



BACHELOROPPGAVE

DIASET – MOBIL INSULINDOSEKALKULATOR

FORFATTERE:

GUDBRAND GRANAASEN OLIMB, TROND HAGEHAUGEN OG MICHAEL
KORNBREKK URRANG

Dato: 23.05.2012

SAMMENDRAG

Tittel:	Diaset – Insulindosekalkulator		Dato : 23.5.2012
Deltakere:	Gudbrand Granaasen Olimb		
	Trond Hagehaugen		
	Michael Kornbrekk Urrang		
Veileder:	Harald Liodden		
Oppdragsgiver:	Steinar Dalby		
Stikkord:	Kalkulator, Insulin, Applikasjon, Android, iPhone		
Antall sider/ord: 84 /15 200	Antall vedlegg: 12	Publiseringsavtale inngått: Ja	
<p>Denne Bacheloroppgaven går ut på å utvikle en mobilapplikasjon for barn- og unge med diabetes, slik at beregning av insulindoser som diabetikere skal injisere etter hvert måltid effektiviseres. I dag gjøres nemlig mange av disse utregningene manuelt for hånd, og applikasjonen Diaset har derfor som hensikt å gjøre hverdagen til diabetikere enklere.</p> <p>Applikasjonen benytter seg av en lokal database som inneholder forskjellige typer matvarer og tilhørende verdier i karbohydrater. Databasen som benyttes er matvaretabellen, hentet fra www.matvaretabellen.no og er utgitt av Mattilsynet, Helsedirektoratet og Universitetet i Oslo.</p> <p>Med bakgrunn i målgruppen har designet vært en viktig del av dette prosjektet. Det endelige designet reflekterer gruppens nettside og logo som har vært en rød tråd gjennom prosjektets levetid.</p> <p>Diaset er utviklet for brukere med både iPhone og Android smattelefoner.</p> <p>Den engelske tittelen på prosjektet er Diaset - insulin dose calculator.</p>			

SUMMARY

Title:	<u>Diaset - insulin dose calculator</u>	Date :23.5.2012
Participants	<u>Gudbrand Granaasen Olimb</u> <u>Trond Hagehaugen</u> <u>Michael Kornbrekk Urrang</u>	
Supervisor	<u>Harald Liodden</u>	
Employer:	<u>Steinar Dalby</u>	
Keywords	<u>Calculator, Insulin, Application, Android, iPhone</u>	
Number of pages/words: 84/ 15 200	Number of appendix: 12	Availability: Open
<p>This Bachelor project is to develop a mobile application for children and adolescents with diabetes, in the hope of giving them an easier way for calculating insulin doses that needs to be injected after each meal. Today, a lot of diabetics have to do these calculations manually by hand, but with the Diaset application we hope to make this process more efficient.</p> <p>The application is using a local database that contains different types of food and corresponding values in carbohydrates. The database that is used is a food nutrition table from www.matvaretabellen.no and is released by the Norwegian Food Safety Authority, Norwegian Directorate of Health and University of Oslo.</p> <p>Based on the target group the design of Diaset has been an important part of this project. The final design reflects the projects website and logo. This has been a recurring theme throughout the project life.</p> <p>Diaset is an application optimized for users with both iPhone and Android smartphones.</p>		



diasæt



Gudbrand Olimb
Trond Hagehaugen
Michael Kornbrekk Urrang

Forord

Denne rapporten er utarbeidet som et hovedprosjekt ved Høgskolen i Gjøvik, og er avslutningen på et treårig bachelorstudium innenfor både dataingeniør- og medieteknologi.

Før oppgaveforslagene til årets bachelor ble lagt ut var vi alle i gruppen innstilt på å utvikle et konkret produkt i tillegg til arbeidet med en rapport. Da vi oppdaget "Diaset mobil insulindosekalkulator", ref vedlegg 11 som en mulig oppgave fattet vi raskt interesse og tok tidlig kontakt med oppdragsgiver Steinar Dalby. Vi er i dag svært takknemlige for at vi ble tildelt oppgaven som gjennom hele semesteret har gitt oss en rekke spennende utfordringer, og ikke minst et særdeles godt læringsutbytte.

Vi ønsker å rette en stor takk til alle våre testere av Diaset som har satt av egen fritid til å hjelpe oss med å forbedre applikasjonen til det endelige sluttprodukt:
Stian Moen, Petter Gresmo, Kjersti Brennbakken, Helene Berge, og Ole Kristian Fløystad.

Vi vil også takke vår hovedveileder gjennom prosjektet, Harald Liodden, som i tillegg til å holde oss på stø kurs gjennom prosjektet, også har vært behjelpelig med sin brede kunnskap rundt databaser, et område som har vært svært essensielt for oss i dette prosjektet.

Sist men ikke minst vil vi gi en stor takk til Øivind Kolloen som med sin kunnskap om webstandarder har gitt oss gode råd og anbefalinger underveis i arbeidet med utviklingen av applikasjonen Diaset.

Gjøvik, 23.05.2012


Gudbrand Granaasen Olimb


Michael Kornbrekk Urrang


Trond Hagehaugen

Innhold

Kapittel 1	Introduksjon	1
1.1	Problemområde	1
1.2	Oppgavedefinisjon	2
1.3	Målgruppe i oppgaven og rapporten	3
1.4	Formålet med oppgaven	3
1.5	Bakgrunn og kompetanse	4
1.6	Rammer	4
1.6.1	Tid og kostnad	4
1.6.2	Arbeidsmetode	4
1.6.3	Ansvarsfordeling	5
1.7	Øvrige roller	6
1.7.1	Utviklingsgruppen	6
1.7.2	Oppdragsgiver	6
1.7.3	Veileder	7
1.8	Organisering av rapport	7
1.8.1	Terminologi og definisjoner	7
1.8.2	Utforming av rapport	9
Kapittel 2	Analyse	10
2.1	Ønsker og avgrensning	10
2.2	Valg av utviklingsmodell	11
2.3	Valg av utviklingsmiljø	13
2.4	Programmeringsspråk	14
Kapittel 3	– Kravspesifikasjon	15
3.1	Systembeskrivelse	15
3.1.1	Kort om kravet til applikasjonen	15
3.1.2	Applikasjonens omgivelser	15
3.1.3	Applikasjonens brukere	15
3.2	Rammekrav	16
3.2.1	Ytelse og lagring	16
3.2.2	Sikkerhet og personvern	16
3.2.3	Brukervennlighet	16
3.2.4	Livssyklus	16
3.2.5	Nøyaktighet	17
3.2.6	Portabilitet	17

3.2.7	Utregningsformler.....	17
3.3	Systemets funksjonelle egenskaper.....	19
3.3.1	Use Case diagram.....	19
3.3.2	Høynivå Use Case.....	20
3.3.3	Detaljerte Use Case.....	22
3.3.4	Konseptuelt klassesdiagram.....	27
Kapittel 4	- Design.....	28
4.1	Grovt Design.....	28
4.2	Wireframes (trådkisser).....	28
4.3	Responsiv design.....	29
4.4	Endelig Design.....	31
4.5	Logo og ikon.....	34
4.5.1	Diaset logo.....	34
4.5.2	Diaset ikon.....	34
4.6	Gjennomgang av sidestruktur.....	35
4.6.1	Hovedside index.HTML.....	35
Kapittel 5	Implementering og koding.....	40
5.1	Database.....	40
5.1.1	Matvaretabell.....	40
5.1.2	Databaseløsning.....	40
5.1.3	Databasemodell.....	42
5.1.4	Implementering av matvaretabellen.....	44
5.1.5	Matvaretabellen 2012.....	47
5.1.6	Oppdateringer og endringer i databasen.....	48
5.2	Koding.....	48
5.2.1	Database og lokal lagring.....	48
5.2.2	HTML5 & CSS3.....	51
5.2.3	JavaScript.....	53
Kapittel 6	Testing og kvalitetssikring.....	56
6.1	Selvtesting.....	56
6.2	Eksterne testere.....	57
6.3	Sikkerhetskopi.....	57
Kapittel 7	Publisering av Diaset.....	58
7.1	App Store.....	58
7.2	Google Play.....	59

Kapittel 8	Avslutning	61
8.1	Resultat	61
8.2	Problemer og valg underveis	61
8.2.1	Problemer	61
8.2.2	Valg	62
8.3	Videreutvikling	62
8.4	Kritikk av oppgaven	63
8.5	Gruppeevaluering	64
8.5.1	Arbeidsfordeling og organisering	64
8.5.2	Individuelle refleksjoner og tanker	66
8.6	Konklusjon.....	69
Kapittel 9	Litteraturliste	70
Kapittel 10	Vedlegg.....	72
10.1	Wireframes	73
10.2	C++ program, modifisering av SQL fil.....	74
10.3	Epost mattilsynet	75
10.4	Brukertester	76
10.5	Møtereferater	77
10.6	Statusrapporter.....	78
10.7	Prosjektplan - Diaset, Gudbrand Olimb, Trond Hagehaugen, Michael Urrang	79
10.8	Product backlog	80
10.9	Prosjekt 2011: DiaSet - Del 1-Prosjektbeskrivelse-systemutvikling	81
10.10	Prosjekt 2011: DiaSet - Del 2-Kravspesifikasjon.....	82
10.11	Oppgavebeskrivelse: Oppgave 13 - Diaset - insulinkalkulator for Smarttelefon ...	83
10.12	Prosjektavtale	84

Figuroversikt

Figur 1 Scrum utviklingsmodell	12
Figur 2 Use Case diagram	19
Figur 3 Konseptuelt klassediagram	27
Figur 4 Wireframes av applikasjonen.....	29
Figur 5 Layout eksempel på tvers av enheter	30
Figur 6 Diaset applikasjonen presentert på forskjellige enheter	31
Figur 7 Bilde med og uten sidetittel og undermeny.....	32
Figur 8 Diaset logo.....	34
Figur 9 Diaset ikon.....	34
Figur 10 Index.html	35
Figur 11 Matvarer.html	36
Figur 12 Kurv.html.....	36
Figur 13 Beregning.html.....	37
Figur 14 Bekreft.html	37
Figur 15 Instruksjoner.html.....	38
Figur 16 Informasjon.html.....	38
Figur 17 Info.html.....	38
Figur 18 Personalia.html	39
Figur 19 Innstillinger.html	39
Figur 20 Favoritter.html	39
Figur 21 Databasemodell	42
Figur 22 Endelig databasemodell	43
Figur 23 Applikasjonen på Google Play	60

Tabelloversikt

Tabell 1 Matvaretabellen 2006	44
Tabell 2 CSV formatert tabell av hovedkategori	45
Tabell 3 CSV formatert tabell av matvarer	45
Tabell 4 Ferdig modifisert fil med matvarer.....	46
Tabell 5 Ferdiggenerert kildekode	46
Tabell 6 Matvaretabellen 2012	47

Formeloversikt

Formel 1 Beregning for injiseringsdose.....	18
---	----

Kapittel 1 Introduksjon

1.1 Problemområde

Diabetes type 1 er en kronisk autoimmun sykdom¹ hvor en person har høyt blodsukkernivå grunnet relativ eller absolutt insulinmangel. Ved høyt blodsukker etter måltider må en type 1 diabetiker injisere insulin for å bringe blodsukkeret tilbake til sitt normale nivå.

Bakgrunnen for dette prosjektet er at det i dag finnes veldig mange diabetikere som selv må vite hvor mye insulin de må kompensere med etter hvert måltid for å få det riktige blodsukkernivået i kroppen. Før hver insulininjeksjon, må diabetikeren manuelt regne ut hvor mange internasjonale enheter² insulin han/hun må injisere ved hjelp av insulinpumpe eller sprøyter for å korrigere blodsukkeret. For mange, spesielt barn og eldre, kan dette være en lang og krevende prosess, da foreldre eller hjemmesykepleier må bekrefte at verdien er korrekt før inntaket av insulin kan gjøres. Det finnes i dag informasjon for hvor mye det skal kompenseres for i forskjellige matretter via skriftlige tabeller som er tilpasset diabetikere. Diabetikeren må bruke tabellen samt formelen for utregninger, for å få den riktige verdien å kompensere med. Har måltidet bestått av mange matretter blir denne utregningen raskt en unødvendig tung prosess.

Det er ønskelig med en løsning som tar i bruk den nye teknologien i både telefoner og medisinsk utstyr for å gjøre livet til type 1-diabetikere lettere og gjøre utredningsprosessene kortere.

En hensiktsmessig løsning for å avlaste de rammede bør kunne gjøre følgende:

- Hjelp til med å gi mer korrekte injiseringer.
- Skape trygghet for kjente, kjære og den syke.

Da denne typen diabetikere er helt avhengige av insulin til gitte tider, vil sikkerhet og etterfølgelse av internasjonale standarder være i absolutt høysete.

¹ Kroppens eget forsvarsapparat mot bakterier og virus oppfatter friske celler som fremmede og dreper dem(1)

²En internasjonal enhet av insulin er definert som den biologiske effekten av 1/22milligram ren insulin(2)

Det finnes i dag en del gode hjelpemidler for å måle blodsukkernivået med automatisk overføring av måledata til bla. insulinpumpe, pc og applikasjon på mobiltelefon og for visning av graf for utvikling av blodsukkernivået over tid. Det som er utfordringen er at alle selv må beregne karbohydratmengden og taste resultatet inn på insulinpumpa. Det finnes altså ingen applikasjon som dekker totalbehovet til diabetikeren, selv om det finnes en del applikasjoner som delvis dekker behovet. Noen av de tilgjengelige applikasjonene er enten gode på matvaretabeller eller utregning, samt at de fleste har et dårlig grensesnitt for brukeren. I tillegg er flesteparten av dagens applikasjoner utviklet for internasjonale markeder uten norsk språk som en tilgjengelighet.

Prosjektet vil være basert på systemutviklingsprosjektet "Diaset – mobil insulindosekalkulator"(3) ref. vedlegg 10.

1.2 Oppgavedefinisjon

Hensikten med oppgaven er å utvikle applikasjonen Diaset for å beregne injiseringsdoser av insulin som skal injiseres etter inntak av karbohydrater. Applikasjonen skal basere seg på tidligere kravspesifikasjon fra "Diaset - Del 2-Kravspesifikasjon", ref. vedlegg 10.

Applikasjonen vil hovedsakelig være beregnet for barn- og unge fra 7 års alder, og har som mål å fungere som en avlastning for diabetikere og pårørende som frem til i dag har sett seg nødt til å gjøre disse utregningene manuelt. En viktig del av oppgaven vil være å utforme og utvikle et grafisk grensesnitt som er tilpasset brukergruppen på en slik måte at det er både enkelt og motiverende å bruke.

Måten dette vil bli gjort på er at brukeren av applikasjonen selv skal kunne velge de matvarer og måltider som personen har inntatt ut i fra et grafisk grensesnitt i form av en liste/tabell løsning. Matvarer og måltider som inntas ofte vil også kunne lagres av brukeren i en egen kategori for favoritter, slik at disse på et senere tidspunkt vil være lett tilgjengelig. Applikasjonen vil automatisk gjøre de utregninger som er påkrevd for videre å gi brukeren informasjon om nødvendig antall enheter insulin som personen er nødt til å injisere for å få blodsukkernivået tilbake på et normalt nivå.

For at dette i det hele tatt skal være en forsvarlig løsning vil hver enkelt bruker av applikasjonen ved en førstegangs registrering måtte taste inn en unik konstant/faktor som brukeren har fått av sin fastlege. Denne verdien vil deretter være lagret for brukeren slik at applikasjonen til neste gang er klar for bruk. Brukerens egen faktor er helt sentral for at applikasjonen skal kunne gi vedkommende de rette utregninger for insulinenehetene.

1.3 Målgruppe i oppgaven og rapporten

Opgaven ble primært utviklet for barn mellom 7 og 15 år, men ettersom arbeidet med applikasjonen foregikk, ble målgruppen utvidet til å gjelde fra 7 års alder og oppover. Dette vil si at så å si alle personer med diabetes type 1 kan benytte seg av applikasjonen forutsatt at de har en mobil enhet som smarttelefon eller nettbrett.

Selve rapporten er ment for veileder, oppdragsgiver, sensor, studenter og ellers personer som kan finne den av interesse.

1.4 Formålet med oppgaven

Formålet med oppgaven er å utvikle en enkel og brukervennlig applikasjon for smarttelefoner som vil forenkle hverdagen for personer med diabetes. Applikasjonen blir primært laget for å gjøre det lettere for brukeren å regne ut hvor mange internasjonale enheter insulin han eller hun trenger. En stor del av tyngden ble lagt på å gjøre applikasjonens grensesnitt så enkelt at barn fra 7 års alder og eldre med mindre teknologisk erfaring likevel skal kunne benytte seg av applikasjonen.

1.5 Bakgrunn og kompetanse

Vi er tre studenter ved Høgskolen i Gjøvik som har utviklet applikasjonen, henholdsvis Gudbrand Granaasen Olimb, Trond Hagehaugen og Michael Kornbrekk Urrang.

Gudbrand og Trond studerer bachelor i dataingeniør hvor de tidligere har programmert i språkene C og C++. Michael Kornbrekk Urrang studerer bachelor i Medieteknologi og har tidligere erfaringer fra programmeringsspråket Java.

1.6 Rammer

1.6.1 Tid og kostnad

Det er fastsatt en tidsramme for prosjektet fra januar til mai 2012. Rammene rundt eventuelle kostnader vil bli tatt i samhandling med oppdragsgiver. Utviklingsgruppen har tilgang til både iPhone- og Android baserte telefoner, og det vurderes derfor som lite sannsynlig at kostnader knyttet til teknisk utstyr vil påløpe. Utviklingsgruppen har valgt gratis utviklingsmiljø, ref. kapittel 2.3, og eventuelle kostnader for å kode applikasjonen vil derfor ikke påbeløpe. For opplastning til App Store har utviklingsgruppen ingen egen utviklerkonto, og oppdragsgiver har sagt seg villig til å dekke kostnaden for en utviklerkonto hvis applikasjonen skal lastes opp på App Store.

Utviklingsgruppen har fått tildelt eget webområde for prosjektet av Høgskolen i Gjøvik.

Det ble derimot valgt å bruke eget webområde som hjemmeside for prosjektet, da utviklingsgruppen i fellesskap kom frem til at det vil spares tid på å oppdatere nettsiden på eget webområde.

1.6.2 Arbeidsmetode

Utviklingsgruppen har gjennom hele prosjektet jobbet tett sammen og på denne måten sikret at arbeidsfordelingen delt på hver av gruppe medlemmene dermed har blitt tilnærmet like stor.

I startfasen gikk mye av tiden på å danne seg et godt fellesgrunnlag og en god forståelse av oppgaven vi hadde fått. Det ble diskutert hvordan vi på en best mulig måte skulle klare å gjennomføre prosjektet, hvilke tidligere erfaringer, teknologiske kunnskaper og ferdigheter hver av gruppe medlemmene besatte.

Vi valgte å benytte oss av systemutviklingsmodellen Scrum i dette prosjektet. Arbeidsmetoden har naturlig nok fulgt de prinsipper som kjennetegner dette rammeverket. Sprint er et nøkkelord i Scrum og passet gruppens tanker om arbeidsmetode, derfor ble Scrum et naturlig valg som systemutviklingsmodell. Den totale arbeidsmengden har blitt delt inn i mindre intervaller der gruppen sammen har jobbet for å gjøre ferdig sprinter som til slutt dannet det endelige sluttproduktet. Mer om hvordan vi jobbet etter Scrum prinsippet finnes i kapittel 2.2.

Arbeidet underveis har blitt løst via fellesmøter hvor hele utviklingsgruppen har vært samlet på gruppe- og datarom ved Høgskolen. Enkelte oppgaver har blitt løst individuelt, men gruppen har gjennom hele prosjektet hatt bra kjemi og dratt god fordel av å jobbe sammen og dele tilbakemeldinger om de problemsituasjoner som har oppstått.

I tillegg til møter med både veileder og oppdragsgiver, har det også blitt holdt jevnlige statusmøter innad i utviklingsgruppen hvor temaer som prosjektstatus, gjenværende oppgaver og tidsfrister har blitt diskutert. Det ble notert referater fra samtlige møter underveis, og enkelte av disse er også publisert på gruppens egen nettside.

1.6.3 Ansvarsfordeling

Ansvarsforholdet er som tidligere skrevet i prosjektplanen, ref. vedlegg 7, er fordelt likt mellom alle personer i utviklingsgruppen. I prosjektet finnes det én utviklingsgruppe, én oppdragsgiver og én veileder. Etter veileders vurdering av prosjektplanen har ansvarsfordelingen blitt oppdatert. Arbeidet med applikasjonen deltes inn i tre ansvarsområder og ble dermed som følger:

Gudbrand Granaasen Olimb: Gruppeleder, ansvar for funksjonalitet med JavaScript.

Michael Kornbrekk Urrang: Designansvarlig, ansvar for koding av HTML og CSS.

Trond Hagehaugen: Databaseansvarlig, ansvar for oppsett og implementering av database.

1.7 Øvrige roller

Utviklingsgruppen består som nevnt tidligere, ref. kapittel 1.5 av tre studenter fra Høgskolen i Gjøvik, Gudbrand Olimb, Trond Hagehaugen og Michael Kornbrekk Urrang.

Oppdragsgiver for prosjektet er Steinar Dalby som har flere års erfaring fra sin jobb i Gitek AS. Veileder for prosjektet er Harald Liodden som er høgskolelektor ved Høgskolen i Gjøvik.

1.7.1 Utviklingsgruppen

Gudbrand Olimb

Epost: gudbrand.olimb@gmail.com

Telefon: 906 65 190

Rolle: Prosjektleder

Trond Hagehaugen

Epost: trond.hagehaugen@gmail.com

Telefon: 958 67 796

Rolle: Kontaktperson

Michael Urrang

Epost: michael@urrang.com

Telefon: 959 08 065

Rolle: Grafikk - og designansvarlig

1.7.2 Oppdragsgiver

Steinar Dalby

Epost: sd@gitek.no

Telefon: 957 87 457

1.7.3 Veileder

Harald Liodden

Epost: harald.liodden@hig.no

Telefon: 61 13 52 93

1.8 Organisering av rapport

1.8.1 Terminologi og definisjoner

Web standarder: Er formelle og tekniske spesifikasjoner som beskriver aspekter ved Internett-sider. Er en samlebetegnelse på blant annet HTML, JavaScript og CSS(4).

JavaScript: Er et skriptspråk som er hovedsakelig for å tilføre dynamiske elementer til nettsider. Helt sentralt i prosjektet Diaset og står for hoved- funksjonaliteten til applikasjonen(5).

