lokal branch, hvori mere end et commit er blevet tilføjet. Når en linje peger på en anden betyder det, at et merge foregik, ved dette commit. Prikkerne på de forskellige linjer repræsenterer et commit.



Figur 6: Git commit historik

Alt tekstredigering er foregået i *LaTeX*, hvilket har været meget brugbart, da større dokumenter kunne opdeles i mindre filer, som kunne arbejdes på individuelt. Dette har tilladt, at begge gruppemedlemmer kunne redigere samme dokument uden der opstod konflikter. Desuden sikres det, at de nyeste billeder og lignende altid er opdateret, da *LaTeX* refererer til disse i stedet for at indsætte dem direkte.

De fleste diagrammer er blevet udviklet i Microsoft Visio 2013, da denne havde indbyggede skabeloner til alle de diagrammer, der blev set som nødvendige. Den eneste undtagelse er Use Case diagrammet og den første domæmemodel, som blev udviklet på hjemmesiden www.creately.com.

10 Resultater

Nedenfor er de væsentligste resultater beskrevet. Der refereres til accepttestspecifikationen, hvis det ønskes at få et større overblik over de udførte tests.

Overordnede resultater

- Få en præcis tid til ankomsten for en bus ved et stoppested.

- En måde at administrere busruter.
- Relationelle databaser til persistering af data.
- Web service til database adgang fra mobil applikationen.
- Simulering af busser.

Mobil applikation

Mobil applikationen kan vise busruter, med dennes stoppesteder og kørende busser. Det er samtidigt muligt at få vist tiden til ankomst for en bus ved et valgt stoppested. Ruter kan desuden favoriseres, således de gemmes lokalt.

Administrations værktøj

Det kan lade sig gøre, at administrere busruter. Dette indebærer at oprette, slette og ændre ruter, stoppesteder og busser, samt hvilken rute de forskellige busser er tilknyttet.

Databaser

Den distribuerede database er skabt med MySQL og består af otte tabeller. Disse indholder informationer vedørende busruter, stoppesteder samt busser og deres GPS-koordinater.

Den lokale database er skabt med SQLite og består af fem tabeller. Disse indeholder informationer om busruter og stoppesteder og er således et udsnit af den distribuerede database. Denne database oprettes sammen med mobil applikationen, på de telefoner den installeres på.

Web service

Web servicen fungerer som et mellemlid mellem mobil applikationen og MySQL databasen. Den står desuden også for at fjerne meget arbejde fra mobil applikationen og lade dette udføres på serveren.

Simulator

Det er muligt at simulere en eller flere busser der kører på deres tilknyttede busrute.

11 Diskussion af opnåede resultater

Som det kan ses i overstående afsnit, lykkes det at udvikle et software system, der kan vise en præcis tid til ankomst for en bus ved et valgt stoppested, hvilket er hoved funktionaliteten i hele systemet. Ydermere er det muligt nemt at oprette nye busruter, stoppesteder og busser og persistere dem på en database, så det nemt kan blive vist på mobil applikationen.

Selve mobil applikationen kører stabilt og selv ved et eventuelt tab af internet forbindelse vil dette blive håndteret og brugeren notificeret. Bussens placering og tiden til ankomst for denne ved et stoppested er begge meget præcise og ligger indefor et acceptabelt niveau. Brugergrænsefladen, der er blevet udviklet til mobil applikationen, er blevet gjort brugervenlig og overskuelig. Knapperne er tydelige og med en beskrivende tekst. Information bliver vist således, at der ikke kan skabes tvivl om, hvad der beskrives. Dataforbruget ved mobilt data netværk er også holdt ved et minimum da der hentes igennem en web service som håndterer komliceret database tilgang. Tidudregninger sker ligeledes på databasen så mobil applikationen ikke skal bebyrdes med dette. Dette resulterede i en mobil applikationen, som ikke udfører meget arbejde, hvilket også afspejles i størrelsen af applikationen på mobil telefonen.

