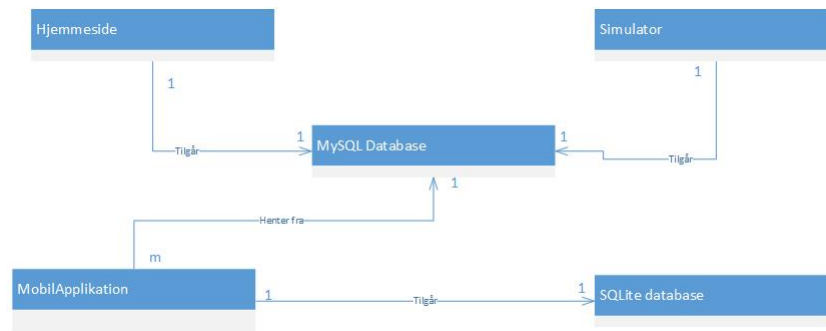


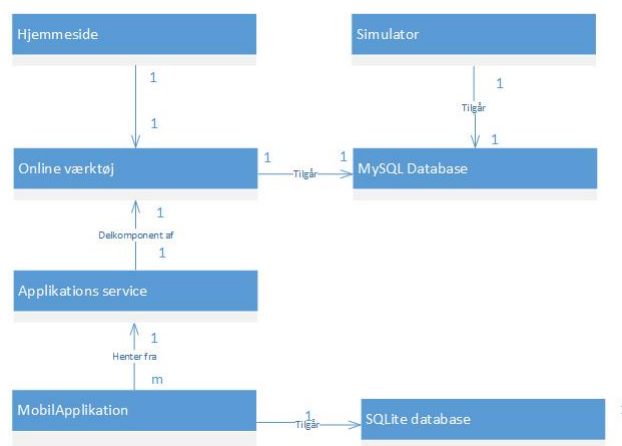
1 Specifikations- og analysearbejdet

Ved påbegyndelse af projektet blev der undersøgt, hvilke domæner projektet skulle bestå af. Dette var i særdeleshed vigtigt, da sytemet består af undersystemer, som udelukkende interagerer over en distribueret database. På figur 1, kan det første udkast til modellen ses.



Figur 1: Første udkast af domænemodellen

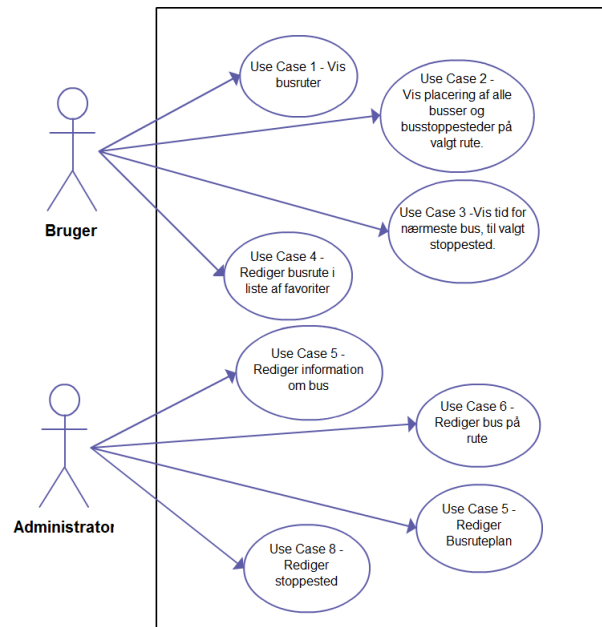
Under udviklingen af systemet, udviklede domæne modellen sig til en ny version. De fleste komponenter forblev de samme, men applikationens tilgang til den distribueret database, blev ændret til at være indirekte. Det vil sige den tilgår en online service, som igen tilgår databasen. Dette sikrer en lav kobling mellem databasen og applikationen, samt en høj samhørighed, da database tilgangen bliver samlet i en komponent ikke direkte knyttet til applikationen. På figur 2 kan den færdige domænemodel ses.



Figur 2: Færdig domænemodellen

Efter udviklingen af domænemodellen, blev der udtænkt de brugssituationer systemet kunne blive udsat for. Simulatoren blev ikke set som en komponent i denne sammen-

hæng, 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 ??, kan det Use Case diagram, systemet er bygget op omkring, følges.



Figur 3: 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 i en brugssituation vil det 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 nogen 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 meget godt.

Til kravene sat i kravspecifikations dokumenter, er der blevet udviklet en accepttestspeci-

fikation, 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.¹

Analyse- og specifikations arbejdet har fungeret rigtigt godt, 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.

¹Sekvensdiagrammer kan ses i kontekst i systemarkitektur dokumentet under 5: *Logisk view*, 6: *Process/Task view* og 8: *Implementerings view*