Phonegap: Er et utviklingsverktøy for å konvertere kode skrevet som nettsider, om til webapplikasjoner for mobile enheter. Gir også smarttelefonen tilgang til native funksjoner som GPS, kamera, Internett og SMS(6).

HTML5: Er et markeringsspråk for formatering og strukturering av nettsider. Definerer elementer som for eksempel tekst, lister og menyer(7).

CSS3: Er et språk som brukes til å definere utseende på de elementer som er nevnt i et HTML-dokument. Bestemmer alt fra bakgrunnsfarger, skriftstørrelse, rammer med mer(8).

jQuery: Er et rammeverk skrevet av JavaScript-utviklere som på en effektiv måte lar brukere laste ned og få tilgang til dynamiske funksjoner og ferdigutviklede stilsett (9).

Relasjonsdatabase: Er en database der data ligger lagret i en eller flere tabeller. Med relasjonsdatabaser kan det opprettes tabeller, legge inn data og manipulere data(10).

SQL: SQL står for "Structured Query Language" og er et programmeringsspråk som benyttes i forbindelse med relasjonsdatabaser(10).

Referanseintegritet: Referanseintegritet betyr at verdien i en fremmednøkkel alltid skal være null eller samsvare med en annen tabells primærnøkkel. Ingen fremmednøkkel skal kunne referere til noe som ikke finnes i en annen tabell(10).

Xcode: Utviklerverktoy for å bygge applikasjoner til Apple-baserte produkter som iPhone og iPad(11).

Eclipse: Utviklerverktoy for å bygge applikasjoner til Android-baserte produkter som smarttelefoner, nettbrett og lignende(12).

Native applikasjon: Applikasjon som kjører via enhetens operativsystem og enhetens Firmware³(13).

Webapplikasjon: Program utviklet som en nettside med web standarder og videre konvertert om til det som kan ligne en nativ applikasjon på mobile enheter. Brukes ofte i forbindelse med utvikling av applikasjoner for flere plattformer(14).

Wireframes: Visuell fremstilling av det designet som tenkes av aktuelle elementer for en nettside eller applikasjon(15).

CSV: CSV står for "comma-separated values". En fil i CSV format lagres som tekst og kan inneholde både bokstaver og tall, separert med komma. Dette formatet brukes til å legge inn dataene i en tabell i en database. Hver linje i en CSV formatet fil tilsvarer én rad i en tabell(16).

Nettsky: I vår rapport menes nettsky som datalagring på eksterne servere som har tilknytning til Internett(17).

³ Tar seg av enhetens basisoperasjoner, for eksempel tilgang til Internet, SMS og GPS.

1.8.2 Utforming av rapport

- Font: Calibri
- Linjeavstand: 1,5
- Skriftstørrelser
 - **Overskrift 1:** 24
 - **Overskrift 2:** 15
 - **Overskrift 3:** 13
 - **Tekst:** 12
- Overskrifter skrives i fet skrift.

Kapittel 2 Analyse

2.1 Ønsker og avgrensning

Ved oppstarten av prosjektet ønsket alle parter hurtigst mulig å definere hva applikasjonen ville bestå av, hvordan utseende på applikasjonen burde være og hvor mange funksjoner det skulle legges til rette for. Både oppdragsgiver og utviklingsgruppen hadde flere tanker rundt dette, og det ble derfor avholdt et møte hvor tanker og ideer ble delt.

Oppdragsgiver hadde spesielt ønske om en enkel applikasjon som skulle være brukervennlig for barn. Han ønsket en applikasjon med store knapper, enkel layout og funksjoner som lot brukeren endre design. I tillegg til dette burde det også være lagt til rette for å lagre favoritter samt personalia om brukeren. Knappene skulle være formet med fine farger og ha klare ikoner for hva hver enkelt knapp gjorde. Det viktigste for oppdragsgiver var først og fremst at applikasjonen som tidligere nevnt skulle være brukervennlig for barn, ettersom han selv ønsket produktet til egne barnebarn. Favorittfunksjonen var det oppdragsgiver fokuserte mest på, slik at brukeren kunne ha mulighet til å lagre favorittmåltider som spesielle frokoster og middager uten å måtte velge hvert enkelt matprodukt for hver gang applikasjonen ble brukt.

Oppdragsgivers tanker om applikasjonens design og funksjonalitet sto i tråd med hva utviklingsgruppen hadde sett for seg ved starten av prosjektet. Siden vi godt forsto fra starten av at målgruppen var veldig viktig for oppdragsgiver, tenkte vi straks på klare farger med store knapper. Disse knappene skulle gjøre det enkelt å navigere rundt for brukeren. Vi så for oss en funksjon for *“drag and drop”* hvor brukeren dro de matvarer han/hun hadde spist i *“sitt måltid”* lokalisert i en bunnmeny, men bestemte oss raskt etter samtale med oppdragsgiver at dette kunne vise seg å bli en vanskelig funksjon å lage/implementere. Vi så derfor mot andre løsninger som ville ha tilsvarende funksjonalitet.

I løpet av idémyldringen ble det også lagt tyngde på hva utviklingsgruppen følte de kunne mestre slik at prosjektet ikke skulle mislykkes. Siden utviklingsgruppen ikke har utviklet applikasjoner med HTML og JavaScript tidligere, ønsket hverken oppdragsgiver eller utviklingsgruppen å presse på applikasjonen mye vanskelig funksjonalitet som kunne føre til at prosjektet ikke ble fullført i tide.

2.2 Valg av utviklingsmodell

I oppstartsfasen diskuterte vi hvilken systemutviklingsmodell som skulle benyttes i applikasjonen. Vi så for oss tre forskjellige modeller vi kunne jobbe etter, Fossefall, Inkrementell utvikling og Scrum.

Fossefall er en utviklingsmodell, der kravspesifikasjon ligger i bunn. Prosjektet utvikles steg for steg, gjennom stegene Analyse, Design, Implementering og integrasjon, testing og vedlikehold. Utfordringen med denne modellen er at den er låst, det vil si at man i utgangspunktet ikke skal gå tilbake til steg som er gjennomført(18).

Inkrementell modell er en utviklingsmodell der kravspesifikasjonen ligger i bunn, men utvikles i inkremer⁴. Hvert inkrement i prosjektet følger de samme stegene som i fossefallsmodellen. Dette gjør at utviklingsgruppen raskt får se resultater, og får tidlig vite om prosjektet er på rett vei. Utviklingsgruppen kan prioritere inkrementene etter eget ønske(19).

Scrum er en utviklingsmodell som har mange likhetstrekk med inkrementell modell, men utvikles med sprinter. En sprint er en fast avgrenset tidsperiode i den fastsatte utviklingstiden, og en sprints varighet er vanligvis fra to til fire uker. Hver sprint skal resultere i et inkrement av applikasjonen. Hva som skal utvikles i en sprint er fastsatt av en product backlog. Denne product backlogen er en liste med oppdelte biter av hva som skal utføres i prosjektet. I Scrum har utviklingsgruppen daglige møter hvor det diskuteres hva som skal gjøres. Etter hver sprint har utviklingsgruppen en sprint gjennomgang, der utviklingsgruppen

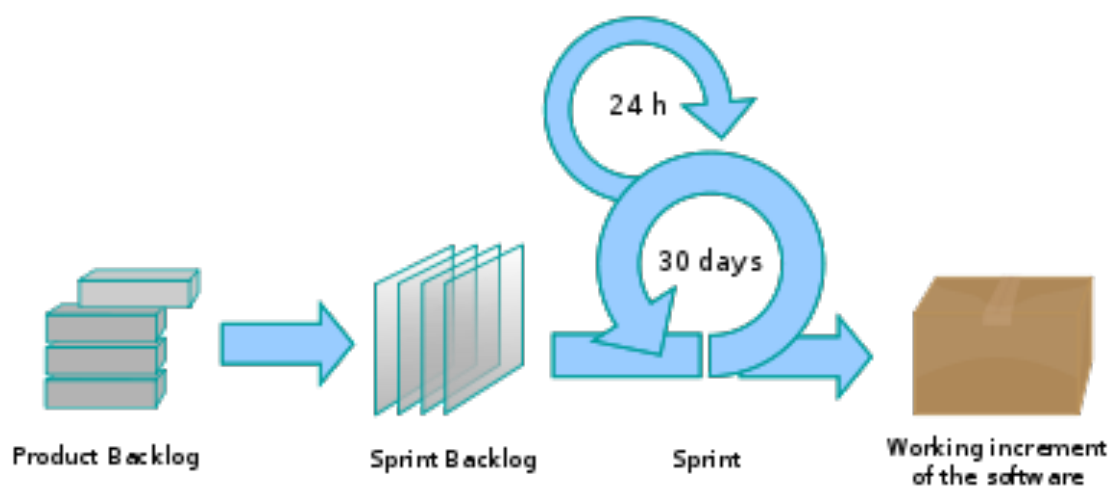
⁴ En del/bit av en applikasjon som skal utvikles

ser tilbake på en gjennomført sprint. Mengden dokumentasjon i Scrum er opp til utviklerne å bestemme(20).

Utviklingsgruppen fikk tilgang til kravspesifikasjonen som ble laget i fjor, og oppdragsgiver ønsket av vi skulle bruke denne kravspesifikasjonen som utgangspunkt i vår applikasjon. Vi følte at det ikke kunne lages en kravspesifikasjon vi kunne følge slavisk hele veien, siden det var stor sannsynlighet at noe ville endres underveis i prosjektet. Vi ville da ha gått tilbake ett eller flere steg, noe som gjør at Fossefallsmodellen ikke ble en aktuell metode å utvikle prosjektet etter.

Inkrementell modell har vi vurdert som en meget aktuell modell for vårt prosjekt. Vi har kravspesifikasjonen i bunn, men følte at hvert inkrement burde ha en fast tidsplan. Det at vi ikke kunne holde oss til en fast tidsplan gjorde at denne modellen heller ikke ble valgt.

Scrum som utviklingsmodell ga en god oversikt over vårt prosjekt. Det å legge opp til en fast tidsplan på hva som implementeres underveis var en viktig faktor, da vi ønsket å bruke like lang tid på alle inkrementene i utviklingsprosjektet. Scrum legger også opp til jevnlig møter med oppdragsgiver, noe som vil være gunstig for utviklingsgruppen siden Steinar Dalby til daglig jobber i Gitek AS sine lokaler ved Høgskolen i Gjøvik. Dokumentasjon er opp til utviklingsgruppen å bestemme, og siden vi ønsket noe dokumentasjon, falt det endelige valget på Scrum(20, 21). Product Backloggen er vedlegg 8.



Figur 1 Scrum utviklingsmodell

2.3 Valg av utviklingsmiljø

I starten av prosjektet har forskjellige utviklingsmiljøer blitt testet og vurdert av utviklingsgruppen.

Titanium studio

Er et utviklingsmiljø som baserer seg på JavaScript, HTML og CSS. Selv om Titanium studio baserer seg på web programmering, og applikasjonene er native applikasjoner.

Titanium studio er Open Source program, og er kompatibelt med iOS, Android og Blackberry. Titanium studio kan utvikle applikasjoner på plattformene Det kan kun utvikles Android applikasjoner på Windows PC og iPhone applikasjoner på Mac OS X(22).

Rhomobile

Er et utviklingsmiljø som bruker felles kildekode for alle kompatible plattformer. Kompatible plattformer er iOS, Android, Blackberry og Windows Phone. Rhomobile applikasjoner er native applikasjoner, støtter lokale databaser og bruker Ruby som kildekode(23).

MoSync

Er et utviklingsmiljø som baserer seg på kildekode i C/ C++, HTML 5 og JavaScript. Utviklerne velger selv hvilke av kodespråkene de vil utvikle i. MoSync er kompatibelt med iOS, Android, Windows Mobile, Windows Phone, Symbian, JavaME og Moblin plattformer. Det oppgis på MoSyncs hjemmeside at det også snart vil være kompatibelt med Blackberry. Mosync er primært beregnet for native applikasjoner(24).

Appmobi

Er et utviklingsmiljø som baserer seg på kildekode i HTML 5 og JavaScript. Felles kildekode for både Android og iPhone applikasjoner. Appmobi utvikler applikasjoner som web applikasjoner(25).

Sencha Tools

Er det utviklingsverktøyet vi har fått foreslått av Steinar Dalby som blir brukt av Gitek AS i sine applikasjoner. Sencha Tools er kompatibelt med iOS, Android og Blackberry og baserer seg på kildekode i JavaScript, HTML 5 og CSS3(26).

Eclipse (Windows) og Xcode (Mac) med Phonegap (Cordova)

Phonegap inkluderes i Android og iOS via Eclipse og Xcode, og utvikles i Eclipse og Xcode miljøene. Eclipse og Xcode er gratis utviklingsmiljøer som man kan utvikle forskjellige typer applikasjoner i. Phonegap er et rammeverk som lar deg bruke web standarder til å utvikle applikasjoner og baserer seg på HTML5, JavaScript og CSS3. Phonegap kan i tillegg til å være web basert, også få tilgang til for eksempel kamera, SMS, video og kompass.

Utviklingsgruppen har tidligere erfaring med C/C++ og ville også tilegne seg kunnskap i HTML5, JavaScript og CSS3. Etter at utviklingsgruppen hadde testet ut utviklingsmiljøene, valgte utviklingsgruppen å benytte seg av Appmobi. Dette utviklingsmiljøet virket oversiktlig og greit å forstå, og baserte seg på nettopp HTML5, JavaScript og CSS3. Applikasjonen kompiles i nettsky og kan lastes inn på mobilen direkte via nettskyen for testing og kjøring.

Etter at vi hadde satt oss inn i Appmobi fant vi ut at kompilering av applikasjonen ble veldig tidkrevende å få testet kontinuerlig. Utviklingsgruppen valgte da å forkaste Appmobi som utviklingsverktøy til fordel for "Eclipse og Xcode med Phonegap (Cordova)". Denne avgjørelsen ble tatt i en tidlig fase av prosjektet, slik at prosjektet ikke ble hindret på grunn av mye endring av kode.(11, 12, 27)

2.4 Programmeringsspråk

Språkene som applikasjonen kodes i er HTML5, JavaScript, SQL og CSS3. Mer om dette kommer i kapittel 5.2.

Kapittel 3 – Kravspesifikasjon

I dette kapitlet vil vi dekke funksjonaliteten og kravene til applikasjonen.

Kravspesifikasjonen er utarbeidet etter tidligere kravspesifikasjon, “Diaset - Del 2- Kravspesifikasjon”(3).

3.1 Systembeskrivelse

3.1.1 Kort om kravet til applikasjonen

I all hovedsak skal applikasjonen Diaset være et hjelpemiddel for utregning av internasjonale insulindoseenheter for personer med diabetes. Applikasjonen skal fungere lokalt på brukerens egen smarttelefon og gjøre utregningene ut fra hvilke matvarer, aktiviteter og blodsukkernivå brukeren har hatt. Innholdet i applikasjonen vil bestå av en stor sammenhengende modul med flere små tilknyttede funksjoner, samt en favorittfunksjon hvor brukeren kan lagre sine favorittmåltider.

3.1.2 Applikasjonens omgivelser

Applikasjonen skal kunne kjøres både på Android og iOS baserte mobile enheter.

3.1.3 Applikasjonens brukere

Applikasjonen skal kunne benyttes av personer fra 7 års alder og oppover med smarttelefon. For brukeren skal applikasjonen være enkel å bruke, det skal ikke kreves noen form for kursing eller erfaring på forhånd for å benytte applikasjonen.

3.2 Rammekrav

3.2.1 Ytelse og lagring

- Minimumskrav for å kunne kjøre applikasjonen er at brukeren har en smarttelefon med enten Android eller iOS operativsystem.
- Operativsystemets minimumskrav for Android er Android 2.2, mens applikasjonen for iOS produkter vil fungere på alle operativsystemversjoner.
- Applikasjonen skal ikke ta mer diskplass enn maksimalt 10 MB.
- Applikasjonen skal kjøre lokalt på smarttelefonen, slik at ingen nettverksforbindelser skal være nødvendig for at applikasjonen kan kjøre på normal måte.

3.2.2 Sikkerhet og personvern

- Det vil bli lagret personopplysninger i applikasjonen med navn og telefonnummer. Disse dataene skal ikke være tilgjengelig for noen andre - eller tredjepart, siden dataene skal lagres lokalt på egen smarttelefon, og at applikasjonen ikke skal ha tilgang til Internett.

3.2.3 Brukervennlighet

- Applikasjonen skal i hovedsak være utviklet på norsk, men med mulighet for videreutvikling til internasjonale brukere og engelsk språk.
- Programmet skal være så enkelt at brukerne skal være i stand til å benytte seg av all funksjonalitet innen én times bruk av applikasjonen.

3.2.4 Livssyklus

- Det skal være mulighet for å implementere ny funksjonalitet i fremtiden.

3.2.5 Nøyaktighet

- Resultatene fra utregningene forventes å være 100 % nøyaktig etter de formlene som brukes og verdiene som er satt.
- Minst 97 % av verdiene i næringstabellen skal være korrekte.
- Applikasjonen skal benytte seg av standardiserte utregningsformler i kalkuleringen.
- Det forventes en oppetid på databasen på 100 %.

3.2.6 Portabilitet

Programmet skal uten problemer fungere på mobile enheter med Android og iOS operativsystem. Dette tilsvarer produkter som iPhone, iPad, smarttelefoner med Android samt alle nettbrett med Android i bunn.

3.2.7 Utregningsformler

Kalkulering av antall injiseringsdoser regnes ut gjennom flere steg. Brukeren har en egen faktor for utregning av injiseringsdoser basert på blodsuktermålinger. Ut ifra dette beregnes en faktor til, som brukes for utregning av injiseringsdoser basert på antall karbohydrater i et måltid.

500 regelen

En bruker av applikasjonen vil ha en egen gitt faktor som alle diabetikere har fått av sin fastlege. Denne er unik for hver bruker, og vil være konstant. Denne faktoren kalles også insulin - karbohydrat ratio (IK). IK faktoren som brukeren taster inn i vår applikasjon baserer seg på 500 regelen, utarbeidet av dr Paul C Davidsson(28):

$$IK \text{ faktor} = \frac{500}{\text{Antall døgndoser}}$$

Antall døgndoser er ukjent, så denne må regnes ut for å finne ut hvor mange injiseringsdoser som skal kompensere for blodsukkernivået i blodet.

$$\text{Antall døgndoser} = \frac{500}{\text{IK faktor}}$$

100 regelen

Injiseringsdoser skal kalkuleres før ethvert måltid, det må korrigeres for blodsukkernivået i blodet før matinntak. Insulinfølsomheten til brukeren må først utregnes.

$$\text{Insulinfølsomhet} = \frac{100}{\text{Antall døgndoser}}$$

$$\text{Injiseringsdoser (e)} = \frac{\text{Blodsukker (mmol/l)} - 5}{\text{Insulinfølsomhet}}, \text{ hvor 5 er normalt blodsukker}$$

Karbohydrater

For å regne ut injiseringsdoser etter et måltid, benyttes faktoren som brukeren har tastet inn i applikasjonen.

$$\text{Injiseringsdoser (e)} = \frac{\text{Antall karbohydrater}}{\text{IK faktor}}$$

Endelig formel

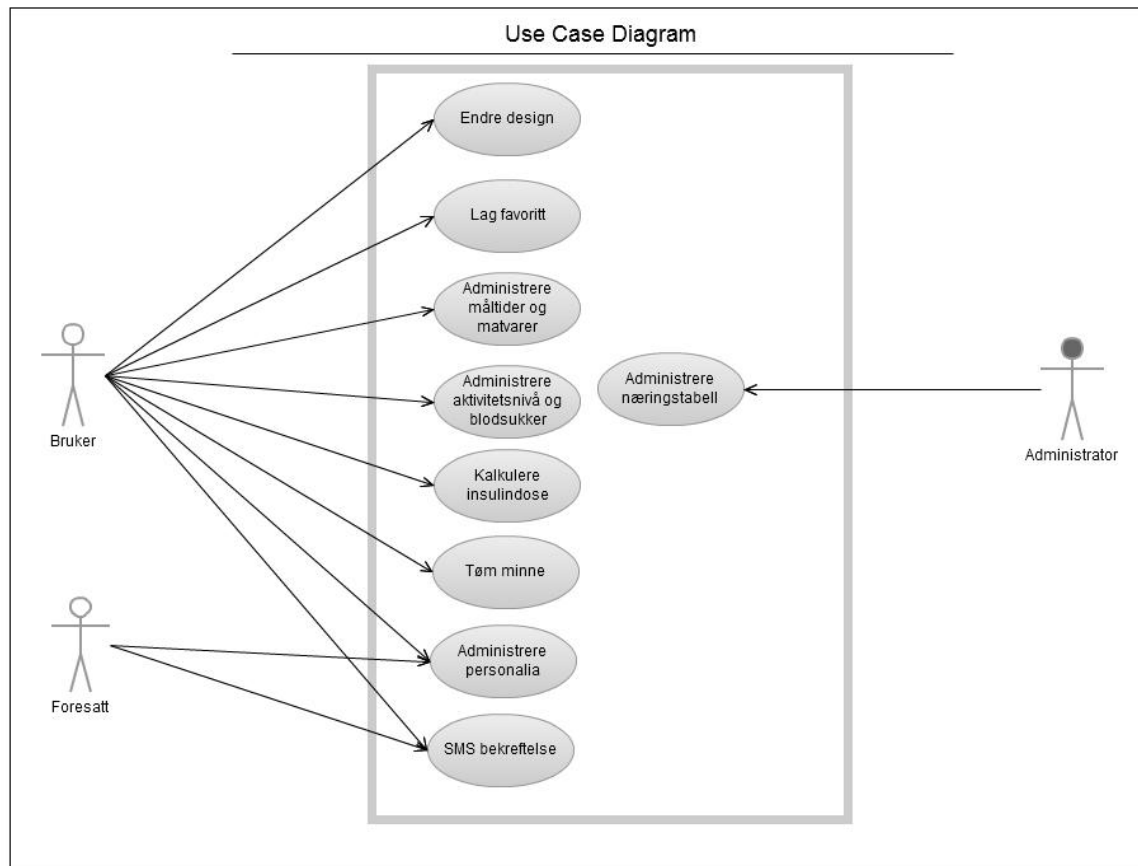
Vi slår sammen formlene vi har og får følgende formel for bruk i applikasjonen:

$$\text{Injiseringsdoser (e)} = \frac{(\text{Blodsukker} - 5)}{\text{Insulinfølsomhet (IF)}} + \frac{\text{Antall karbohydrater}}{\text{IK faktor}}$$

Formel 1 Beregning for injiseringsdose

3.3 Systemets funksjonelle egenskaper

3.3.1 Use Case diagram



Figur 2 Use Case diagram

3.3.2 Høynivå Use Case

Navn: Administrere næringstabell.