Før det var muligt at få indtegnet ruter, stoppesteder og busser på mobil applikationen, skulle implementeres en metode til at oprette og gemme disse på den distribuerede databases. Til dette formål blev administrations hjemmesiden udviklet. Denne lader en administrator tilføje busser, busruter og stoppested til systemet på en simpel måde. Specielt ruteindtegnings værktøjet er meget brugervenligt, da denne lader administratoren oprette en rute ved få tryk. Brugergrænsefladen kunne dog forbedres og forsimples, da den i øjeblikket udelukkende afspejler funktionalitet. Dette er dog en relativ simpel process, da hjemmesiden er blevet opbygget ved brug af tre-lags modellen, i form af MVC. Der er ikke blevet implementeret administrator rettigheder på hjemmesiden, da det i testformål vil virke bebyrdelse, at skulle logge ind, ved hver tilgang. Dette skal dog implementeres i tilfælde af distribuering af systemet da det ikke skal være muligt for hvem som helst at tilgå hjemmesiden.

Den distribuerede database indeholder alt persisteret data systemet skal bruge. Samtidig

bliver den brugt i sammenhæng med tidsudregninger, som udføres direkte på databasen i form af Stored Procedures og Functions. Dette er blevet gjort, da den igennem udregningen skal bruge det data, der allerede er persisteret. Hvis dette skulle gøres i web servicen, ville det betyde flere kald til databasen, hvilket vil resultere i en lavere responstid.

Mobil applikationen gør også brug af lokal persistering, også i form af en database. Den bliver brugt til at gemme og hente ruter, som bruges ofte, da der således spares på dataforbruget og indlæsningen af ruter foregår hurtigere. Struktureringen af tabellerne er blot et udsnit af den distribuerede database, hvilket betyder, at data skal indskrives på samme måde, og læst data kan bruges direkte.

Web servicen er blevet lavet for at abstrahere arbejde væk fra telefonen, og over på en hurtigere server. Desuden er den blevet udviklet for at telefonen ikke tilgår databasen direkte, da dette ville skabe et stort sikkerhedsproblem.

Da meget arbejde foregår på serveren og mobil applikationen blot formidler data, og indtegner dette på et kort, skal der ikke meget arbejde til at udvikle applikationen til en ny platform.

Da det ikke var muligt at få adgang til GPS positionen for fysiske busser, blev der udviklet en simulator til dette formål. Denne er yderst brugervenlig, og kræver ikke mange interaktioner for at starte en simulering. Når simuleringen er startet kræves der ikke længere bruger interaktion, før den skal stoppes igen.

Opgbygningen af denne simulator gjorde det særdeles nemmere at teste på busser, frem for at være afhængig af reelle bus positioner.

12 Projektets fortræffeligheder

Nedenfor er en række funktionaliteter og arkitekturer i projektet, som gruppen er specielt stolt af, beskrevet.

Mobil applikation

Da mobil applikationen var den vigtigste del af projektet, blev der sat stort fokus på, at

få bygget denne med et godt design. Der blev også sat fokus på brugervenlighed samt intuitivitet. Dette blev opnået ved, at brugeren skal fortage så få interaktioner med mobil applikation som muligt, for at blive vist ønsket information. Samtidig er det tydeligt, hvad de forskellige knapper bruges til.

Web service

Der blev hurtigt fastslået, at der skulle ske så lidt arbejde i mobil applikation som muligt. Dette blev løst ved at abstrahere tilgangen til den distribuerede database og alle dennes funktioner, væk fra mobil applikationen og ind på web service. Dette har gjort det muligt at udføre meget dataprocessering på en server, i stedet for en telefonen. Desuden muliggøres det, at nemmere at kunne udvikle lignende applikationer til andre mobile styresystemet så som iOS og Windows Phone.

13 Forslag til forbedringer af projektet eller produktet

Under udarbejdelsen af et softwareprojekt er altid plads til forbedringer, både mht. til arbejdsprocessen og selve produktet. Det gælder også for dette projekt. Nedenfor er en række elementer i projektet, som kunne have været optimeret.