Mål: Målet er å vise, legge til og redigere matvarer i næringstabellen.

Primær aktør: Administrator.

Beskrivelse: Næringstabellen er viktig for tilnærmet all funksjonalitet i applikasjonen, og letter arbeidet med å velge matvarer og utarbeide måltider slik at kalkulering av insulindose blir riktig.

Navn: Endre design.

Mål: Brukeren kan endre designet på applikasjonen.

Primær aktør: Bruker.

Beskrivelse: Brukeren skal kunne endre designet på applikasjonen etter eget ønske for å sette et personlig preg på applikasjonen.

Navn: Lag favoritt.

Mål: Brukeren kan lagre favorittmåltider.

Primær aktør: Bruker.

Beskrivelse: Brukeren skal kunne lagre favorittmåltider etter å ha valgt ut flere matvarer som sammen danner et måltid.

Navn: Administrere måltider og matvarer.

Mål: Brukeren kan administrere måltider og matvarer.

Primær aktør: Bruker.

Beskrivelse: Brukeren skal kunne velge ut matvarer og måltider fra næringstabellen og antall gram spist av hver matvare.

Navn: Administrere personalia.

Mål: Brukeren eller foresatt kan administrere personalia.

Primær aktør: Bruker eller foresatt.

Beskrivelse: Brukeren skal kunne lagre/ endre personalia. Med personalia menes navn og telefonnummer til foresatt, samt å lagre sin individuelle korreksjonsfaktor som har blitt gitt av brukerens fastlege.

Navn: Administrere aktivitetsnivå og blodsukkernivå.

Mål: Brukeren kan administrere aktivitetsnivå og blodsukkernivå.

Primær aktør: Bruker.

Beskrivelse: Brukeren skal kunne taste inn sitt blodsukkernivå før hvert måltid, samt velge en eventuelt korreksjonsprosent for aktivitetsnivå (ingen, lav, middels, høy, veldig høy).

Navn: Tøm minne.

Mål: Brukeren kan tømme minne.

Primær aktør: Bruker.

Beskrivelse: Brukeren skal kunne slette personalia, favoritter og designvalg.

Navn: Kalkulere insulindose.

Mål: Brukeren kalkulerer insulindosen som skal bli satt.

Primær aktør: Bruker.

Beskrivelse: Brukeren skal kunne få kalkulert riktig insulindose som skal injiseres, basert på at blodsukkernivået er målt, matvarene (måltidene) er valgt og aktivitetsnivået er valgt.

Navn: SMS bekreftelse.

Mål: Foresatte får en bekreftelsesmelding på antall injiseringsdoser.

Primær aktør: Foresatte.

Beskrivelse: Brukeren får kalkulert riktig insulindose som skal injiseres, basert på at blodsukkernivået er målt, matvarene (måltidene) er valgt og aktivitetsnivået er valgt. Deretter skal applikasjonen kunne sende en bekreftelsesmelding på antall doser insulin som skal injiseres.

3.3.3 Detaljerte Use Case

Navn: Administrere næringstabell.

Mål: Målet er å vise, legge til og redigere matvarer i næringstabellen.

Aktør: Administrator.

Forutsetninger: Databasen må være tilgjengelig for administrator.

Ferdigtilstand: Databasen er oppdatert med nye/endret data eller forblir uendret fordi det kun var et ønske om å se på dataene.

Trigger: Nye eller endrede data fra matportalen(29) eller at brukeren selv ønsker å legge inn spesielle produkter som ikke finnes i databasen.

Spesielle hensyn: Det er to mulige måter å oppdatere databasen på.

Normal hendelsesflyt:

Dette use case har to normale hendelsesflyt.

- 1.1 Administrator endrer dataene direkte i kildekoden.
- 1.2 Programmet kompiles og reinstallerer/oppdateres.
- 2.1 Administrator logger på webområde
- 2.2 Administrator ønsker å vise matvarene i næringstabellen, og trykker på den.
- 2.3 Administrator legger til /endrer/ ser på dataene direkte i MySQL databasen.
- 2.4 Administrator legger inn nygenerert kildekode.
- 2.5 Programmet kompiles og reinstallerer/oppdateres.

Alternative handlingsmønstre:

Det forsøkes å legge inn uriktige data (feil datatyper, for mye eller for lite data, eller dobbeltoppføringer). Siden administrator må gjøre databaseendringer direkte i kildekode vil administrator forbli på punkt 1.2 til feilen er rettet, eller 2.3 til riktige data blir satt.

Navn: Endre design.

Mål: Brukeren kan endre designet på applikasjonen.

Aktør: Bruker.

Forutsetninger: Ingen.

Ferdigtilstand: Brukeren har valgt sitt design.

Spesielle hensyn: Brukeren kan kun velge mellom noen forskjellige alternativer på design.

Trigger: Bruker ønsker å endre design på applikasjonen.

Normal hendelsesflyt:

1. Bruker trykker på innstillinger i hovedmenyen.
2. Brukeren får opp en liste med mulige design på sin applikasjon
3. Brukeren velger sitt design og bekrefter.

Alternative handlingsmønstre:

Brukeren avbryter. Nytt designvalg forkastes og standard design velges til brukeren fullfører denne modulen.

Navn: Lag favoritt.

Mål: Brukeren kan lagre favorittmåltider.

Aktør: Bruker.

Forutsetninger: Brukeren har valgt ut en eller flere matvarer fra matvaretabellen.

Ferdigtilstand: Brukeren har lagret et favorittmåltid

Spesielle hensyn: Ingen

Trigger: Brukeren ønsker å lagre et måltid som brukeren spiser ofte.

Normal hendelsesflyt:

1. Brukeren velger ut en eller flere matvarer.
2. Brukeren taster inn mengden av hver matvare som spises/drikkes i et måltid.
3. Brukeren trykker på "lagre favorittmåltid".

Alternative handlingsmønstre:

- Brukeren avbryter lagringen av favoritter. Går videre til beregning.

Navn: Administrere måltider og matvarer.

Mål: Brukeren kan administrere måltider og matvarer.

Aktør: Bruker.

Forutsetninger: Databasen må være tilgjengelig.

Ferdigtilstand: Brukeren har valgt ut matvarer.

Trigger: Brukeren ønsker å velge ut matvarer for å beregne antall kalorier i et måltid.

Spesielle hensyn: Ingen.

Normal hendelsesflyt:

1. Brukeren velger matvarer fra listemeny.
2. Brukeren blar gjennom og velger matvare(r) etter ønske.
3. Brukeren trykker "Din mat" og får opp alle valgte matvarer.
4. Brukeren taster inn antall gram spist/drukket av aktuell matvare.
5. Brukeren trykker bekreft.

Alternative handlingsmønstre:

- Brukeren velger ikke ut matvarer og trykker "Din mat". Brukeren får beskjed om at "Din mat" er tom, og forblir på punkt 2.

Navn: Administrere aktivitetsnivå og blodsukkernivå.

Mål: Brukeren kan administrere aktivitetsnivå og blodsukkernivå.

Aktør: Bruker.

Forutsetninger: "Din mat" er bekreftet.

Ferdigtilstand: Brukeren har valgt matvarer, aktivitetsnivå og blodsukkernivå

Trigger: Brukeren har bekreftet "Din mat" og ønsker å sette aktivitetsnivå og blodsukkernivå.

Spesielle hensyn: Ingen.

Normal hendelsesflyt:

1. Bruker har bekreftet "Din mat" og får opp ny dialog.
2. Bruker taster inn blodsukkernivå.
3. Bruker velger korreksjonsprosent basert på aktivitetsnivå.
4. Bruker trykker bekreft/kalkuler.

Alternative handlingsmønstre:

- Bruker taster inn feil format på blodsukkernivå. Bruker forblir på punkt 2.

Navn: Administrere personalia.

Mål: Brukeren eller foresatt kan administrere personalia.

Aktør: Bruker eller foresatt.

Forutsetninger: Ingen.

Ferdigtilstand: Foresatt er lagret med navn og telefonnummer, samt korreksjonsfaktor gitt av lege er lagret.

Trigger: Foresattes navn og nummer ønskes lagret, samt korreksjonsfaktor lagres.

Spesielle hensyn: Ingen.

Normal hendelsesflyt:

1. Bruker velger personalia fra hovedmenyen
2. Bruker taster inn navn og telefonnummer på foresatt
3. Bruker taster inn sin korreksjonsfaktor
4. Bruker trykker lagre
5. Bruker får bekreftelse på at personalia er oppdatert/lagret

Alternative handlingsmønstre:

- Bruker prøver å bekrefte feil format på navn (bare bokstaver), telefonnummer (åtte siffer) eller feil korreksjonsfaktor (heltall). Bruker får beskjed om feil format på dataene og må repetere steg 2 og 3 til riktig format er tastet inn.

Navn: Tøm minne

Mål: Brukeren kan fjerne all data om personalia, designvalg og favoritter.

Aktør: Bruker.

Forutsetninger: Ingen, men det må være lagret personalia, design eller favoritter.

Ferdigtilstand: Minne er tømt.

Trigger: Bruker ønsker å fjerne dataene i applikasjonen.

Normal hendelsesflyt:

1. Bruker ønsker å tømme minne, og trykker på tøm minne i innstillinger.

Alternative handlingsmønstre:

- Bruker forsøker å tømme tomt minne. Ingen ting skjer.

Navn: Kalkulere insulindose.

Mål: Brukeren kalkulerer insulindosen som skal bli satt.

Aktør: Bruker.

Forutsetninger: "Din mat" er bekreftet, samt verdiene for aktivitetsnivå og blodsukkernivå er satt.

Ferdigtilstand: Brukeren får ferdigkalkulert sin injiseringsdose for et måltid.

Trigger: Brukeren ønsker å få sin korrekte injiseringsdose for et måltid.

Normal hendelsesflyt:

1. Bruker trykker bekræft/kalkuler i samme dialog som brukeren setter blodsukkernivå.
2. Antall injiseringsdoser vises på skjermen. Formelen er angitt i punkt 3.2.7

Alternative handlingsmønstre:

- Bruker får opplyst feil injiseringsdose, og en alvorlig medisinsk situasjon kan inntreffe.

Navn: SMS bekreftelse.

Mål: Foresatte får en bekreftelsesmelding på antall injiseringsdoser.

Aktører: Foresatt og bruker.

Forutsetning: Injiseringsdose er kalkulert.

Ferdigtilstand: Foresatte får en bekreftelsesmelding på hvor mange doser som skal injiseres med insulin.

Trigger: Brukeren ønsker å få bekreftet at injiseringsdosen som er regnet ut er korrekt.

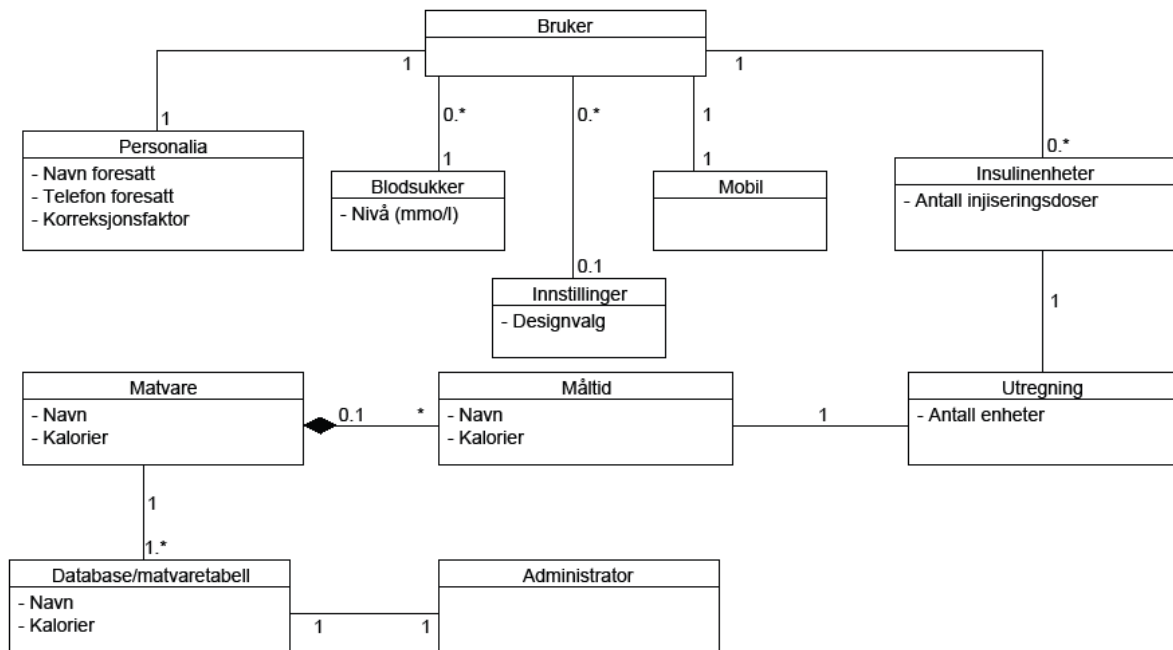
Normal hendelsesflyt:

1. Brukeren får opp sitt resultat av kalkuleringen.
2. Brukeren får spørsmål om bekreftelsesmelding skal selges.
3. Telefonen sender en autogenerert tekstmelding til foresattes telefon satt i personalia.

Alternative handlingsmønstre:

- Bruker ønsker ikke å bekrefte injiseringsdosen. Bruker forblir på punkt 1 til bruker avslutter.

3.3.4 Konseptuelt klassediagram



Figur 3 Konseptuelt klassediagram

Kapittel 4 - Design

4.1 Grovt Design

Det første vi startet med i selve designprosessen var å hånd - skissere hvordan vi opprinnelig så for oss en potensiell layout til Diaset. Det ble naturlig nok en rekke forskjellige forslag, og spesielt var tankene rundt hvordan hovedmenyen skulle se ut svært mange. Oppdragsgivers og prosjektets mål var som kjent at applikasjonen skulle appellere til barn- og unge, og dette var derfor en viktig faktor å ta høyde for. Med bakgrunn i dette ønsket vi å utvikle et design som ikke bare var fult funksjonelt, men også interessant og spennende for målgruppen å ta i bruk.

De første skissene av Diaset applikasjonen ble svært grovt tegnet og uten noen som helst tanker om fargevalg, bakgrunnsbilder eller lignende. Hovedformålet i denne prosessen var trossalt å gi utviklingsgruppen en felles tankegang om hvordan designet til applikasjonen kunne bli løst.

4.2 Wireframes (trådskisser)

Neste steg i designprosessen var å sette opp wireframes for hver enkelt av de ulike sidene til Diaset applikasjonen. Når det gjelder wireframes blir dette oftest brukt til å sette opp selve skjelettet eller rammen til det som skal bli en nettside, og dermed hvordan de forskjellige sidene tenkes å fungere sammen. Det tas med andre ord lite hensyn til farge, grafikk og typografi⁵, men istedenfor en sides funksjonalitet, atferd og hvordan innholdet skal presenteres. Da vi i denne oppgaven bygget vår applikasjon som en webapplikasjon fikk vi nytt godt av wireframes ettersom det på mange måter er en nettside optimalisert for mobile enheter. For utenom å sette opp wireframes for hvordan applikasjonen ville presenteres på en smarttelefon, fikk vi i tillegg skissert hvordan Diaset kunne fremstå på større enheter som iPad og nettbrett med operativsystemet Android.

⁵ Utforming og behandling av bokstaver, skrift og grafiske elementer(30.)

Nedenfor vises enkelte mindre eksempler på noen av de wireframes som ble utviklet. Samtlige wireframes kan i sin helhet finnes under vedlegg 1 .



Figur 4 Wireframes av applikasjonen

4.3 Responsiv design

Om man spør den gjennomsnittlige person på gaten om hva han eller hun forbinder med internett i dag, vil man i de fleste tilfeller få som svar at internett er noe personen har tilgang til via en datamaskin og gjerne brukes i hjemmet eller på jobb. Sannheten er at internett tar over stadig flere enheter som brukes i dagliglivet, eksempler på dette kan være både mobiler, nettbrett og tv. I dag kalles nettopp disse enhetene for kjerneprodukter med hensyn på hvor internett hyppigst blir benyttet. Det interessante er at med dagens teknologiske utvikling vil det ikke være lenge til produkter som hvitevarer med kjøleskap, komfyr eller kaffetraktere alle leveres med display med tilgang til internett.

I forbindelse med denne utviklingen har det de seneste årene åpnet seg et nytt og viktig behov for å optimalisere nettsider uavhengig av skjermstørrelse, kalt responsiv design.

Vår Diaset applikasjon er som nevnt tidligere i rapporten bygget som en web-applikasjon.

I all enkelthet vil dette si at applikasjon er bygget opp som en hvilken som helst nettside, og via utviklingsmiljøer videre konvertert om til en nedlastbar mobilapplikasjon.

Både applikasjonsmarkedene for iPhone- og Android brukere, henholdsvis App Store og Google Play tillater sine brukere å laste ned mobile applikasjoner til diverse mobiler og

nettbrett med forskjellige størrelser. I den forbindelse ble vi enige om å optimalisere Diaset applikasjonen slik at den fungerer på alle tenkelige enheter uavhengig av skjermstørrelse. For å få dette til handler det om å skrive applikasjonens skjelett med HTML, CSS og JavaScript på en måte som gjør at applikasjonen selv registrerer hvilken skjermstørrelse den installeres på. Ved hjelp av denne informasjonen vil designet til applikasjonen formatere seg på en best tenkelig måte for brukeren (31).



Figur 5 Layout eksempel på tvers av enheter

I figuren ovenfor vises et godt eksempel på hvordan designet på nettsider eller applikasjoner kan løses på tvers av enheter med forskjellig skjermstørrelse. Av naturlige årsaker har enheter med stor skjermstørrelse mer plass til å presentere innhold som tekst og bilder enn på enheter med mindre skjermstørrelse. Ut i fra disse bakgrunnsfaktorene handler det om å ikke gi brukeren av for eksempel en mobiltelefon en løsning med mindre innhold, men istedenfor gjøre om på designet slik at brukeren fremdeles har tilgang på alt av innhold, men på en oversiktlig og luftig måte.

Med bakgrunn i responsiv design er Diaset applikasjonen utviklet med web standarder som gjør at applikasjonen kan benyttes på alle tenkelige enheter, uavhengig av skjermstørrelse. Dette vil si at for både hovedsiden og alle listevisninger inkludert i applikasjonen endrer seg i størrelse avhengig av størrelsen på skjermen som applikasjonen blir brukt på. Nedenfor vises en illustrasjon på hvordan Diaset ser ut på henholdsvis pc, nettbrett og mobil på et eldre design av applikasjonen.



Figur 6 Diaset applikasjonen presentert på forskjellige enheter

4.4 Endelig Design

Gjennom hele designprosessen fra håndskisering, wireframes, testing og til slutt ferdig løsning, føler vi med godt grunnlag at vi traff bra med våre ideer allerede fra starten av.

Sammenliknes de første skissene i prosjektet med dagens sluttprodukt er det ikke veldig mye som er forandret. Den største og viktigste merkbare endringen vi valgte å gjøre underveis med designet finnes i hovedmenyen hvor vi gikk bort fra de tre opprinnelige vertikale menyknappene, over til fire større og mer fargerike knapper i et rutenett. Mye av begrunnelsen for denne endringen ligger i målgruppen og hva vi underveis følte ville være den beste løsningen for å tilfredsstille de som brukere.

Diaset bygger på jQuery's rammeverk og består i dag av et enkelt og ryddig design med mye farger og store knapper som skal gjøre det enkelt for nye brukere å ta i bruk applikasjonen.

At Diaset bygger på jQuery biblioteket vil si at applikasjonen benytter en rekke ferdigløsninger for design på blant annet sidetittel og toppmeny, knapper og skjemaer. Alle elementer som ikke er forhåndsdefinert i jQuery rammeverket er selvkodet og lagt til i bunnen av det opprinnelige dokumentet.

På hovedsiden til Diaset finnes logoen plassert øverst, noe som for øvrig er særegent for denne siden da resten av sidene i applikasjonen viser en mindre oversiktstittel med informasjon om hvor brukeren befinner seg i programmets syklus. Under logoen følger så den nevnte hovedmenyen som ble endret underveis i designprosessen. Tanken ved å la hver av de fire menyknappene inneha sin særegne farge, er at brukeren på denne måten vil gjøre seg bedre kjent med menyen for hver gang applikasjonen blir tatt i bruk. Verdt å merke seg er det også at ikonet for knappen "Personalia" endres avhengig av hvilket design som velges under siden for innstillinger. Det er lagt opp til fire ulike temaer og utviklingsgruppen valgte en løsning der gutter, jenter, menn og kvinner alle har hvert sitt tema med passende fargekombinasjoner.

I bunn av applikasjonen finner man en statisk meny som til en hver tid befinner seg nederst uavhengig av hvilken side man befinner seg på. Bakgrunnen for dette valget er at det skal være enkelt for brukeren å navigere seg frem og tilbake mellom matvarer, kurv og forside som er de viktigste sidene under bruken av applikasjonen. Bunnmenyen er for øvrig lik for alle de fire temavalgene i applikasjonen, og verken farge eller ikoner endres derfor her.

Øvrige sider i applikasjonen består for det meste av listevisninger og innfyllingsskjemaer for matvarer med tilhørende verdier. Bakgrunnsfargen for både lister, skjemaer og tekst endres avhengig av hvilket temavalg brukeren har lagret. For å ta et eksempel vil dette si at fargen lilla, er en rød tråd gjennom alle sidene for standardtemaet "Jente", men dersom man endrer tema til "Gutt" vil få fargen grønn som gjennomgangstema.

En av jQuery's mange løsninger som vi valgte å ta i bruk er metoden for visning av sidetittel og bunnmeny. For hver av sidene i Diaset vil både sidetittel og bunnmeny til en hver tid være synlig når brukeren først kommer inn på siden. I de aller fleste tilfeller vil dette være praktisk ettersom hovedinnholdet for den representative



Figur 7 Bilde med og uten sidetittel og undermeny

siden har nok av plass til å presenteres. Skjermstørrelsen for smarttelefoner er derimot varierende, og enkelte vil kanskje føle at tittel og bunnmeny er overflødig og derfor tar opp unødvendig mye plass på skjermen. Ved et enkeltklikk på applikasjonen vil derfor sidetittel og bunnmeny minimeres og ikke vises på brukerens skjerm. På denne måten kan brukeren selv velge om han/hun ønsker at hovedinnholdet for siden applikasjonen befinner seg på skal vises i fullskjermsmodus. Når brukeren har lest ferdig innholdet på en side, vil et nytt klikk i applikasjonen igjen åpne for sidetittel og bunnmeny.

4.5 Logo og ikon

4.5.1 Diaset logo

Vi fant tidlig ut at det ville være både nødvendig og gunstig med en egen logo for applikasjonen. Ikke bare kunne dette hjelpe oss som med utarbeidelsen av det endelige designet til selve applikasjonen, men også fungere til markedsføring for Diaset på de forskjellige markeder og nettsider den skulle publiseres på.



Figur 8 Diaset logo

Etter en del skissering fant vi frem til enkelte elementer vi følte passet applikasjonen og målgruppen den skulle utvikles for. Da nettopp målgruppen består av barn- og unge fra 7års alder, valgte vi å gå for en løsning med mye farger og symboler.

Vi er svært fornøyde med prosjektets logo, og mottok også svært gode tilbakemeldinger fra både oppdragsgiver og publikum. Den sterke lilla fargen som er sentral i logoen skulle også etter hvert vise seg å bli gjennomgangstemaet for selve Diaset applikasjonen, og kan på mange måter ses på som en rød tråd gjennom hele designprosessen.

4.5.2 Diaset ikon

Applikasjoner på mobile enheter presenteres i dag som mindre ikoner i en menyliste for brukeren. Da Diaset skulle være tilgjengelig for både iPhone- og Android enheter måtte vi derfor i tillegg til applikasjonens logo, utvikle et eget ikon for visning på smarttelefoner. Etersom ikoner vist på en smarttelefon er relativt lite, ønsket vi å lage et ikon med få detaljer slik at det tydelig skulle komme frem hvilken applikasjon som hørte til ikonet.



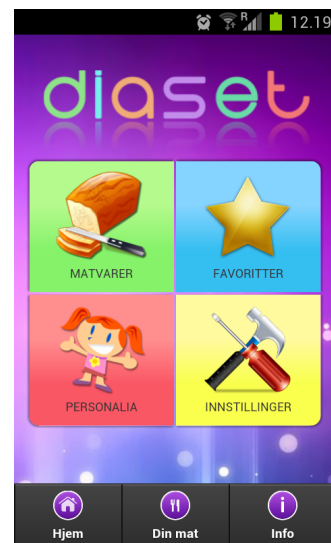
Figur 9 Diaset ikon

Resultatet ble en firkant med forbokstaven til Diaset "D", skrevet med samme font som ble benyttet i prosjektets logo. I senter av ikonet finnes et hjerte som også kan reflekteres tilbake til logoen.

4.6 Gjennomgang av sidestruktur

4.6.1 Hovedside index.HTML

Dette er selve forsiden til applikasjonen og dermed den første siden brukeren blir presentert for ved oppstart. På denne siden finnes Diaset logoen øverst i tillegg til to forskjellige menyer, hovedmeny og bunnmeny. Hovedmenyen består av fire forskjellige knapper, henholdsvis "matvarer, favoritter, personalia og innstillinger". Hver av menyknappene sender brukeren videre til forskjellige undersider i applikasjonen med hver sine funksjoner.

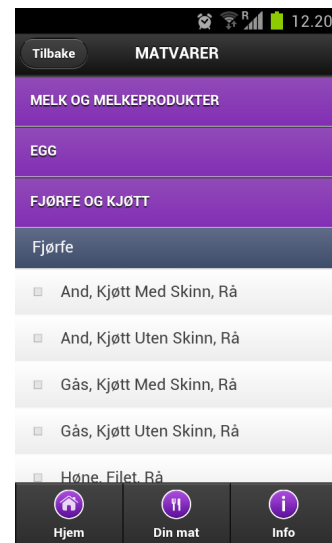


Figur 10 Index.html

Bunnmenyen er helt sentral i vår applikasjon og statisk for samtlige sider i applikasjonen. Med dette menes at bunnmenyen til en hver tid er tilgjengelig for brukeren slik at han / hun på en effektiv måte kan navigere seg frem og tilbake i applikasjonen uavhengig av hvilken side brukeren befinner seg på. Bunnmenyen består av tre knapper, henholdsvis "Hjem, Din mat og Info".

Matvarer.HTML

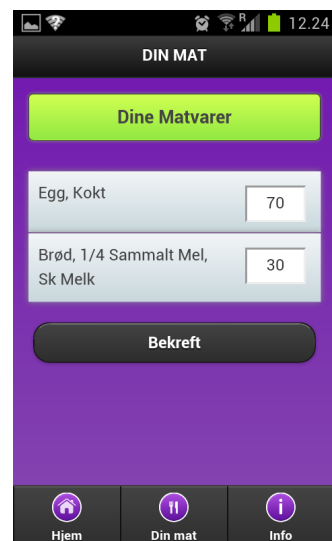
Denne siden henter samtlige matvarer forhåndslagt i databasen og presenterer disse for brukeren i en listevisning. Listevisningen er delt inn i tre forskjellige kategorier for hovedkategori, underkategori og matvarer. Dette vil si at brukeren i første omgang får presentert databasens 10 hovedkategorier for matvarer. Ved å trykke på en hovedkategori åpnes denne og underkategorier tilhørende denne hovedkategorien vises under. Hver av hovedkategoriene har forskjellig antall underkategorier, men med 52 underkategorier totalt blir det ca.5-6 underkategorier per hovedgruppe. Når brukeren klikker på en underkategori får han/hun opp alle de matvarer som hører til i gruppen. Ved siden av matvarenavnet finnes en avkryssingsboks som brukeren huker av dersom matvaren skal tas med i beregningen. Matvaren blir da automatisk lagt til i "Din mat".



Figur 11 Matvarer.html

Kurv.HTML

For denne siden vil alle matvarer som brukeren har valgt presenteres i en ny listevisning. Ved hjelp av en input boks fyller brukeren inn mengden gram som er spist / skal spises av hver enkelt matvare. Når mengden gram for hver matvare er fylt ut trykkes det på bekreft knappen nederst på siden.

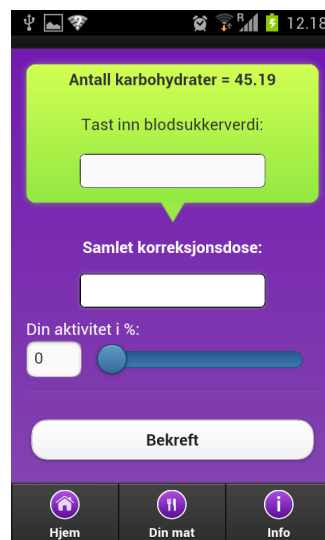


Figur 12 Kurv.html

Beregning.HTML

Øverst på beregningssiden blir totalinnholdet av karbohydrater i måltidet presentert for brukeren. Under denne verdien finnes en rød boks hvor brukeren taster inn sitt aktuelle blodsukkernivå som er helt sentralt for den videre beregningen. Den samlede verdien fra karbohydrater og blodsukkernivå blir deretter vist som "samlet verdi" i en egen boks.

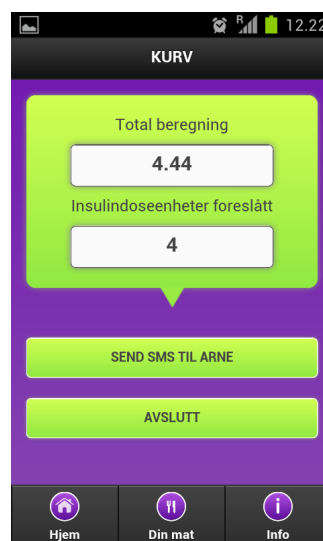
Det siste alternativet som finnes på beregningssiden er en glidebryter hvor brukeren kan justere den samlede verdien etter hvilket nivå med aktivitet som er utført i det aktuelle tidsrommet. Dersom brukeren ikke har vært i særlig aktivitet eller mot formodning ikke ønsker å ta aktivitetsnivå med i beregningen, kan glidebryteren unngås.



Figur 13 Beregning.html

Bekreft.HTML

Dette er den siste siden i beregningsprosessen og gir brukeren resultatet av den totale beregningen. Dersom brukeren ønsker å sende verdien videre til sine foresatte kan dette gjøres ved å klikke på "send SMS" knappen. Hvis man ikke ønsker å sende verdien via SMS trykker brukeren på knappen "Avslutt".



Figur 14 Bekreft.html

Info.HTML

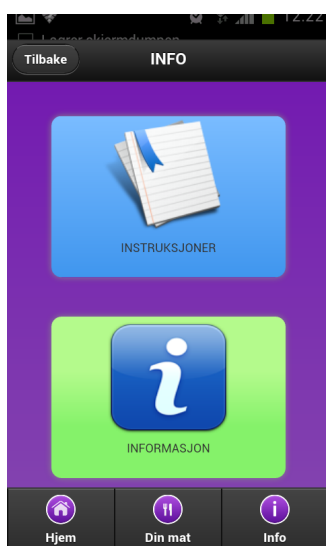
På denne siden blir brukeren presentert for to alternativer i form av store knapper med tydelige ikoner og tekst. Disse knappene er henholdsvis "Instruksjon" og "Informasjon".

Instruksjoner.HTML

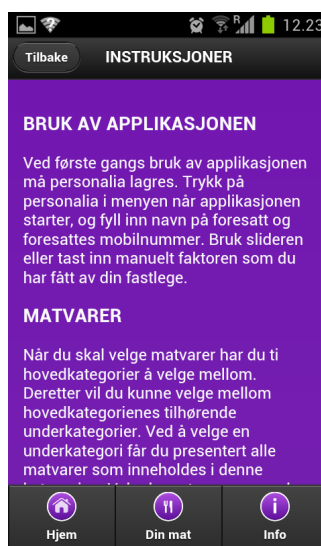
En instruksjonsside for applikasjonen som tar for seg hver enkelt del av prosessen og forklarer brukeren om alle funksjonene i applikasjonen. Det er anbefalt at nye brukere alltid leser gjennom denne siden før applikasjonen tas i bruk.

Informasjon.HTML

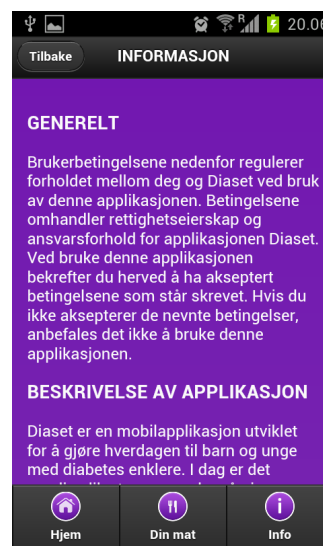
Side for rettighet - og brukerbetingelser ved bruk av applikasjonen Diaset.



Figur 17 Info.html



Figur 15 Instruksjoner.html



Figur 16 Informasjon.html

Personalia.HTML

På denne siden registrerer brukeren sin personalia i form av foresattes navn, foresattes mobilnummer og brukerens personlige IK - faktor som er gitt av vedkommets fastlege.

Figur 18 Personalia.html

Innstillinger.HTML

Fire knapper gir her brukeren mulighet til å endre på selve designet til applikasjonen. Ved å trykke på de forskjellige knappene endres bakgrunnsfargen til applikasjonen etter brukers ønske.

Brukeren har også muligheten til å tømme minne på applikasjonen via "Tøm minne" knappen. Ved å tømme minne slettes all informasjon om brukeren og brukerens evt. favoritter.

Figur 19 Innstillinger.html

Favoritter.HTML

Dersom en bruker ønsker å lagre en matvareliste for fremtidig bruk vil disse bli presentert under denne siden. Alle favoritter vises i en listevissning der favorittens navn i tillegg til antall karbohydrater vises. To knapper gir brukeren mulighet til enten å benytte favoritten, eller slette den fra applikasjonens database.

Figur 20 Favoritter.html

Kapittel 5 Implementering og koding

5.1 Database

Applikasjonen baserer seg i stor grad på databasen, og det ble tidlig klart at databasen i applikasjonen ref. punkt 3.2.1 må kjøre lokalt, da det ikke skulle være nødvendig med nettverksforbindelse for at applikasjonen skulle fungere på normal måte. Vi har sett på to forskjellige måter å løse dette på i kapittel 5.1.2, og føler at løsningen vi har valgt er en god løsning. Databasen som vi har valgt er en HTML5 - database som kjører lokalt i applikasjonen(32).

5.1.1 Matvaretabell

Databasen er en matvaretabell fra 2006 utgitt av Mattilsynet, Helsedirektoratet og Universitetet i Oslo og er hentet fra www.matportalen.no/matvaretabell. Matvaretabellen fra 2006 inneholder ca. 1200 matvarer med tilhørende verdier. Alle verdier i denne tabellen bortsett fra karbohydrater pr 100 gram er ekskludert, da disse ikke har noen betydning for utregning av insulindoser. Matvaretabellen er fritt tilgjengelig og kan benyttes av alle, ref. (33).

Etter 08.03.12 har denne siden blitt flyttet til www.matvaretabellen.no, og inneholder en ny og oppdatert 2012 tabell. Her har det skjedd endringer med fjernede matvarer, nye matvarer og oppdaterte verdier. Disse endringene er foreløpig ikke implementert i applikasjonen. Mer om dette i kapittel 5.1.5

5.1.2 Databaseløsning

Utviklingsgruppen så på flere mulige løsninger på hvordan databasen skulle opprettes, kjøres og vedlikeholdes på hver enkelt enhet. Etter å ha studert forskjellige løsninger fant utviklingsgruppen to løsninger som var aktuelle.

Den første løsningen vi så på var å legge databasen på en server, hvor databasen blir lastet ned til telefonen og oppdateres jevnlig når oppdateringer kommer ut. Fordelene med denne

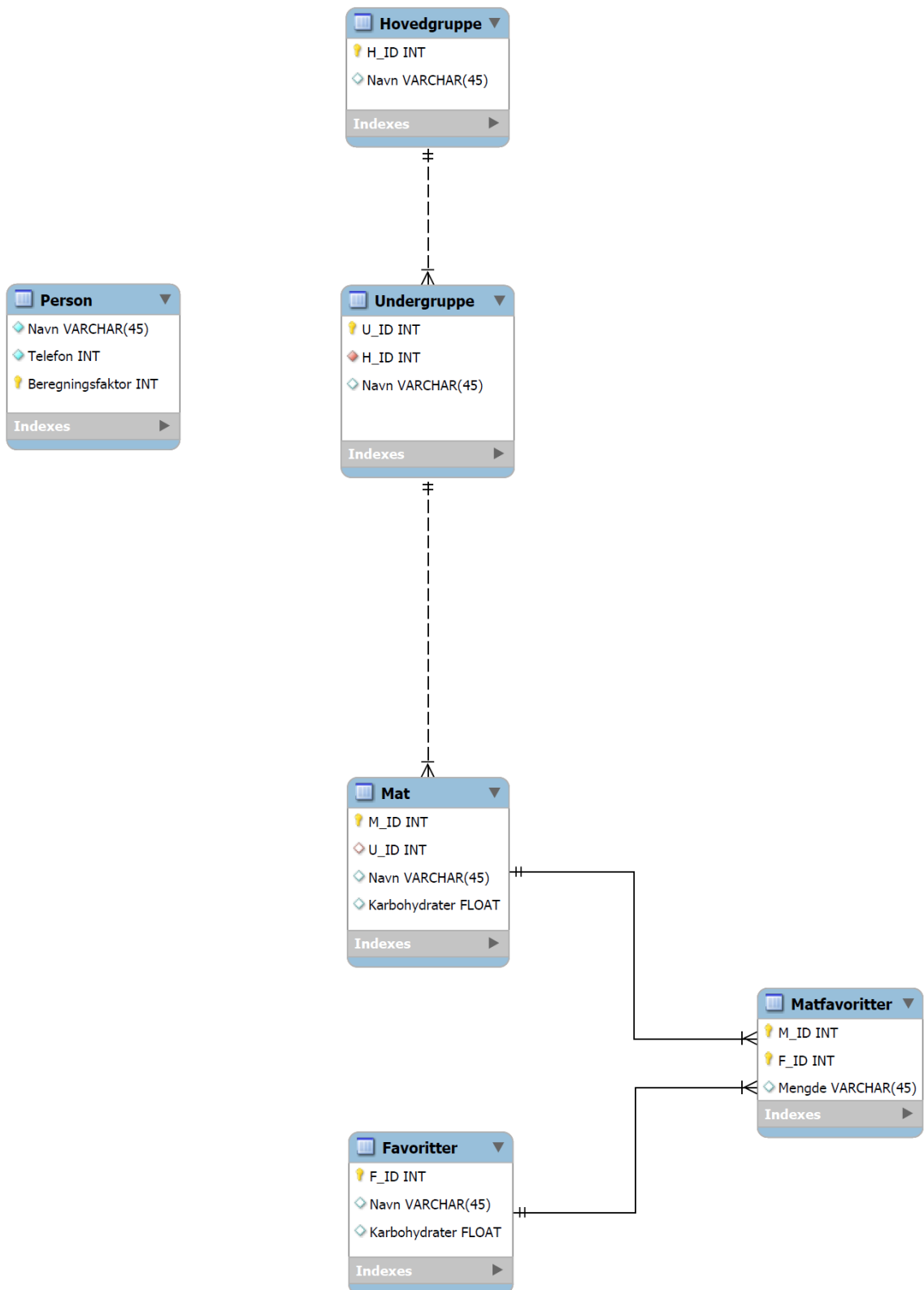
Løsningen er at det ville være lettere å vedlikeholde og oppdatere databasen i applikasjonen til enhver tid. En ulempe vil være at brukeren må ha tilgang til nettverk når oppdateringer kommer ut.

Etter å ha studert løsningen fant utviklingsgruppen ut at det ville være vanskelig med det utviklingsmiljøet og gruppens kunnskaper innenfor kodespråkene å gjennomføre. Med denne løsningen vil man også ha en viss risiko for at det kan forekomme nedetid på databasen, det vil si at man ikke får lastet ned databasen til applikasjonen.

Den andre løsningen av de to mulige løsningene utviklingsgruppen er å kjøre et databasescript for generering av databasen direkte i applikasjonen. Fordelene og ulempene ved løsning to, er at det ville bli mye unødvendig databasekoding i applikasjonen, selv om det ville bli lettere å implementere. Databasen vil også være vanskeligere og mer tungvinn å vedlikeholde for administrator. Med denne løsningen vil brukeren ikke oppleve noen nedetid i forbindelse med databasen.

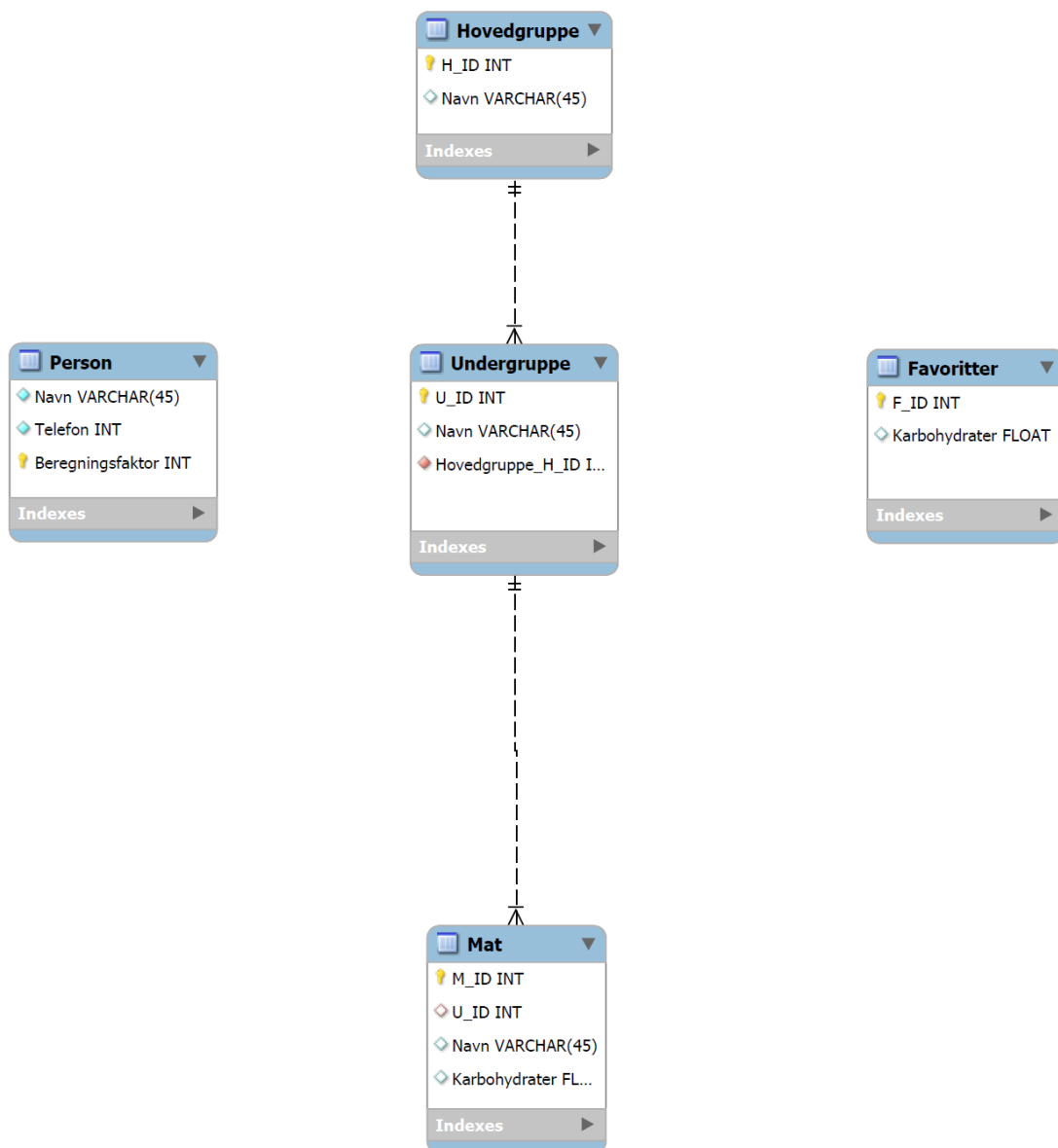
Selv om den første databaseløsningen ville være et naturlig valg grunnet færre ulemper, valgte utviklingsgruppen å bruke løsning nummer to. Grunnen til dette er oppdragsgivers ønske om en lokal database med minst mulig bruk av nettverk. Det ble i samråd med oppdragsgiver besluttet at løsning to også ville være det beste valget da utviklingsgruppen hadde lite erfaring med kodespråkene fra tidligere. Alle ønsket at prosjektet skulle gå fremover og ville derfor ikke lage et stort hinder i prosessen med utviklingen av applikasjonen.