GUI - Administrations hjemmeside

Brugergrænsefladen på administrations hjemmesiden kunne være bedre. Hvis vinduet på hjemmesiden mindskes vil de forskellige elementer ikke længere passe i deres respektive felter. Detet kan nemt blive løst ved at lave et nyt .css stylesheet, der vil gøre hjemmesiden til en mere behagelig brugeroplevelse. Der er dog blevet fokuseret mere på funktionalitet frem for smuksering i dette projekt, derfor er der ikke blevet sat tid af til dette.

Server

Serveren der hoster både administrations hjemmesiden, web servicen og MySQL databasen, er en billig, men ikke særlig kraftig, server og kan derfor sandsynligvis ikke håndtere et stort antal samtidige brugere af mobil applikationen. Dette er dog ikke nødvendigt i

udviklings øjemed, men i tilfælde af at systemet skal distribueres, kan hele funktionaliteten på serveren nemt flyttes til en anden.

Administrator login

Administrator rettigheder skal kunne godkendes, før hjemmesiden kan tilgås. Dette er ikke blevet implementeret, da det blev set som en bebyrdelse i udviklingen af systemet, da dette ville betyde et login, ved hver test. I tilfælde af at systemet skal distribueres, skal denne funktionalitet dog implementeres.

14 Konklusion

TrackABus systemet er produktet af en idé, der blev realiseret, igennem velfungerende arbejdsprocesser og metoder. Dette bliver tydeligt afspejlet i det endelige produkt.

Ved projektstart blev der brugt meget tid, på at udforme krav, samt at diskutere designvalg. Dette resulterede i, at der var en klar, rød tråd igennem hele projekt, som også sørgende for at gruppen kunne arbejde relativt autonomt. Disse krav og designvalg skabte et solidt fundament for opbyggelsen af systemet. Et eksempel på dette kunne være den tidlige beslutning, at systemet skulle opbygges efter tre-lags modellen. Dette gjorde opbygningen af komponenter overskuelige, og nemme at udskifte. Dette har i særdeleshed kommet til udtryk i form af, at en iterativ arbejdsmetode blev taget i brug, så store dele af systemet måtte refaktoreres efter ny viden blev opnået.

Det kan konkluderes at, det endelige systemet opfylder alle de krav der blev sat i kravspecifikationen. Mobil applikationen kan nemt tilgås og bruges af en vilkårlig bruger, som vil kunne notificeres om tiden til ankomst for en bus ved et valgt stoppested meget præcist. Administratoren kan på en særdeles simpel måde, tilføje busser og ruter til systemet.

Måden hvorpå udregninger bliver abstraheret væk fra mobil applikationen gør, at der skabes en meget lav kobling mellem mobilen og resten af systemet. Dette vil desuden også give en høj samhørighed, da applikationen kun udfører det arbejde der er nødvendigt. Simulatoren er blevet udviklet, da der ikke var tilgængelighed til data fra fysiske busser.

Denne komponent har således ikke kun gjort det muligt at simulere en bus i bevægelse, men også gjort det synligt for udviklerne, at udregninger fungerede korrekt. Ved brug af denne gøres det altså væsentligt nemmere at teste systemet.

Hvorom alting er, kan det konkluderes at dette system ville være særdeles brugbart for enhver person, som ofte kører med en bus. Systemet vil, til hver tid, give brugeren mulighed for at undersøge, hvor en bus er, samt hvornår den ankommer til det stoppested, hvor brugeren ønsker at påstige bussen.

15 Referencer

Der er ikke nogen nævneværdige referencer i dette dokument, da de hjemmesider der henvises til, er vedhæftet som fodnoter. Tilgængæld refereres der til en række dokumenter, som er udviklet af projektgruppen. Disse er listet nedenfor:

- TrackABus_KravSpecifikation.pdf
- TrackABus_Accepttestspecifikation.pdf
- TrackABus_Systemarkitektur.pdf