5.1.3 Databasemodell



Figur 21 Databasemodell

Denne databasemodellen er den første som ble laget. Databasen er normalisert til 3 tabeller for mat, samt en favoritt-tabell med en koblingstabell. Denne databasemodellen gikk utviklingsgruppen bort i fra, og det er kun matvaretabellene som er implementert med JavaScript og SQL. Personalia og favoritter er løst med bruk av lokal lagring istedenfor. Denne måten fant utviklingsgruppen det enklest å implementere. Endelig databasemodell ble som følger:



Figur 22 Endelig databasemodell

5.1.4 Implementering av matvaretabelen

Utviklingsgruppen har gjennomført flere steg i prosessen med å lage og implementere databasen til applikasjonen.

1. Lastet ned hele databasen i Excel format fra nettsiden. Under ser vi hvordan Excel formatet ser ut (før 8.3.12)

MATVARETABELLEN 2006																										
Navneliste for næringsstoffer i eget ark																										
Forkortelser: M = Manglende verdi, Ref = Referanse																										
	Netto	Ref	Vann	Ref	Energi1	Ref	Energi2	Ref	Protein	Ref	Fett	Ref	Mettet	Ref	Trans	Ref	Enumet	Ref	Flerum	Ref	Kolest	Ref	Karbo	Ref	Stivel	R
	%		g		kJ		kcal		g		g		g		g		g		g		mg		g		g	
1	1 Melk og melkeprodukter																									
2	1.1 Melk og melkebasert drikke																									
3	01.171	0	85	117	248	60	58	60	2,8	117	0,1	117	0,1	50	0	50	0	50	0	50	1	50	11,6	117	0	1
4	01.170	0	90	117	185	60	44	60	3,2	117	1,5	117	0,9	117	0,1	117	0,4	117	0	117	7	117	4,4	117	0	1
5	01.120	0	84	117	281	60	66	60	2,9	117	1,3	117	0,8	117	0,1	117	0,4	117	0	117	6	117	10,8	117	0	1
6	01.082	0	90	117	186	60	44	60	3,3	117	1,5	117	0,9	117	0,1	117	0,4	117	0	117	7	117	4,4	117	0	1
7	01.099	0	84	117	288	60	68	60	2,9	117	1,3	117	0,8	117	0,1	117	0,4	117	0	117	6	117	11,2	117	0	1
8	01.013	0	88	117	250	60	60	60	2,9	117	3,3	117	2,3	50	0,1	50	0,6	50	0,1	50	5	117	4,6	117	0	1
9	01.001	0	88	117	277	60	66	60	3,2	117	3,9	117	2,4	117	0,2	117	1,1	117	0,1	117	16	117	4,6	117	0	1
10	01.109	0	86	117	319	60	77	60	3,2	117	5	117	3,1	117	0,2	117	1,3	117	0,1	117	21	117	4,7	117	0	1
11	01.196	0	80	81	362	60	86	60	4	81	2,3	81	1,4	81	0,1	81	0,7	81	0	81	7	81	11,8	81	0,4	8
12	01.084	0	88	117	261	60	63	60	3,1	117	3,8	117	2,3	117	0,2	117	1	117	0,1	117	16	117	4	117	0	1
13	01.085	0	92	117	139	60	33	60	3,2	117	0,4	117	0,2	117	0	117	0,1	117	0	117	4	117	4,1	117	0	1
14	01.083	0	88	117	261	60	63	60	3,1	117	3,8	117	2,3	117	0,2	117	1	117	0,1	117	16	117	4	117	0	1
15	01.002	0	90	117	192	60	46	60	3,3	117	1,5	117	0,9	117	0,1	117	0,4	117	0	117	7	117	4,7	117	0	1
16	01.192	0	91	117	160	60	38	60	3,3	117	0,7	117	0,4	117	0	117	0,2	117	0	117	3	117	4,6	117	0	1
17	01.118	0	90	117	193	60	46	60	3,3	117	1,5	117	0,9	117	0,1	117	0,4	117	0	117	7	117	4,8	117	0	1
18	01.119	0	90	117	193	60	46	60	3,3	117	1,5	117	0,9	117	0,1	117	0,4	117	0	117	7	117	4,8	117	0	1
19	01.104	0	83	117	305	60	72	60	3,9	117	1,4	117	0,9	117	0,1	117	0,4	117	0	117	7	117	11	117	0,4	1
20	01.079	0	82	117	328	60	78	60	4	117	1,5	117	0,9	117	0,1	117	0,4	117	0	117	7	117	12	117	0	1
21	11.001	0	87	509	296	60	71	60	1,3	509	4,1	509	1,8	509	M	20	1,6	509	0,5	509	16	509	7,2	50	0	1
22	01.208	0	88	80	253	60	60	60	1,4	80	3,2	80	1,3	80	M	20	1,1	80	0,6	80	13	50	6,5	80	M	2
23	01.206	0	87	80	273	60	65	60	1,5	80	3,3	80	1,7	80	M	20	0,9	80	0,4	80	13	50	7,4	80	M	2
24	10.021	0	83	71	302	60	71	60	2	71	1	71	0,8	80	0	50	0,1	80	0	80	0	80	13	71	0,6	8
25	01.197	0	84	81	294	60	70	60	3,4	81	1,6	81	1	81	0,1	81	0,4	81	0	81	7	81	10,2	81	0,2	8
26	01.127	0	85	117	265	60	63	60	3,3	117	1,5	117	0,9	117	0,1	117	0,4	117	0	117	7	117	9	117	0	1
27	01.008	0	85	117	270	60	64	60	3,4	117	1,5	117	0,9	117	0,1	117	0,4	117	0	117	7	117	9,2	117	0	1
28	01.003	0	91	117	141	60	33	60	3,3	117	0,1	117	0,1	117	0	117	0	117	0	117	1	117	4,8	117	0	1
29	1.2 Yoghurt																									
30	01.074	0	81	117	337	60	80	60	3,9	117	1,4	117	0,9	117	0,1	117	0,4	117	0	117	6	117	12,9	117	0	1

Tabell 1 Matvaretabelen 2006

2. Matvaretabellen ble så inndelt i tre Excel ark med hovedkategori, underkategori og matvarer, for videre å lagres i CSV format. Under ser vi hvordan underkategori og matvarer ser ut, kolonnene tilsvarer meta - dataene, men har ikke overskrifter i CSV format.

	A	B	C	D
1	1	1	Melkebasert drikke	
2	2	1	Yoghurt	
3	3	1	Fløte og rømme	
4	4	1	Ost	
5	5	2	Egg	
6	6	3	Fjørfe	
7	7	3	Lam	
8	8	3	Storfe	
9	9	3	Svin	
10	10	3	Diverse	
11	11	3	Tilberedt kjøtt	
12	12	3	Pølser	
13	13	3	Produkter og pålegg	
14	14	4	Rå fisk	
15	15	4	Skalldyr	
16	16	4	produkter og pålegg	
17	17	4	Hel fisk	
18	18	4	tilberedt fisk	

Tabell 2 CSV formatert tabell av hovedkategori

	A	B	C	D
1	1	1	Biola, syrnnet melk, blåbær, mager	58
2	2	1	Biola, syrnnet melk, naturell	44
3	3	1	Cultura, syrnnet melk, bringebær	66
4	4	1	Cultura, syrnnet melk, naturell	44
5	5	1	Cultura, syrnnet melk, pære	68
6	6	1	Geitemelk, langtidsholdbar	60
7	7	1	Helmelk	66
8	8	1	Kaffemelk, 5%	77
9	9	1	Kakao av lettmelk, tilberedt	86
10	10	1	Kefir, syrnnet melk	63
11	11	1	Kulturmilk, skummet	33
12	12	1	Kulturmilk, syrnnet melk	63
13	13	1	Lettmelk, 1,5 %	46
14	14	1	Lettmelk, ekstra lett, 0,7%	38
15	15	1	Lettmelk, laktoseredusert, 1,5 %	46
16	16	1	Lettmelk, langtidsholdbar, 1,5 %	46
17	17	1	Milkshake, jordbærsmak	72
18	18	1	Milkshake, sjokoladesmak	78

Tabell 3 CSV formatert tabell av matvarer

3. Alle Excel arkene ble lastet opp på webområde for generering av tabellene i MySQL. Alle tabellene ble så lastet ned igjen som ferdig generert SQL script for oppretting og innsetting i tabellene.

4. Da utviklingsgruppen kodet i JavaScript, måtte vi endre de nedlastede SQL scriptene til JavaScript for generering av databasen i applikasjonen. Det ville ta for lang tid å manuelt endre 1200 linjer med SQL script, og utviklingsgruppen modifiserte derfor disse filene slik at resultatet ble som følger:

```
(279, 13, "Spekeskinke", 0)
(280, 13, "Stabburpølse", 2.9)
(281, 13, "Svinekam, røkt", 0.3)
(282, 13, "Sylte", 0)
(283, 13, "Wienersnitzel, rå", 5.8)
(284, 13, "Biff stroganoff", 3.5)
(285, 13, "Cheeseburger, med brød, dressing mv, kjøpt", 24.3)
(286, 13, "Fårikål av lammekjøtt", 3.4)
(287, 13, "Hamburger, dobbel, med brød, dressing mv, kjøpt", 18.3)
(288, 13, "Hamburger, ekstra kjøtt, med brød, dressing mv, kjøpt", 16.5)
(289, 13, "Hamburger, enkel, med brød, dressing mv, kjøpt", 27.6)
(290, 13, "Kyllingburger, med brød, dressing mv, kjøpt", 24.8)
(291, 13, "Kyllinggyte, med tomat, løk, sopp", 2.5)
(292, 13, "Lammefrikassé", 2.4)
(293, 13, "Lasagne med kjøttdeig", 10.9)
(294, 13, "Lasagne med kjøttdeig, fryst, kjøpt", 11.5)
(295, 13, "Lungemos, kjøpt", 1)
(296, 13, "Lys lapskaus, storfekjøtt 10% fett", 4.4)
(297, 13, "Moussaka, med kjøttdeig", 10)
(298, 13, "Pasta med kalkun, ostesaus, fryst, kjøpt", 12.6)
(299, 13, "Pasta med kjøttsaus, Bolognese, fryst, kjøpt", 14.4)
(300, 13, "Pytt i panne, stekte poteter med kjøtt og løk, fryst, kjøpt", 16)
(301, 13, "Ris med biffkjøtt, søtsur saus, fryst, kjøpt", 17)
(302, 14, "Blåkveite, rå", 0)
(303, 14, "Brisling, rå", 0)
(304, 14, "Havål, rå", 0)
```

Tabell 4 Ferdig modifisert fil med matvarer

Utviklingsgruppen utviklet et program i C++ som leser inn filene med SQL scriptene, endrer disse og skriver ut en ny fil med JavaScript syntaks for databasen. Denne programkoden finnes i vedlegg 2. Under ser vi hvordan JavaScript kildekoden ble sendt ut:

```
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (307, 14, "Laks, oppdrett, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (308, 14, "Laks, vill, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (309, 14, "Makrell, juli-sept(fet), rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (310, 14, "Makrell, mai-juni, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (311, 14, "Pigghå, håbrann, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (312, 14, "Røye, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (313, 14, "Sild, fetsild, sommersild, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (314, 14, "Sild, storsild, vintersild, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (315, 14, "Ørret, oppdrett, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (316, 14, "Ål, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (317, 14, "Blålange, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (318, 14, "Breiflabb, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (319, 14, "Brosme, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (320, 14, "Gjedde, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (321, 14, "Horngjel, rå", 0);');
transaction.executeSql('INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES (322, 14, "Hvitting, rå", 0);');
```

Tabell 5 Ferdiggenerert kildekode

5.1.5 Matvaretabellen 2012

Den nyeste oppdaterte matvaretabellen ble lagt ut 8.3.2012. Problemet med den nye tabellen er at den ikke er tilgjengelig i samme nedlastbare Excel format som tabellen fra 2006. Den nye versjonen med oppdaterte verdier for 2012 er ikke bygd opp på samme måte som tidligere, noe som gjør at alle matvarene ligger uten kategorier i 2012 versjonen.

	A	B	C	D	E	F	G
1	Matvaretabellen 2012						
2	MatvareID	Matvare	Spiselig del	Ref	Vann	Ref	Kilojoule
3			%		g		kJ
4	01.001	Helmelk, 3,5 % fett	100	0	87	117	263
5	01.002	Lettmelk, 1,2 % fett	100	0	90	117	180
6	01.003	Skummet melk	100	0	91	117	141
7	01.008	Sjokolademelk	100	0	85	51	250
8	01.011	Yoghurt, naturell	100	0	85	117	311
9	01.013	Geitmelk, langtidsholdbar	100	0	88	117	249
10	01.015	Kremfløte, 38 % fett	100	0	56	117	1491
11	01.018	Iskrem, fløtebasert	100	0	64	70	812
12	01.020	Sjokoladepudding, Piano	100	0	76	117	447
13	01.023	Vaniljesaus, Piano	100	0	72	117	623
14	01.025	Camembert, ost	100	0	49	117	1343
15	01.027	Cheddar, ost	100	0	37	117	1637
16	01.028	Cottage Cheese, ost	100	0	80	117	400
17	01.030	Edamer, ost	100	0	41	117	1458
18	01.032	Gamalost	100	0	43	117	883
19	01.035	Jarlsberg, ost	100	0	41	117	1525

Tabell 6 Matvaretabellen 2012

Utviklingsgruppen har fått bekreftet fra mattilsynet at det muligens vil bli laget en ny versjon i samme format som tabellen fra 2006, men når denne blir tilgjengelig hadde de ingen opplysninger om pr. dags dato, ref. vedlegg 3. Det har blitt besluttet i samråd med oppdragsgiver å ikke oppdatere matvaretabellen før en versjon tilsvarende 2006 versjonen legges ut for nedlastning. Grunnen til at oppdateringen utsettes er at det i dag vil være et altfor omfattende arbeid å endre den nye versjonen.

5.1.6 Oppdateringer og endringer i databasen

For å oppdatere databasen kan administrator/utviklere gjøre oppdateringene manuelt direkte i kildekoden, og ny versjon må slippes på Google Play og App Store.

Dersom det blir gjort endringer opp mot MySQL databasen på webområde kan administrator enten legge til endringene manuelt i kildekoden, eller laste ned hele tabellen i SQL syntaks samt noe manipulering av denne filen for å kunne kjøre den igjennom programmet, ref. vedlegg 2 for å legge til endringene i kildekoden. Ny versjon må slippes på Google Play og App Store.

5.2 Koding

I dette underkapitlet gjennomgår vi i korte trekk kodingen av applikasjonen. De fleste kodesnutter som blir presentert nedenfor er kun utdrag fra enkelte funksjoner.

5.2.1 Database og lokal lagring

Databasekodingen er i SQL, et programmeringsspråk i forbindelse med databaser. Ingen linjer i applikasjonens koding av databasen, inneholder kun SQL syntaks. Dette er fordi man må henter og jobber med databasen via JavaScript. Lokal lagring er en del av HTML5 - databasen som vi har benyttet oss av, et eksempel på dette vises i kapittel 5.2.3.

Opprettelse av tabeller

All databasekoding skjer med Java- og SQL script ved opprettelse av tabeller, hvor det lages en variabel for SQL kall. Det kjøres så en såkalt "transaction" for å lage tabellene. Under ser vi et utdrag fra denne funksjonen:

```

function lagtabeller() {
    var hovedkategori = 'CREATE TABLE IF NOT EXISTS hovedkategori(H_id
    INTEGER NOT NULL PRIMARY KEY , navn TEXT NOT NULL)';

    try {
        // Lager hovedkategoritabell:
        db.transaction(function(transaction) {
            transaction.executeSql(hovedkategori, [], nullDataHandler,
            errorHandler);
        });
    }
}

```

I løpet av utviklingsfasen ble det klart at databasen ikke takler fremmednøkler. Vi besluttet i samråd med veileder å ikke bruke det, da det kun er snakk om tre tabeller som ville hatt fremmednøkler. Det at vi ikke får brukt fremmednøkler, gjør at vi dessverre mister referanseintegriteten i databasen. Måten vi har løst dette på er å hente innholdet i en tabell basert på ID'en i en annen tabell:

```

transaction.executeSql('SELECT * FROM underkategori WHERE HO_id =
'+underRad[i].title+',' ,...)
/* Her henter vi ut underkategoriene til en hovedkategori */

```

Innsetting i tabeller

All innsetting i tabeller kjøres med et kall "transaction.executeSql()" for hver rad i tabellene.

```

function innsetttabell() {
    db.transaction(
        function(transaction) { /* Setter inn verdier i mat */
            transaction.executeSql('INSERT INTO hovedkategori(H_id,navn) VALUES
            (1,"Melk og melkeprodukter");');
            transaction.executeSql('INSERT INTO hovedkategori(H_id,navn) VALUES
            (2,"Egg");');
        }
    );
}

```

Visning av tabellene i HTML

Måten tabellene presenteres til brukeren på, er først en liste med hovedkategoriene, deretter en liste med underkategori, og til slutt en liste med matvarene. Måten vi har løst dette på, er å lage en tabell for alle kategoriene og deretter listene. Det er én liste per hovedkategori, én liste per hovedkategoris tilhørende underkategorier, og én liste per underkategoriens tilhørende matvarer. Tabellen til hovedkategoriene er lagt i HTML dokumentet *matvarer.HTML*, mens tabellene for underkategori og matvarer er lagt i JavaScript-filen *diaset.js*.

```
<!-- Første tabell i HTML fil, hovedkategori -->
<ul id="hovedKategori" data-role="listview" data-theme="a"></ul>
// Under ser vi hvordan underkategori presenteres:

transaction.executeSql('SELECT * FROM underkategori WHERE HO_id =
'+underRad[i].title+''', [], function(transaction, results){
    for (var c = 0; c < results.rows.length; c++) {
        var Rad1 = results.rows.item(c); // Henter alle radene
        var li = document.createElement("li"); // Lager liste
        var ul = document.createElement("ul"); // Lager tabell
        var span = document.createElement("span");

        // Setter underkategoriens ID i tabellen til dens egen id i liste:
        li.setAttribute("id", Rad1['U_id']);
        li.setAttribute("class", "under"); // Setter klasse

        // Setter tabellens id til Matkategori:
        ul.setAttribute("id", "matKategori"+Rad1['U_id']);
        ul.setAttribute("class", "matKat"); // Setter klasse

        // Setter tittelen til Underkategoriens ID:
        ul.setAttribute("title", Rad1['U_id']);
        ul.setAttribute("style", "display: none;"); // Setter style

        // Lager listeelementenes navn:
        var liText = document.createTextNode(Rad1['navn']);
        span.appendChild(liText); // appender listeelementene
        li.appendChild(span);
```



```
document.getElementById("underKategori"+Rad1['HO_id']).appendChild(li);  
// Henter underkategori-liste  
  
document.getElementById("underKategori"+Rad1['HO_id']).appendChild(ul);  
// Henter underkategori-tabell  
span.onclick = CollapserMat;  
// Mulig å vise/skjule underkategori  
}
```

5.2.2 HTML5 & CSS3

Dokumentstrukturen for Diaset er skrevet i markeringsspråket HTML5, som i dag er i ferd med å ta helt over etter at HTML4 har blitt benyttet de siste 11 årene. HTML kan veldig kort forklares som et språk for formatering av nettsider der dokumentene er bygget opp av elementer som igjen er avgrenset med tagger med forskjellige betydninger. Sammen utgjør alle de forskjellige elementene et fullstendig HTML-dokument som kan publiseres på internett enten i form av en nettside, eller applikasjon som har vært tilfelle for vårt prosjekt. Det er nettleseren som både leser og tolker HTML-dokumentene, og det er derfor helt sentralt at alle elementer og tagger er definert riktig slik at nettleseren klarer å tolke det korrekt.

Den relativt ferske oppdateringen fra HTML4 til HTML5 har lenge vært heftig debattert, og det sier seg egentlig selv at elleve år i teknologiens verden er en fryktelig lang periode. Den etterlengtede overgangen til HTML5 har åpnet for en rekke nye muligheter med tanke på både struktur og funksjonalitet. Spesielt er det verdt å merke seg at strukturen med HTML5 gjør dokumenter langt mer oversiktlig enn hva tilfellet var med den foregående versjonen. Grunnen til dette er blant annet at HTML5 kommer med egendefinerte tagger for sentrale elementer som header, navigasjon, sidefelt og footer, elementer som for øvrig alle er hyppig benyttet i vår applikasjon.

For Diaset er samtlige sider i applikasjonen separate HTML-dokumenter som består av en rekke ulike elementer og tagger som nevnt ovenfor. Totalt består applikasjonen av elleve forskjellige sider som alle har lik oppbygning, men med ulikt hovedinnhold for presentasjon. Standard oppsett for hver sidene er som vist nedenfor:

```
<html>
<head>
  <title> Diaset </title>
  <meta data ----- >
</head>
<body>
<header> Sidenavn </header>
<content>
  <h3> Overskrift </h3>
  <form> Skjema </form>
  <navbar> Hovedmeny </navbar>
  <button> Knapp </button>
</content>
<footer>
  <navbar> Navigasjonsmeny
  <ul> <li> Menyelementer </li></ul>
  </navbar>
</footer>
</body>
```

Et hvert HTML-dokument startes og avsluttes med standarddeklarasjonen `<html></html>`. Deretter følger `<header>` elementet som i tillegg til å definere tittel også henter elementer som stilsett og script som skal inkluderes på siden(34). For Diaset hentes stilsettet jQuery og applikasjonens eget JavaScript for database og funksjonalitet innenfor header elementet.

Elementet `<body>` er det som tar opp mest plass i hvert av dokumentene. Innenfor dette elementet finnes alt som er av synlig innhold på hver av sidene. Innholdet her er plassert inn i tagger, og eksempler på dette kan være overskrifter, skjemaer, lister, knapper med mer.

Det siste elementet er <footer> og definerer selve bunninnholdet for applikasjonen. I Diaset footeren ligger det tagger som definerer en listemeny med ikoner som dermed danner den statiske bunnmenyen for hver side i applikasjonen.

CSS3 er språket som definerer designet på de elementer som er beskrevet i HTML-dokumentene. Med CSS3 settes blant annet bakgrunnsfarge, tekstfarge, skriftstørrelse og avstand for ønskede elementer. Da CSS3 overtok for CSS2 åpnet det for en hel haug med nye spennende muligheter som tidligere var en tungvint prosess å få til. For å ta et eksempel måtte man med CSS2 lage transparente (gjennomsiktige) bilder med avrundede kanter for å få til en slik effekt. I CSS3 er det forhåndsdefinert kodesnutter som automatisk gjør dette for deg.

5.2.3 JavaScript

JavaScript er et scriptspråk som brukes for å gi dynamiske elementer i en nettside. JavaScript brukes primært til å lage enkle funksjoner som kan bytte ut, fjerne, endre eller legge inn elementer som for eksempel tekst, i HTML-dokumenter. Vi har hentet mye inspirasjon fra bøkene "JavaScript the Definitive Guide"(35) og "JavaScript for Web Developers 2nd Edition"(36).

Diaset prosjektet bygger på JavaScriptene *diaset.js* og to jQuery JavaScript, *jQuery-1.7.1.min.js* og *jQuery.mobile-1.1.0.js*. *Diaset.js* er et egenutviklet script hvor det meste av funksjonaliteten til applikasjonen er skrevet. Under kodingen av *Diaset.js* var vi veldig bevisste på å bruke like variabler i de forskjellige funksjonene slik at hele koden skulle få et godt helhetsinntrykk, selv om vi på enkelte steder måtte avvike fra dette. Koden er forsøkt bygget opp på en logisk måte, hvor funksjoner er plassert etter applikasjonens bruksprosess slik at det er lett å navigere i dokumentet. En av funksjonene som gir et inntrykk av utviklingsgruppens scripting er "*favoritt*" funksjonen som følger.

```

/* Lagre favorittmåltid */
function favoritt(){
    var a =localStorage.getItem("total"); // Henter totalen i måltidet
    var i = localStorage.getItem('favoritt'); // Henter Favorittmåltid

    var answer = confirm ("Vil du lagre matvarene som et favorittmåltid? ");
    if (answer){ // Henter svar
        var favoritt = prompt("Navn på favoritt",""); // Spør om lagring
        if (favoritt != "") { // Ingen favoritt
            if(i == null){ // Ingen favoritt lagret fra før
                localStorage.setItem("favoritt", 0); // Begynn på 0
                i=0;
                localStorage.setItem("f"+i,favoritt); // Lagrer Navn
                localStorage.setItem("f"+i+"f",a); //Lagre karbohydrater
            }
            else
                ++i; // Teller opp i
            localStorage.setItem("favoritt", i); // Lagrer neste nr 'i'
            localStorage.setItem("f"+i,favoritt); // Lagrer Navnet
            localStorage.setItem("f"+i+"f",a); // Lagrer karbohydrater
            alert("Favoritt lagret!"); // Beskjed til bruker
            window.location.href="beregning.html"; //Send til side
        }
    }
    else
        alert("Det ble ikke laget noen favoritt") //Ville ikke lage fav.
        window.location.href="beregning.html"; // Gå videre til beregning
}

```

De to andre JavaScriptene som ble benyttet var *jQuery-1.7.1.min.js* og *jQuery.mobile-1.1.0.js*. jQuery er et "rammeverk" også kalt et bibliotek som på en oversiktlig og enkel måte lar deg ta full utnyttelse av webstandarden JavaScript. Det briljante med jQuery og andre lignende rammeverk er at selv uten særlig kunnskap eller tidligere erfaringer med skriving av JavaScript, kan man likevel benytte scriptspråkets mange muligheter uten særlige problemer. Årsaken til dette gjenspeiles også i begrepet "bibliotek", og kan forklares med at jQuery inneholder en stor mengde ferdiglagde komponenter og funksjoner. Som følge av dette slipper man selv å skrive en eneste linje med JavaScript for å få tilgang til en hel haug med kreative løsninger til eget bruk, da disse bibliotekene er gratis å bruke.

Rammeverkets eget slagord, "Write less, do more" er for øvrig en kort og presis oppsummering av biblioteket.

Kapittel 6 Testing og kvalitetssikring

I løpet av prosjektet har det blitt gjennomført en rekke ulike tester for å evaluere og forbedre kvaliteten til applikasjonen. Måter dette har blitt utført på er både via selv- og ekstern testing som sammen har åpnet for å finne frem til løsninger med forbedringspotensialer. Dette har i tillegg gitt oss respons på allerede eksisterende løsninger som fungerte som planlagt. Områder som er blitt testet er blant annet brukervennlighet, design, og at applikasjonen er kompatibelt både for iPhone- og Androidenheter.

6.1 Selvtesting

Selvtesting er metoden som hyppigst er blitt benyttet gjennom prosjektets levetid. Prosjektgruppen har til stadighet testet at applikasjonen og dens funksjoner fungerer som planlagt via utviklingsverktøyene Eclipse og Xcode. Her har vi hatt mulighet for å teste og debugge⁶ Diaset ved hjelp av både emulator⁷ og gruppens egne mobile enheter. Det er også benyttet ulike nettlesere som Google Chrome og Safari på pc for å kontrollere at applikasjonens funksjonalitet og responsivt design fungerer vel så bra på store skjermstørrelser, som på smarttelefoner.

Ettersom vi i dette prosjektet har fulgt utviklingsmodellen Scrum, ble det i startfasen satt av en egen sprint for testing, utprøving og evaluering av applikasjonen mot slutten av prosjektet. Som følge av dette har de viktigste testene vært gjennomført etter at beta-versjonen av Diaset ble sluppet, og eksterne testere utenom utviklingsgruppen har derfor i stor grad vært påvirkende i denne fasen.

⁶ Debugge er en form for testing av kode for å finne feil

⁷ Emulator er en virtuell enhet som brukes for å kjøre en applikasjon som om det var en reell enhet.

6.2 Eksterne testere

Potensielle og fremtidige brukere har hatt en viktig rolle under testingen av applikasjonen Diaset. I tillegg til tilbakemeldinger fra medelever fikk vi også i prosjektets siste innlagte sprint testet applikasjonen opp mot prosjektets målgruppe, nemlig barn - og unge fra 7års alder. Tilbakemeldingene vi har fått fra disse gruppene har hatt høy prioritet, og konklusjonene har vært positive.

Etter at betaversjonen ble lagt ut fikk vi tre studenter til å gjøre individuelle tester av applikasjonen. Tilbakemeldingene ble notert som brukertester og er medfølgende i rapporten som vedlegg 4.

6.3 Sikkerhetskopi

Sikkerhetskopi og lagring har vært en viktig faktor i prosjektet. Samtlige medlemmer i utviklingsgruppen har til en hver tid har hatt kopier av både dokumenter og applikasjons kode liggende på egne maskiner. Det har også vært benyttet både Dropbox⁸ og nettskyer for å sikre at ingen informasjon skulle gå tapt underveis. Bruken av disse løsningene har vist seg klokt, da vi ikke har merket noe til tap av data eller informasjon.

⁸ Dropbox er en gratis tjeneste som lar deg sikkerhetskopiere og dele personlig filer med andre personer.

Kapittel 7 Publisering av Diaset

For å formidle Diaset ut til potensielle brukere finnes det ulike marked for både iPhone- og Android baserte enheter. Fremgangsmåten for publisering varierer avhengig av hvilken plattform som ønskes brukt. Nedenfor kommer det en kort forklaring på prosessen for de to aktuelle markedene i dette prosjektet.

7.1 App Store

Apple har svært strenge krav og regler for opplastning av applikasjoner til sitt marked, App Store. For å i det hele tatt kunne utvikle en applikasjon til Apples produkter må utvikleren ha en Mac datamaskin ettersom programvaren for utvikling til deres produkter ikke er tilgjengelig for Windows. I tillegg må det opprettes en Apple ID⁹ med utviklerkonto for å få tilgang til nødvendig programvare. Operativsystemet "Snow Leopard" til Mac er et minimumskrav for å kunne utvikle applikasjoner med Apples utviklerprogram Xcode.

Med dagens teknologiske utvikling oppdateres operativsystemer for både datamaskiner og smarttelefoner svært hyppig. En svak side ved Apple og deres utviklerprogram er derfor at firmaet til en hver tid krever at operativsystemet for både Mac og iPhone er av nyeste tilgjengelig versjon. Dette fører til stadig nye kostnader for utviklere, og vi måtte selv i dette prosjektet oppdatere eget operativsystem to ganger for å ha muligheten til å gjøre Diaset tilgjengelig for brukere med nyeste versjon av iPhone.

Programvaren for utvikling av applikasjoner til Apples produkter er gratis, og kan lastes ned via firmaets egen nettside. Apple tar seg likevel godt betalt dersom du ønsker å publisere din ferdige applikasjon på App Store, og en årsavgift på ca 600 kroner kreves for publisering i App Store.

⁹ Apple ID er et brukernavn som du kan bruke til alt som har med Apple å gjøre.

Hvis utvikleren har planer om å tjene penger på sin applikasjon liggende på App Store er det også verdt å merke seg at Apple tar 30% av alle kommende inntekter. I tillegg stilles det svært strenge krav til hver applikasjon, og en stor andel av applikasjoner som forsøkes publisert til App Store blir derfor aldri godkjent.

Vi hadde i dette prosjektet ikke budsjettet med kostnader for publisering mot App Store, og selv om applikasjonen er klarert for opplastning er denne prosessen ikke gjennomført da oppdragsgiver ennå ikke har ytret ønske om dette.

7.2 Google Play

Google Play erstattet det opprinnelige Android Market i midten av mars 2012, og er markedet for applikasjoner til Android-baserte enheter. I motsetning til Apple er bygging utvikling for Android på en fri og åpen plattform, noe som gjør at stort sett hvem som helst med interesse for faget enkelt kan både utvikle og publisere sine applikasjoner for omverdenen.

Applikasjoner for Android kan utvikles på både Mac og Windows-baserte datamaskiner siden programvaren Eclipse er tilgjengelig for begge operativsystemer. Det stilles ingen krav til egen konto for utvikling, og applikasjoner kan enkelt overføres til egen smarttelefon eller nettbrett via Eclipse. Om det ønskes å publisere applikasjonen på Google Play påløper likevel en årsavgift på ca 150 norske kroner. Etter at denne avgiften er betalt får man full tilgang til å publisere applikasjoner for både gratis- og betalbare nedlastninger.

Utviklergruppen hadde fra tidligere Google Play konto og Diaset er publisert for gratis nedlastning i alle land for brukere med Android, selv om applikasjonen pr i dag kun er på norsk. Bildet nedenfor viser applikasjonen Diaset publisert på markedet for Android enheter, Google Play.

Diaset
mcurrang

★★★★★ (6)
INSTALLERT

Denne appen er kompatibel med din NetCom Samsung GT-I9000.

Mer fra utvikler

Stupido
MKURRANG
★★★★★ (3)
Gratis

Les mer >

Beskrivelse

Diaset er en mobilapplikasjon utviklet for barn- og unge med diabetes, og har som hensikt å effektivisere prosessen med beregning av insulindoser som diabetikere må injisere etter et måltid. I dag blir nemlig mange av disse utregningene gjort manuelt for hånd, og applikasjonen Diaset har derfor som mål å gjøre hverdagen til diabetikere litt enklere. Applikasjonen benytter seg av en lokal database som inneholder forskjellige typer matvarer og tilhørende verdier i karbohydrater. Databasen som brukes er matvaretabellen, hentet fra www.matvaretabellen.no og er utgitt av Mattilsynet, Helsedirektoratet og Universitetet i Oslo.

Besøk utviklerens nettsted > Utviklers e-postadresse >

Skjermbilder fra programmene

Antall karbohydrater = 45,19
Fast inn blodsukkerverdi:
Samlet korreksjonsdose:
Din aktivitet i %
Bekreft

MATVARER

- MELK OG MELKEPRODUKTER
- EGG
- FJØRFE OG KJØTT
- FISK OG SKALLDYR
- KORNVARER
- FRUKT OG GRØNT
- SUKKERPRODUKTER
- MARGARIN OG OLJE

OM DENNE APPEN

VURDERING: ★★★★★ (6)

OPPDATERT: 15. mai 2012

GJELDENE VERSJON: 1.0

KREVER ANDROID: 2.2 og nyere

KATEGORI: Sunnhet og trening

INSTALLASJONER 10 - 50

STØRRELSE: 1,3M

PRIS: Gratis

Figur 23 Applikasjonen på Google Play

Kapittel 8 Avslutning

8.1 Resultat

Fra begynnelsen av prosjektet har hele hensikten med applikasjonen handlet om å gjøre hverdagen for personer med diabetes lettere, spesielt rettet mot barn. Utviklingsgruppen og oppdragsgiver ønsket hele tiden å lage en mobilapplikasjon som er så enkel at alle kan bruke den, samt at applikasjonen skulle være avansert nok til å gjøre trygge beregninger for brukeren. Resultatet vi sitter igjen med etter prosjektet er det oppdragsgiver har ønsket seg fra start. Alle krav fra starten av prosjektet har blitt oppfylt samt send SMS funksjonen som var veldig viktig for oppdragsgiver. Resultatmessig ble prosjektmålet nådd, selv om det alltid vil være forbedringer som kan gjøres.

På nåværende tidspunkt har applikasjonen blitt sluppet til Google Play, men ikke til App Store. Grunnen til dette er som nevnt tidligere at App Store krever en betalbar utviklingskonto som utviklingsgruppen ikke innehar per i dag. Vi er veldig fornøyd med prototypversjonen som er utviklet og at applikasjonen er god nok til å bli sluppet på ett av de to store markedene. Utviklingsgruppen planlegger å se an utviklingen av antall nedlastninger og vurderinger på Google Play før det vurderes om videreutvikling er et tema. En eventuell avgjørelse angående videreutvikling vil bli tatt med oppdragsgiver.

8.2 Problemer og valg underveis

8.2.1 Problemer

I starten av prosjektet var utfordringen å få til en fungerende lokal database. Dette bød på en del problemer, siden vi hadde tre tabeller og ikke var helt sikre på hva som var den beste løsningen med å gå fra en tabell til neste. Noe vi prøvde var å lage en tabell med hovedkategori i ett html-dokument, og deretter sende med aktuell ID til neste tabell for å finne tilhørende underkategori. Denne løsningen kunne vi ikke benytte oss av, siden Phonegap ikke klarte å håndtere referanseoverføring gjennom HTML sider i det nyeste Android operativsystemet, 4.0.3. Dette var noe vi ikke fikk gjort noe med, siden problemet lå i Phonegap (37). Vi valgte da å se etter andre løsninger for databasen.

Et annet problem som oppsto med denne databaseløsningen var at det ikke kunne legges inn mer enn 512 linjer i databasen, på grunn av at cache minnet satt av til Phonegap databasen ble fullt. Utviklingsgruppen gikk da videre med arbeidet og endte opp med den databaseløsningen vi har i dag.

8.2.2 Valg

Gjennom prosjektet har utviklingsgruppen gjort spesielt et valg underveis som vi i ettertid innser kunne vært gjort annerledes.

Utviklingsgruppen valgte i starten av prosjektet å scripte de fleste av funksjonene på norsk. Det viser seg i ettertid at det beste hadde vært å kode og kommentere på engelsk, siden all koding på nett og i store bedrifter foregår på engelsk. Vi angrer på at dette ikke ble gjort fra starten, da dette vil være en omfattende prosess å endre nå i ettertid.

8.3 Videreutvikling

Under utviklingen av applikasjonen så vi hele tiden at det var flere nye funksjoner vi kunne legge til, men som ikke ble gjort på grunn av tidspresset med oppgaven. I de følgende avsnittene kan du lese om noen av de funksjonene vi så for oss.

To av de funksjonene vi fort skjønte vi ville implementert er en *“vis favoritter i måltid”* og en søkefunksjon i matvarelisten. *“vis favoritter i måltid”* funksjonen ville vist hvilke matvarer brukeren hadde valgt i hvert sitt favorittmåltid. I funksjonen skulle bruker kunne legge til, endre eller fjerne enkelte matvarer fra hvert enkelt måltid. Mens en søkefunksjon i matvarelisten ville gjøre det lettere for brukeren og navigere hvis han/hun kun skulle ha en enkel matvare og ikke flere.

En annen ting vi ser for oss som en mulig videreutvikling er at det legges til en funksjon som gjør applikasjonen flerspråklig slik at den vil nå ut til ennå flere brukere. Pr i dag er den primært laget for norske brukere, som hindrer en stor brukermasse.

Vi forsøkte å sende SMS direkte fra applikasjonen, men dette viste seg å være vanskelig å gjennomføre med tanke på tidspresset ved prosjektet. Derfor gikk vi for en enkel løsning som sendte bruker videre til SMS vinduet med en ferdigskrevet SMS. Beklageligvis fungerer ikke dette på iPhone, og dette vil derfor være en videreutvikling som er mer kritisk for applikasjonen.

Selv om applikasjonen er godt utviklet allerede, vil applikasjoner alltid ha et stort forbedringspotensial på grunn av teknologiens raske utvikling. Vi håper de ønskelige funksjonene en dag kan bli laget, spesielt hvis det viser seg at applikasjonen blir tatt i bruk og faktisk hjelper personer med diabetes til å få en lettere hverdag.

8.4 Kritikk av oppgaven

I utgangspunktet har vi ikke så mye å utsette eller kritisere oppgaven for. Oppgaven var definert på en godt forklart måte, som gjorde det slik at vi ikke følte det var mange spørsmål å stille ved den.

Det eneste vi føler kan sees på som kritikk av oppgaven er oppdragsgivers ønske om å ikke inkludere leger, sykepleiere osv. under utviklingen av applikasjonen. Grunnen til oppdragsgivers ønske om dette, var at det kunne bli vanskelig å realisere applikasjonen som følge av et mulig negativt søkelys mot den. Ut ifra oppdragsgivers ønske kan vi ikke verifisere at beregningene i applikasjonen er i henhold til norske regler og standarder. Noe som igjen vil påvirke brukerne av applikasjonen i negativ forstand.

8.5 Gruppeevaluering

Ved prosjektstart hadde ingen av de involverte i utviklingsgruppen veldig gode kunnskaper om kodespråkene og læringskurven har vært bratt igjennom hele prosjektet. Vi har heller ikke tidligere hatt så stort ansvar selv for hvordan vi ønsker å styre prosjekters retning som med dette prosjektet. Denne gangen måtte vi ta hensyn til oppdragsgivers ønsker, tidspress og skolens forventninger til hver enkelt. Det har vært en utfordring selv om det å jobbe med prosjekt som arbeidsform har vært utrolig lærerikt for oss alle. Vi er helt sikre på at arbeidet med prosjektet har gitt oss mye erfaring vi kan ta med oss videre i studie- og jobbsammenheng.

Det er stor enighet i utviklingsgruppen at arbeidet med prosjektet har vært spennende, utfordrende og veldig lærerikt. Vi føler alle at vi klarte å gjennomføre det vi bestemte oss for ved prosjekt start og er meget fornøyd med det produktet vi leverte. I punkt 8.5.2 har hver av oss skrevet et lite refleksjonsnotat om arbeidet.

8.5.1 Arbeidsfordeling og organisering

Fordelingen av ansvar mellom utviklingsgruppens medlemmer har vært i henhold til forprosjekt og nevnt i kapittel 1, ref. kapittel 1.7.1. Ved å følge den bestemte ansvarsfordelingen har vi hatt en relativt jevnt fordelt arbeidsmengde igjennom hele prosjektet.

Under utviklingsfasen fordelte vi oppgavene i henhold til ansvarsfordelingen over de tre områdene HTML/CSS, database og JavaScript. Dette gjorde at Michael Urrang primært konsentrerte seg om kodingen av HTML og design. Grunnen til at Michael fikk dette som hovedoppgave er hans bakgrunn og erfaring fra tidligere oppgaver knyttet til medieteknologi studie og hans rolle i utviklingsgruppen.

Trond Hagehaugen fattet tidlig interesse for databasen og det ble da naturlig at han fikk ansvaret for implementeringen av denne. Gudbrand Olimb jobbet som prosjektleder med det overordnede ansvaret for utviklingsgruppen, hans primære arbeidsoppgaver var JavaScript og applikasjonens funksjonalitet. Denne arbeidsfordelingen viste seg i ettertid å være svært effektiv, en viktig faktor for prosjektets resultat.

Rollefordelingen som ble laget i prosjektplanen har også vist seg å være en viktig og enkel faktor som har gjort det enkelt for hele utviklingsgruppen å holde kontroll over sine oppgaver. Poenget med rollefordelingen har jo i samsvar med ansvarsfordeling vært å fordele oppgaver i gruppen. Å ha en prosjektleder var et naturlig valg når det jobbes med et prosjekt. Selv om vi ved prosjektstart bestemte en fast prosjektleder, har alle fått forsøkt seg på oppgavene prosjektleder innehadde. Men det var Gudbrand som offisiell prosjektleder, som kunne ta den endelige avgjørelsen ved splittede meninger.

Å ha en bestemt kontaktperson viste seg å være et av de smarteste valgene vi fortok oss. Trond som var kontaktpersonen har hele tiden hatt løpende kontakt med både veileder og oppdragsgiver. Dette gjorde at gruppen enkelt kunne avtale møter, stille spørsmål og orientere om endringer gjort underveis. Oppdragsgiver og veileder var også meget fornøyd med å kun ha en kontaktperson å forholde seg til.

I bunn og grunn er vi veldig fornøyd med hvordan organiseringen og ansvarsfordelingen av utviklingsgruppen ble.

8.5.2 Individuelle refleksjoner og tanker

Gudbrand Olimb

Jeg er veldig fornøyd med utviklingen av Diaset applikasjonen. Hvordan gruppen har arbeidet sammen i løpet av hele perioden har vært fenomenal. Ut ifra tidligere erfaringer har jeg merket at det blir litt gnisninger innad i prosjektgrupper over lang tid, men det hendte ikke med denne utviklingsgruppen. Alle på gruppen har bidratt til et godt arbeidsmiljø, prosjektets utvikling, og virkelig tatt i et tak når vi trengte det. Som prosjektleder føler jeg at jeg ikke selv kunne vært på en bedre gruppe, alle arbeidsoppgaver har egentlig fordelt seg selv naturlig.

Det å utvikle en applikasjon både til Android og iPhone har vært utrolig inspirerende og lærerikt. Mine tidligere erfaringer med applikasjon utvikling til smarttelefoner er sterkt fraværende og jeg har aldri brukt programmeringsspråkene JavaScript, HTML og CSS veldig mye. Jeg føler vel mest selv at JavaScript er det språket jeg har hatt mest utvikling med, samtidig som det har vært det språket jeg har trivdes best med. Etter arbeidet med prosjektet ser jeg mye lysere på programmering, spesielt når det er rettet til produkter hvor du ser endringer i det du gjør.

Det jeg vil fremheve nå ved prosjektets slutt er spesielt hvordan designet ble og det at vi faktisk klarte å få publisert applikasjonen på Google Play. Det at vi ikke skuffet oppdragsgiver ved å levere et halvferdig produkt gjorde egentlig at vi klarte det målet vi hadde satt oss. Etter å ha hatt samtaler med oppdragsgiver skjønner jeg det også slik på han, at vi har levert over forventning. Dette er en utrolig god følelse og sitte igjen med etter prosjektet. Til slutt vil jeg takke veileder, oppdragsgiver og utviklingsgruppen, Trond Hagehaugen og Michael Urrang for arbeidet med Diaset.

Trond Hagehaugen

Jeg er veldig fornøyd med sluttresultatet av applikasjonen vi har laget. Jeg har tidligere hatt gode erfaringer med å jobbe sammen Gudbrand, og vi har samarbeidet meget bra sammen tidligere. Jeg har ikke jobbet med Michael i noen tidligere prosjekter ved HiG fra før, men vi ble raskt en sammensveiset gruppe som jobbet mot samme mål i dette prosjektet. Det jeg er mest fornøyd med gruppemessig, er at vi har hjulpet hverandre underveis i prosjektet, og hele tiden hatt en positiv innstilling til arbeidet underveis.

Bachelorprosjektet har vært en lærerik prosess hvor jeg har lært mye nytt, og synes det har vært spennende å utvikle en mobilapplikasjon for første gang. Det har vært en utfordring å estimere tiden vi kom til å bruke på denne oppgaven, spesielt siden verken jeg eller noen på gruppa hadde mye erfaring i JavaScript, HTML 5 og CSS3 fra før. Med noen problemer underveis så gikk tiden veldig fort, men vi klarte å komme i mål, noe jeg er veldig fornøyd med.

Selve rapporten er jeg også fornøyd med. Alle har bidratt i like stor grad, slik at det er ingen som har fått som oppgave å skrive den alene. Bacheloroppgaven handler som sagt å lære seg nye emneområder, og det var viktig for oss at alle bidro på alle områder ved oppgaven. At vi alle har bidratt i like stor grad, har vært en suksessfaktor for å realisere prosjektet.

Jeg vil takke Gudbrand, Michael, veileder og oppdragsgiver for å ha lært mye nytt i en spennende prosess!

Michael Kornbrekk Urrang

Diaset som bacheloroppgave har for meg vært et spennende og ikke minst lærerikt prosjekt. Det som først og fremst føles naturlig å trekke frem er hvor godt utviklingsgruppen har fungert sammen gjennom hele prosjektet. Siden oppstarten hvor vi ble enige om prosjektleder og ansvarsområder har oppgaver blitt jevnt fordelt, og jeg sitter igjen med et inntrykk av at samtlige har bidratt til at prosjektet oppnådde det sluttprodukt som er tilfelle.

Som medieteknologistudent hadde jeg før prosjektstart erfaring med utvikling av applikasjoner til Android-baserte smarttelefoner. Selv om tidligere prosjekter var skrevet med Java som native applikasjoner hadde jeg likevel et godt utgangspunkt når vi i dette prosjektet skulle utvikle en webapplikasjon med bakgrunn i mine erfaringer for programmet Eclipse. Jeg var ellers den eneste deltakeren med Mac i vår utviklingsgruppe og har derfor gjennom prosjektet gjort meg godt kjent med utviklingsverktøyet Xcode for applikasjoner til Apples produkter.

Når prosjektet nå nærmer seg slutten kjennes det litt vemodig å skulle avslutte en oppgave som har gitt meg utrolig mye med tanke på både utfordringer og erfaringer. Jeg snakker nok på hele gruppens vegne når jeg sier at jeg er svært godt fornøyd med det endelige resultatet av applikasjonen. Tilbakemeldinger vi har fått fra både medstudenter, veileder og oppdragsgiver har utelukkende vært positive, og jeg har god tro på at applikasjonen kan være til hjelp for både unge- og eldre diabetikere.

Til slutt vil jeg i tillegg til å takke mine to dyktige gruppemedlemmer Gudbrand og Trond, også takke veileder og oppdragsgiver for et spennende og lærerikt prosjekt.

8.6 Konklusjon

I begynnelsen av prosjektet var vi alle litt usikre på hvordan det skulle gå med applikasjonen siden ingen av oss fra tidligere hadde mye erfaring med mobilapplikasjoner. Vi var usikre på om oppgaven ville være omfattende nok som bacheloroppgave og om oppdragsgivers og skolens forventninger ville bli møtt. Selv om oppdragsgiver hadde sine spesielle ønsker for applikasjonen følte vi at applikasjonen til dels kunne styres i den retningen vi selv ønsket. Det gjorde igjen at vi selv føler oppgaven har blitt svært omfattende og ut fra samtaler med veileder har vi forstått at skolens forventninger er besvart. Som nevnt tidligere i rapporten har også oppdragsgiver uttrykk stor glede for applikasjonen og mener prosjektet har levert over hans forventning.

Diaset viste seg til tider å være et vanskelig prosjekt i form av design, programmering og beregninger. Spesielt var det til tider vanskelig å utvikle applikasjonen siden den var ment for to forskjellige plattformer. Vi har lært at det er en stor forskjell på å utvikle en applikasjon til mobil i forhold til det å utvikle en applikasjon til datamaskiner. Begrensningene er flere og hindringene noen ganger større. Når vi utviklet til to forskjellige plattformer var vi hele tiden nødt til å sjekke at funksjoner og operasjoner vi gjorde, fungerte på begge plattformene. Den kontinuerlige testingen igjennom prosjektets levetid viste seg å være klok, siden det flere ganger var små justeringer som måtte gjøres for å få inn funksjonaliteten til å fungere på begge plattformene. Arbeidet med prosjektet lært oss hvordan man lager, implementerer og realiserer en mobilapplikasjon.

Kort oppsummert har tiden med skriving og utvikling av Diaset vært veldig lærerik og spennende. Prosjektet har lært oss mye og det har vært en spennende slutt på bachelor utdannelsen ved Høgskolen i Gjøvik. Vi vil rette en siste spesiell takk til oppdragsgiver, veileder og Øyvind Kolloen som har vært store bidragsytere igjennom prosjektet.

Kapittel 9 Litteraturliste

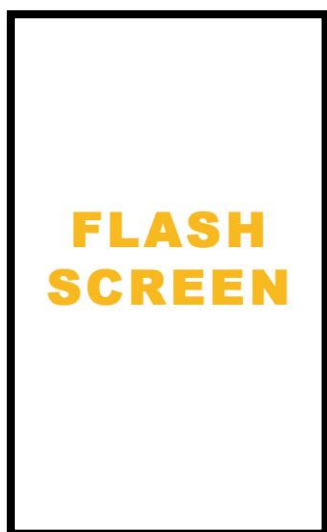
1. Diabetesforbundet. Autoimmune sykdommer. 2010 [Sitert 02.05.2012]; URL: <http://www.diabetes.no/Autoimmune+sykdommer.9UFRHWYI.ips>.
2. Wikipedia. Insulin. Wikipedia; 2012 [Sitert 02.02.2012]; URL: <http://no.wikipedia.org/wiki/Insulin#Insulindose-enheten>.
3. Raassum HÅMEG, Bruvoll Ø. Diaset - mobil insulindosekalkulator. Gjøvik: Høgskolen i Gjøvik, 2011.
4. Wikipedia. Web Standards. [www.wikipedia.org](http://en.wikipedia.org/wiki/Web_standards) 2012 [Sitert 04.04.2012]; URL: http://en.wikipedia.org/wiki/Web_standards.
5. Wikipedia. JavaScript. 2012 [Sitert 10.04.2012]; URL: <http://en.wikipedia.org/wiki/JavaScript>.
6. Phonegap. How Phonegap Works. 2012 [Sitert 14.02.2012]; URL: <http://phonegap.com/about>.
7. W3Schools. HTML 5 Tutorial. 2012 [Sitert 19.02.2012]; URL: <http://www.w3schools.com/html5/default.asp>.
8. W3Schools. CSS3 Tutorial. 2012 [Sitert 19.02.2012]; URL: <http://www.w3schools.com/css3/default.asp>.
9. jQuery. jQuery: The Write Less, Do More, JavaScript Library. 2012 [Sitert 25.02.2012]; URL: <http://jquery.com/>.
10. Hansen KT, Mallaug T. Databaser. Oslo: Gyldendal Norsk Forlag; 2008.
11. Apple. Xcode 4. 2012 [Sitert 28.01.2012]; URL: <https://developer.apple.com/xcode/>.
12. Eclipse Foundation. Eclipse - The Eclipse Foundation open source community website. 2012 [Sitert 28.01.2012]; URL: <http://www.eclipse.org/>.
13. ADL Mobile. Native Apps. 2010 [Sitert 20.04.2012]; URL: <http://adlmobile.wikispaces.com/Native+Apps>.
14. ADL Mobile. Mobile Web Apps. 2011 [Sitert 20.04.2012]; URL: <http://adlmobile.wikispaces.com/Native+Apps>.
15. Wikipedia. Website wireframe. 2012 [Sitert 15.04.2012]; URL: http://en.wikipedia.org/wiki/Website_wireframe.
16. Wikipedia. Comma-separated values. 2012 [Sitert 03.03.2012]; URL: http://en.wikipedia.org/wiki/Comma-separated_values.
17. Wikipedia. Nettskyen. 2012 [Sitert 20.04.2012]; URL: <http://no.wikipedia.org/wiki/Nettskyen>.
18. Wikipedia. Waterfal model. 2012 [Sitert 04.02.2012]; URL: http://en.wikipedia.org/wiki/Waterfall_model.
19. Wikipedia. Incremental build model. 2012 [Sitert 04.02.2012]; URL: http://en.wikipedia.org/wiki/Incremental_build_model.
20. Mountain Goat Software. Introduction to Scrum. 2012 [Sitert 04.02.2012]; URL: <http://www.mountaingoatsoftware.com/topics/scrum>.
21. Wikipedia. Scrum. In: process.png S, editor. 2012.
22. Appcellerator. Platform Overview. 2012 [Sitert 23.01.2012]; URL: <http://www.appcelerator.com/platform>.
23. Rhomobile. Cross-Platform Mobile App Development. 2012 [Sitert 23.01.2012]; URL: <http://www.rhomobile.com>.

24. MoSync. Create iPhone and Android apps with JavaScript and C++. 2012 [Sitert 23.01.2012]; URL: <http://www.mosync.com/>.
25. Appmobi. Appmobi.com - Create Mobile and Web Apps with HTML5 and JavaScript. 2012 [Sitert 28.01.2012]; URL: <http://www.appmobi.com/>.
26. Sencha. Sencha SDK tools. 2012 [Sitert 23.01.2012]; URL: <http://www.sencha.com/products/sdk-tools>.
27. Phonegap. PhoneGap. 2012 [Sitert 23.01.2012]; URL: <http://phonegap.com/>.
28. Saxegaard K, Grønnerud TL. Hvor mye insulin skal tas til maten du spiser?2011 [Sitert 19.3.2012]. URL: http://www.diabetes.no/filestore/-Diabetesno_-_filarkiv/For_tillitsvalgte/Presentasjoner/Diabeteskonferansen_/karbohydratvurdering.pdf.
29. Mattilsynet, Helsedirektoratet, Oslo Ui. Matvaretabell. 2012; URL: www.matvaretabellen.no.
30. Wikipedia. Typografi. 2012 [Sitert 22.03.12]; URL: <http://no.wikipedia.org/wiki/Typografi>.
31. Straydog. straydog selected as responsive web design partner for paladin security. In: responsive-web-design-vancouver-paladin-security.png, editor. 2012.
32. Group WCW. Web SQL database. 2010 [Sitert 10.02.2012]; URL: <http://www.w3.org/TR/webdatabase/>.
33. Mattilsynet, Helsedirektoratet,UiO. Copyright. 2012; URL: <http://www.matvaretabellen.no/#copyright>.
34. Schmitt C, Simpson K. HTML5 Cookbook. Sebastopol, California USA: O'Reilly; 2012.
35. Flanagan D. JavaScript The Definitive Guide. Sebastopol, California, USA: O'Reilly; 2011.
36. Zakas NC. JavaScript for Web Developers 2nd Edition. Indianapolis, Indiana, USA: Wiley Publishing; 2009.
37. Flere forfattere. WebView - URL mechanism is broken - passing parameters does not work. Google.com; 2011; URL: <http://code.google.com/p/android/issues/detail?id=17535>.

Kapittel 10 Vedlegg

10.1 Wireframes

Wireframes Diaset



Figur 1 Oppstartsskjerm



Figur 2 index.html



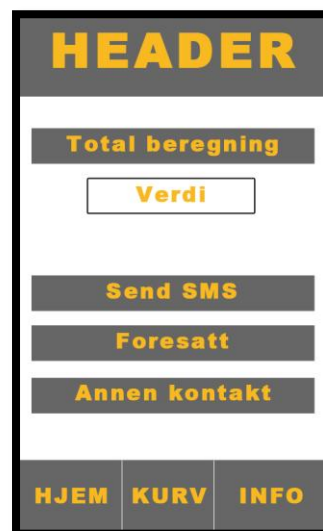
Figur 3 matvarer.html



Figur 4 kurv.html



Figur 5 beregning.html



Figur 6 bekreft.html



Figur 7 innstillinger.html



Figur 8 personalia.html

10.2 C++ program, modifisering av SQL fil.

```

/* Forfattere: Gudbrand Olimb, Trond Hagehaugen, Michael Urrang
Bachelor 2012 - Diaset
Manipulering av sql script
Programmet leser inn autogenerated sql-script av tabellene matvarer og underkategori
Programmet skriver så ut javascriptet for generering av databasen
i applikasjonen
Alle linjene kan da bare kopieres rett inn i databasen
*/

#include <iostream> // Inkluderer filer
#include <fstream>
#include <cstring>

using namespace std;

class Sql { // Klasse for innlesing av linjer
private:
    char navn[150]; // Variabel for én linje i fila
public:
    Sql(); // Constructor
    void skriv_til_fil(ostream* ut); // Skriv og les fra fil
    void les_fra_fil(char t[], istream* inn);
};

// Funksjoner

void skriv_til_fil();
void les_fra_fil();
Sql sql[1200]; // Lager objekter for matvarer
// Sql sql [60] // Lager objekter for underkategori
int siste_sql; // Tellevariabel

int main() { // Leser og skriver fil
    les_fra_fil();
    skriv_til_fil();
    cout <<"Leser inn sql-script fra fil og skriver ut javascript!"<<'\n';

    system("pause");
    return 0;
}

Sql::Sql() { // Nullstiller
    strcpy(navn, "");
}

void Sql::skriv_til_fil(ostream* ut) { // Legger til matvarene i javascript
    *ut << "transaction.executeSql(";
    *ut << '\'';
    *ut << "INSERT INTO mat(M_id,UN_id,navn,kalorier) VALUES ";
    *ut << navn;

    *ut <<";\')";
    *ut <<'\n';
}

```

```

void Sql::les_fra_fil(char t[], istream* inn) { // Leser linje fra fil
    strcpy(navn, t);
}

void skriv_til_fil() { // Skriver matvarer til fil
    ofstream utfil("matvarer.dta");

    for (int i = 1; i <= siste_sql; i++)
        sql[i].skriv_til_fil(&utfil);
}

void les_fra_fil() { // Leser matvarer fra fil
    char buffer[150];
    ifstream innfil("mat.dta");
    siste_sql= 0;

    if (innfil) {
        innfil.getline(buffer, 150);
        while (!innfil.eof() && siste_sql < 1200) {
            sql[++siste_sql].les_fra_fil(buffer, &innfil);
            innfil.getline(buffer, 150);
        }
    } else
        cout << "\nFant ikke filen 'mat.dta'!\n\n";
}

/* Dette er de samme funksjonene som modifierer underkategoriene

void Sql::skriv_til_fil(ostream* ut) { // Legger til underkategoriene i javascript
    *ut << "transaction.executeSql(";
    *ut << '\';
    *ut << "INSERT INTO underkategori(U_id,HO_id,navn) VALUES ";
    *ut << navn;
    *ut <<";\';";
    *ut <<'\n';
}

void skriv_til_fil() { // Skriver underkategori til fil
    ofstream utfil("underkategori.dta");

    for (int i = 1; i <= siste_sql; i++)
        sql[i].skriv_til_fil(&utfil);
}

```

```
void les_fra_fil() { // Leser underkategori fra fil
    char buffer[150];
    ifstream innfil("under.dta");
    siste_sql= 0;

    if (innfil) {
        innfil.getline(buffer, 150);
        while (!innfil.eof() && siste_sql < 1200) {
            sql[++siste_sql].les_fra_fil(buffer, &innfil);
            innfil.getline(buffer, 150);
        }
    } else
        cout << "\nFant ikke filen 'mat.dta'!\n\n";
}

*/
```

10.3 Epost mattilsynet

For personopplysningsgrunner er dette vedlegget redigert

Epost sendt til mattilsynet:

Fra: Trond Hagehaugen
Til: Matportalen
Emne: Matportalen - Kontakt oss: Spørsmål ang excel format

Emne:
Spørsmål ang excel format

Henvendelsen gjelder:
Matvaretabellen

Beskjed:

Hei! Undertegnede jobber med et bachelorprosjekt der vi bruker matvaretabellen som database.

Vi bruker opprinnelig excel arket fra 2006, men skulle gjerne oppdatert til 2012 verdier.

Problemet er at excelarket som ligger på nett ikke er oppbygd på den samme måten som excel arket fra 2006.

Har dere 2012 versjonen i samme "format" som 2006, altså at dere har samme oversikten, feks 1 - melkeprodukter, 1.1 melk og melkebasert drikke, 1.2 yoghurt ... osv, med tilhørende navn og verdier ?

Ditt for-/etternavn:
Trond Hagehaugen

Epost svart fra mattilsynet:

Fra: Ellen Kielland [mailto:Ellen.Kielland@mattilsynet.no]
Til: Trond Hagehaugen
Kopi: Matportalen
Emne: SV: Matportalen - Kontakt oss: Spørsmål ang excel format

Hei,

Dessverre er ikke xl-arket for Matvaretabellen 2012 bygd opp på samme måte som 2006 versjonen.

Vi har fått flere henvendelser om dette og det er mulig at vi utvider 2012 versjonen, men jeg kan dessverre ikke si når dette blir.

Vennlig hilsen

Ellen Kielland
Rådgiver/ernæring
Mattilsynet, hovedkontoret, seksjon omsetning til forbruker

10.4 Brukertestester

Brukertest Diaset – insulindosekalkulator

Navn: Ole Kristian Fløystad

Dato: 25/04 - 2012

Kommentar til funksjonalitet

- Skjøpper egentlig ikke så mye av listen, den er veldig rotete, ser egentlig bare hovedkategorier og kommer heller ikke videre.
- + Personalia funksjonen er enkel, liker at jeg kan fylle inn navnet på foresatt.
- + Informasjonssiden beskriver applikasjonen godt.
- Favoritter, din mat, innstillinger funksjonene fungerer ikke. Men dette har igjen sammenheng med listen skjønner jeg.

Kommentar til design

- + Hovedsiden på applikasjonen er fin, klare knapper med fine farger.
- + Fargene i matvarelisten er fine og enkle. Kunne vært litt mer fargevariasjon

Kort oppsummering

De funksjonene som fungerer er helt ok, designet er enkelt og brukervennlig på hovedsiden. Jeg tror barn kan fatte interesse av de spennende fargene i applikasjonen. Kom egentlig ikke så mye lenger en til hovedsiden og personalia på grunn av feilen med listen, dette drar selvfølgelig ned mye.

Vurdering fra 1 – 6: 2

Brukertest Diaset – insulindosekalkulator

Navn: Petter Gresmo

Dato: 01/05 - 2012

Kommentar til funksjonalitet

- Veldig enkel funksjonalitet
- Jeg trenger ikke trykke mange ganger før jeg kommer igjennom applikasjonen
- Liker at jeg kan lagre favoritter.

Kommentar til design

Din mat siden

- Hva er det brukeren fyller inn? Er det liter? Gram? DI?

Fargene i applikasjonen burde være mulig å endre.

Den lilla fargen i applikasjonen kunne vært litt mer spenstig.

Kort oppsummering

En veldig enkel applikasjon, det kan gjøres litt med designet. Men alt i alt er det en veldig god app til å være et studentprosjekt.

Vurdering fra 1 – 6: 5

Brukertest Diaset – insulindosekalkulator

Navn: Kjersti Brennbakken

Dato: 08/05 - 2012

Kommentar til funksjonalitet

MATVARER listen

- **Stekt egg - 20** HVA ER 20 ?
Er dette overhode relevant for brukeren?

Din mat siden

- På toppen burde det stå: Hva har du spist?
- Hva er det brukeren fyller inn?

Er det gram/liter/dl
- På spørsmål om lagre favoritt:
 - o Burde bli spurt om: Vil du lagre matvarene som et favorittmåltid?
 - o Fjerne det som kommer i skrive feltet
- Det burde kreves at man fyller ut personalia før bruk

Beregning siden

- **Din aktivitet:** Er det %? Eller hva er mengden man slider?

Kommentar til design

- + Designet er oversiktlig og rent.
- + Fargekombinasjonene gjør applikasjonen "lett å lese" og enkel å bruke

Kort oppsummering

Jeg synes applikasjonen er veldig ren og enkel. Veldig enkel å forstå selv om det er noen ting som kan være tydeligere, ref. Tidligere kommentarer. Det burde ikke være vanskelig for brukeren å forstå applikasjonen, hvis endringene i kommentarer blir utført.

Vurdering fra 1 – 6: 5

10.5 Møtereferater

Møtereferat med Harald Liodden

Tid og sted: 25/1 2012 kl 09:00, Høgskolen i Gjøvik, Harald Lioddens kontor a116 d.

Tema: Første møtet, litt prat rundt prosjektets gang.

Referat:

Gannt-skjemaet som skal lages: Bør ha statusrapporter gjennom hele prosessen, 3 stk. Bør passe på at det passer med utviklingsmodell.

- Timelogg lages individuelt for gruppa
- "Hjelp" til database dersom det trengs av veileder.

Tidspunkt for neste møte: Tirsdag 7. februar kl 09:00. Tidspunkt for videre møter avtales etter hvert møte.

Møtereferat med Harald Liodden

Tid og sted: 07/2 2012 kl 09:00, Høgskolen i Gjøvik, Harald Lioddens kontor a116 d.

Tema: Nettside, utviklingsmiljø, appmobi

Referat:

- Tok en kjapp titt på nettsiden,
- Snakker litt om valg av utviklingsmiljø hvor appmobi er et stort tema.
- Snakker om å forsøke å gå for appmobi som utviklingsmiljø i første omgang.

Tidspunkt for neste møte: 22. 02 kl 09:00.

Møtereferat med Harald Liodden

Tid og sted: 22/2 2012 kl 09:00, Høgskolen i Gjøvik, Harald Lioddens kontor a116 d.

Tema: Diverse

Referat:

- Ser på status på nettside, hvordan den ser ut, snakker om mulige forbedringer.
- Snakker litt om hvordan man skal planlegge fremover, hvordan man skal strukturere arbeidet.
- Ser litt på ulike typer databasemodeller som kan velges og som vil passe bra til prosjektet.

Tidspunkt for neste møte: Onsdag 7. mars kl 09:00. Tidspunkt for videre møter avtales etter hvert møte.

Møtereferat med Harald Liodden

Tid og sted: 07/3 2012 kl 09:00, Høgskolen i Gjøvik, Harald Lioddens kontor a116 d.

Tema: Gjennomgang av statusrapport, databasemodell,

Referat:

- Diskuterte databasemodellen som ble sendt inn forrige uke.
- Endringer vi må gjøre tilknyttet databasemodellen
 - o Feil relasjoner
 - o Må normaliseres
 - o Må ha flere tabeller på grunn av relasjonene.
 - o koblingstabell
- Statusrapport var helt grei den.

Tidspunkt for neste møte: 13. 03 kl 09:00.

Møtereferat med Harald Liodden

Tid og sted: 13/3 2012 kl 09:00, Høgskolen i Gjøvik, Harald Lioddens kontor a116 d.

Tema: Parametersøk i database

Referat:

- Fikk svar på hvordan man kan sende med parametere for bruk i søk i neste funksjon i databasen.
- Diskuterte litt rundt fremgangen i prosjektet.
- Snakket om hvordan det går med rapportskrivning samtidig som man arbeider med applikasjonen.

Tidspunkt for neste møte: Tirsdag 27. Mars kl 09:00. Tidspunkt for videre møter avtales etter hvert møte.

Møtereferat med Harald Liodden

Tid og sted: 27/03 2012 kl 09:00, Høgskolen i Gjøvik, Harald Lioddens kontor a116 d.

Tema: Problemer med opprettelse av database

Referat:

- Fant ut at det ikke ligger støtte for foreign key i phonegap.
- Problemer med relasjoner når man ikke kan bruke foregin key.
- Ser på muligheter for å komme rundt det.
- Diskuterer kort hvordan det går med skrivningen av prosjektet.

Tidspunkt for neste møte: 11.04 kl 09:00.

Møtereferat med Harald Liodden

Tid og sted: 11/04 2012 kl 09:00, Høgskolen i Gjøvik, Harald Lioddens kontor a116 d.

Tema: diverse

Referat:

- Ble utsatt til 18.04 fordi det ikke var behov for statusmøte

Tidspunkt for neste møte: 18.04 kl 10:00.

Møtereferat med Harald Liodden

Tid og sted: 18/4 2012 kl 10:00, Høgskolen i Gjøvik, Harald Lioddens kontor a116 d.

Tema: Hvordan ligger vi prosjektet an.

Referat:

- Database problemer
- Tok ett raskt møte med Øyvind Kolloen angående løsning av problemer tilknyttet databasen.
- Gruppen må jobbe hardt fremover

Tidspunkt for neste møte: Onsdag 2. mai kl 10:00. Tidspunkt for videre møter avtales etter hvert møte.

Møtereferat med Harald Liodden

Tid og sted: 02/5 2012 kl 10:00, Høgskolen i Gjøvik, Harald Lioddens kontor a116 d.

Tema: Viste frem fremgangen av applikasjonen, hva som må gjøres

Referat:

- Fikk forslag om å gå til Øyvind kolloen for å finne ut hvordan vi kan få vist listene fra databasen riktig.
- Hvordan vi skal rekke å bli ferdig med resterende funksjonalitet.
- Snakket kort om designet og rapport skiving.
- **Det vil bli kort med tid på slutten.**

Tidspunkt for neste møte: Onsdag 16.mai kl 10:00. Siste møte før levering

Møtereferat med Harald Liodden

Tid og sted: 16/5 2012 kl 10:00, Høgskolen i Gjøvik, Harald Lioddens kontor a116 d.

Tema: Siste møte

Referat:

Kommentarer til rapporten:

- Mangler noe "flyt" i rapporten
- Må få til en rød tråd, er noe "tungt" språk flere steder
- Legge inn tap av referanseintegritet når det gjelder tap av foreign keys
- Logisk datamodell endres til konseptuelt klassediagram, hadde skrevet feil overskrift.

Spørsmål:

1. Lov å legge inn den nye prosjektplanen med endringer etter første innlevering?
2. Kodevedlegg ?
3. Hva er nivået på de som leser rapporten ?

Svar:

1. Ja!
2. Legg med koden i zip fila.
3. Bør være på samme nivå som oss.

Tidspunkt for neste møte: Dette var siste møte


10.6 Statusrapporter

27.02.2012



Statusrapport

Diaset



Gudbrand Olimb, Trond Hagehaugen,
Michael Urrang

Statusrapport

Diaset

Gruppen er nå godt i gang med hovedprosjektoppgaven, og har til nå hatt flere møter både med veileder og arbeidsgiver.

Ferdiggjort

Gruppen har gjennomført store deler av det planlagte arbeidet som skulle vært gjennomført til 27.02.2012.

- **Forarbeid**

Under forarbeidet til prosjektet fullførte vi prosjektplan etter råd fra veileder, leverte prosjektavtalen mellom oss, arbeidsgiver og høgskolen og forsøkte flere forskjellige utviklingsmiljøer. Under valget av utviklingsmiljø kom vi over noen avgjørelser som gikk bort i fra hva vi først antok vi ville gjøre, beskrevet i prosjektplanen. Dette valget har vist seg i ettertid å være et klokt valg.

- **Opprette webside og lage skall for applikasjonen for både Android og Apple**

HEADER

BAKGRUNNSBILDE

MATVARER

PERSONALIA

INNSTILLING

HJEM KURV INFO

Vi har opprettet en webside (www.urrang.com/diaset), hvor vi har oppdatert prosessen med å lage skallet til applikasjonen. Websiden ble litt krevende å lage da høgskolens systemer satte krav til hvordan vi måtte utforme hjemmesiden, dette endte med at vi videresender fra vårt domene på høgskolen videre til vår hjemmeside.

Etter å ha opprettet websiden, startet vi med utarbeidelsen av wireframen av hvordan vi så for oss applikasjonen vil se ut. Denne wireframen kan sees på vår hjemmeside. Vi er nå godt i gang i prosessen med å lage applikasjonen og er nå kommet til steget hvor vi jobber med databasen.

- **Opprette database og implementere**

Databasen bygger på en matvaretabell hentet fra matportalen.no og inneholder det meste av matprodukter og retter. Databasen er opprettet i SQL, og blir videre implementert til applikasjonen via javascript.

Planen videre

- **Ferdig implementering av databasen**

Da vi har noen mangler knyttet til implementeringen av databasen, er dette det første og viktigste steget videre i utviklingen av applikasjonen. Dette håper vi å få på plass snarest.

- **Funksjonalitet og kravspesifikasjon**


Vi skal fortsette på rapporten ved å fullføre kravspesifikasjonen. Funksjonaliteten blir startet på så fort som databasen er på plass.

27.03.12



Statusrapport

Diaset



Gudbrand Olimb, Trond Hagehaugen,
Michael Urrang

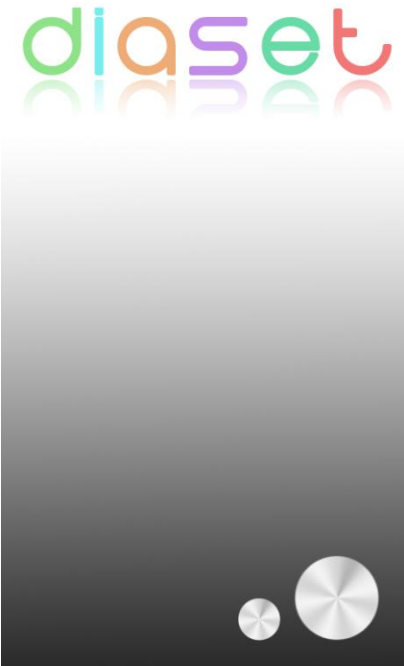
Statusrapport

Diaset

Gruppen er nå godt i gang med implementeringen av databasen, selv om vi har støtt på noen problemer knyttet til opprettelsen av databasen. Dette lager enkelte forsinkelser i prosjektet, og har gjort at vi må forskyve andre oppgaver.

Ferdiggjort

Gruppen har som sagt støtt på enkelte problemer knyttet til databasen, på grunn av dette har vi forsøkt å få gjort enkelte "enkle" oppgaver mens vi løser problemet. Vi har kommet godt i gang med rapport og design for å gjøre applikasjonen så enkel som mulig. Designet og grensesnittet mot barn fra 7års alderen er gjort i samarbeid med oppdragsgiver.



- **Rapport**

Vi er godt i gang med selve hovedrapporten og har startet på arbeidet med kravspesifikasjonen. Vi følte alle at vi ikke kunne la alle deler av prosjektet stoppe selv om vi møtte på problemer knyttet til databasen og implementeringen av den. Derfor flyttet vi en liten del av fokuset over på rapporten slik at vi kan jobbe med den mens vi samtidig løser problemene med databasen. Rapporten blir skrevet etter høgskolens standarder og det korrektur leses ofte, slik at rapporten følger de regler som er satt.

- **Holde fokus på oppdatering på webside**

Vi forsøker hele tiden å oppdatere jevnlig på vår webside (www.urrang.com/diaset), selv om det enkelte ganger kan gå litt tid mellom oppdateringene. Vi oppdaterer websiden de gangene vi føler vi har noen nytt å skrive om, men samtidig vil vi ikke dele all informasjon rundt prosjektet med "publikum".

- **Design applikasjon**

Vi begynner å få et veldig godt bilde på hvordan applikasjonen vil se ut når den er ferdig. Vi jobber hele tiden mot det å gjøre den så enkel som mulig ved å bruke farger som gjør den enkel, morsom og spennende å bruke. Sammen med oppdragsgiver er vi blitt veldig enige om hvordan applikasjonen skal se ut.

Planen videre

- **Ferdig implementering av databasen og rapport**

Da vi har noen problemer med implementeringen av databasen, er dette helt kritisk for vår videre i utviklingen av applikasjonen, og dette har vi nå et meget stort fokus på. Samtidig må vi fortsette med rapporten slik at vi ikke blir hengende etter på verken rapport eller applikasjon. Vi vil egentlig si at dette er en ganske kritisk fase i prosjektet, hvor vi må jobbe ekstra hard for å ikke få store oppgave mengder på et senere tidspunkt.

Statusrapport

30.04.12

Applikasjonen

Gruppen henger litt etter tidsskjemaet og er ikke helt ferdig med applikasjonen.

Det som mangler er noe funksjonalitet, men vi håper å få dette på plass så raskt som mulig, forhåpentligvis innen denne uka.

Rapporten

Rapporten er pr dags dato ca 50 sider, og av innhold som vi ikke har skrevet om så mangler i hovedsak bare analyse i etterkant av prosjektet.

10.7 Prosjektplan - Diaset, Gudbrand Olimb, Trond Hagehaugen, Michael Urrang

Prosjektplan

Bacheloroppgave 2012

Diaset - Mobil insulindosekalkulator



Gudbrand Olimb, Trond Hagehaugen, Michael Urrang

27.01.2012

Oppdatert 10.02.2012

Innhold

1. Mål og rammer	4
1.1 Bakgrunn	4
1.2. Prosjekt mål.....	5
1.2.1 Effektmål	5
1.2.2 Resultatmål.....	5
1.3. Rammer	6
1.4. Målgruppen	6
2. Omfang	7
2.1. Oppgavebeskrivelse.....	7
2.2. Avgrensning.....	8
3. Prosjektorganisering.....	9
3.1. Ansvarsforhold og roller	9
3.1.1. Utviklingsgruppen.....	9
3.1.2. Oppdragsgiver	10
3.1.3. Veileder	10
3.2. Rutiner og regler i utviklingsgruppen	10
3.2.1 Prosjektleder	10
3.2.2 Arbeidstid	11
4. Planlegging, oppfølging & rapportering	12
4.1. Hovedinndeling av prosjektet.....	12
4.1.1 Systemutviklingsmodell.....	12
4.1.2 Prosessrammeverk/utviklingsmiljø	13
4.2. Plan for statusmøter og beslutningspunkter.....	14
5. Organisering av kvalitetssikring.....	15

5.1. Dokumentasjon, standardbruk og kildekode	15
5.3. Risikoanalyse (identifisere, analysere, tiltak, oppfølging)	15
6. Plan for gjennomføring.....	18
6.1 Work Breakdown Structure.....	18
6.2 Milepæler	18
6.3 Tids- og ressursplan.....	19
6.4 Gantt-skjema	20
7. Vedlegg.....	21

Antall ord: 2766

1. Mål og rammer

1.1 Bakgrunn

Diabetes type 1 er en kronisk autoimmun sykdom hvor personen har høyt blodsukkernivå grunnet relativ eller absolutt insulinmangel. Ved høyt blodsukker eller etter måltider må en type 1 diabetiker injisere insulin for å bringe blodsukkeret tilbake til normale nivåer.

Bakgrunnen for dette prosjektet er at det i dag er veldig mange diabetikere som før hver insulininjeksjon, manuelt må regne ut hvor mange internasjonale enheter insulin de må injisere vha insulinpumpe eller sprøyter for å korrigere blodsukkeret.

Det er ønskelig med en løsning som tar i bruk den nye teknologien i både telefon og medisinsk utstyr for å gjøre livet til type 1-diabetikere lettere.

En hensiktsmessig løsning for å avlaste de rammede bør kunne gjøre følgende:

- Hjelp til med å gi mer korrekte injiseringer.
- Skape trygghet for kjente, kjære og den syke.

Da denne typen diabetikere er helt avhengige av insulin til gitte tider, vil sikkerhet og etterfølgelse av internasjonale standarder være i absolutt høysete.

Det finnes i dag en del gode hjelpemidler for å måle blodsukkernivået med automatisk overføring av måledata til bla. insulinpumpe, pc og App på mobiltelefon og for visning av graf for utvikling av blodsukkernivået over tid. Det som er utfordringen er at alle må selv beregne karbohydratmengden og taste resultatet inn på insulinpumpa.

Prosjektet vil være basert på systemutviklingsprosjektet "DiaSet – mobil insulindosekalkulator" laget av Hans Åge Martinsen, Erik Grina Raassum og Øivind Bruvoll laget våren 2011, [ref. vedlegg 1.](#)

1.2. Prosjektmål

1.2.1 Effektmål

Hovedmålet med prosjektet er at brukergruppen får en lettere hverdag. Den skal stimulere til at brukeren får lyst til å bruke applikasjonen for å gjøre hverdagen lettere. Gjennom applikasjonen skal brukerne slippe å foreta manuelle og tungvinte utregninger av mengde insulin som skal injiseres etter måltider og aktiviteter. I samarbeid med oppdragsgiver har vi satt som mål:

- Optimalisere mengden insulin injisert med minst 10 %
- Redusere tiden det tar å manuelt regne ut injiseringsdoser med minst 50 %

1.2.2 Resultatmål

Med prosjektet ønsker vi å levere en fungerende applikasjon for både Android og iPhone baserte plattformer, med presise beregninger, samt et enkelt og motiverende grensesnitt for brukergruppen (fra 7 års alder).

Brukeren skal kunne registrere egne måltider ut ifra en database/katalog over matvarer.

Man vil da få en oversikt over hvor mange kalorier som finnes i et gitt måltid.

Blodsuktermåleren som brukeren har, registrerer blodsukkernivået i blodet, og tastes inn

manuelt i applikasjonen. Når måltid og blodsukkernivå er tatt i betraktning, skal

applikasjonen kalkulere antall enheter insulin som blodsukkerpumpa skal injisere. Deretter

skal en bekreftelsesmelding om antall enheter som skal injiseres sendes via SMS til

pårørende som bekrefter om dette er korrekt på en returmelding.

1.3. Rammer

Det er fastsatt en tidsramme for prosjektet fra januar til mai 2012. Rammene rundt eventuelle kostnader vil bli tatt i samhandling med oppdragsgiver. Da utviklingsgruppen [ref. punkt 3.1.1](#) allerede har Android-baserte telefoner, og oppdragsgiver [ref. punkt 3.1.2](#) har sagt seg villig til å benytte seg av sin Apple-baserte telefon, sees det på som lite sannsynlig at det påløper kostnader knyttet til teknisk utstyr. Det vil ikke bli nødvendig med leie av server da Høgskolen i Gjøvik har tildelt utviklingsgruppen webområde for prosjektet.

1.4. Målgruppen

Målgruppen for applikasjonen som skal utvikles i dette prosjektet er hovedsakelig unge type 1-diabetikere som på egenhånd må injisere insulin ved hjelp av insulinpumpe basert på manuelle utregninger før og etter måltider.

2. Omfang

2.1. Oppgavebeskrivelse

Hensikten med oppgaven er å utvikle applikasjonen Diaset for beregning av injisering av insulin etter inntak av karbohydrater i vanlig mat. Applikasjonen vil hovedsakelig være beregnet for barn og unge fra ca. 7års alder, og har som mål å fungere som en avlastning for diabetikere og pårørende som frem til i dag har sett seg nødt til å gjøre disse utregningene manuelt. En viktig del av oppgaven vil være å utforme og utvikle et grafisk grensesnitt som er tilpasset brukergruppen på en slik måte at det er både enkelt og motiverende å bruke, samt har alle funksjonelle løsninger i bunn.

Måten dette vil bli gjort på er at brukeren av applikasjonen selv skal kunne velge de matvarer/måltider som personen har inntatt ut i fra et grafisk grensesnitt i form av en liste/tabell løsning. Matvarer og måltider som inntas ofte vil også kunne lagres av brukeren i en egen kategori for favoritter, slik at disse vil være lett tilgjengelig.

Videre vil applikasjonen automatisk gjøre de utregningene som er nødvendig for deretter å gi brukeren informasjon om nødvendig antall enheter insulin som personen er nødt til å injisere for å få blodsukkernivået tilbake på et normalt nivå.

For at dette i det hele tatt skal være en forsvarlig løsning vil hver enkelt bruker av applikasjonen ved en førstegangs registrering måtte taste inn en unik konstant/faktor som brukeren har fått henvist fra sin fastlege. Denne verdien vil deretter være lagret for brukeren slik at applikasjonen til neste gang er klar for bruk. Dette er helt nødvendig for at applikasjonen skal kunne gi brukerne de rette utregninger for insulinenhetene.

Personens blodsukkernivå vil kun være mulig å taste inn manuelt, da det vil bli for tidskrevende å implementere kommunikasjon med blodsuktermåleren.

2.2. Avgrensning

- Applikasjonen vil primært fokuseres mot plattformene Apple og Android, da dette er de plattformene vi anser som mest aktuelle i dagens samfunn.
- Applikasjonen vil ikke ha støtte for kommunikasjon mellom blodsuktermåleren og telefonen som brukeren bærer med seg.
- Applikasjonen skal ikke på noen måte direkte injisere insulin.
- Applikasjonen er ikke ment som en kostholdsapplikasjon, men kun som hjelp til utregning av injiseringsdose av insulin.
- Applikasjonen vil i første omgang bli laget norskspråklig med muligheter for utvidelse til flere språk.

3. Prosjektorganisering

3.1. Ansvarsforhold og roller

Ansvarsforholdet er fordelt likt mellom alle personer i utviklingsgruppen. I prosjektet finnes det en utviklingsgruppe, en oppdragsgiver og en veileder.

Utviklingsgruppen består av tre studenter fra Høgskolen i Gjøvik, Gudbrand Olimb, Trond Hagehaugen og Michael Urrang som studerer henholdsvis, dataingeniør og medieteknologi. Oppdragsgiver for prosjektet er Steinar Dalby som har flere års erfaring fra sin jobb i Gitek AS. Veileder for prosjektet vil være Harald Liodden som er høgskolelektor ved Høgskolen i Gjøvik. Arbeidsfordelingen er foreslått slik:

Gudbrand Granaasen Olimb: Gruppeleder, ansvar for funksjonalitet med JavaScript.

Michael Kornbrekk Urrang: Designansvarlig, ansvar for koding av HTML og CSS.

Trond Hagehaugen: Databaseansvarlig, ansvar for oppsett og implementering av database.

3.1.1. Utviklingsgruppen

Gudbrand Olimb

Epost: gudbrand.olimb@gmail.com

Telefon: 90665190

Rolle: Prosjektleder

Trond Hagehaugen

Epost: trond.hagehaugen@gmail.com

Telefon: 95867796

Rolle: Kontaktperson

Michael Urrang

Epost: michael@urrang.com

Telefon: 95908065

Rolle: Grafikk-og designansvarlig

3.1.2. Oppdragsgiver

Steinar Dalby

Epost: sd@gitek.no

Telefon: 95787457

3.1.3. Veileder

Harald Liodden

Epost: harald.liodden@hig.no

Telefon: 61135293

3.2. Rutiner og regler i utviklingsgruppen

3.2.1 Prosjektleder

- Prosjektleder har det overordnede ansvaret for beslutningstaking i utviklingsgruppen. Ved uenigheter tar prosjektleder den endelige beslutningen.
- Prosjektleder har ingen rådighet til å ta beslutninger uten at utviklingsgruppen først er konsultert.
- Prosjektleder har rett til å signere på vegne av utviklingsgruppen.

3.2.2 Arbeidstid

- Alle i utviklingsgruppen skal jobbe et minstekrav på 30 timer i uka på skolen. Det vil si minimum 6 effektive arbeidstimer alle hverdager. Ved uker hvor deltakerne i utviklingsgruppen ikke har anledning til å jobbe minimumskravet, skal det tas en løpende vurdering om det tapte timeantallet skal tas igjen.
- Ved fravær knyttet til sykdom skal det gis beskjed så snart som mulig.
- Arbeidstiden kan justeres ved enighet, fra samtlige i utviklingsgruppen.

Dersom ett eller flere gruppemedlemmer ikke sier seg fornøyd med andre medlemmer i utviklingsgruppen sin innsats/timebruk, skal det sies i fra. I verste fall kan det føre til avskjedigelse fra utviklingsgruppen, dersom det skjer gjentatte ganger.

Beslutninger tas i fellesskap, ved uenigheter har prosjektleder siste ord, [ref. punkt 3.2.1](#). Ved uenigheter skal problemet(ene)/beslutningen(e) diskuteres og løses så godt som mulig i fellesskap. Er det umulig å løse beslutningene, tar gruppeleder beslutningen

Alle personer tilknyttet prosjektet skal til enhver tid ha en positiv og engasjerende holdning mot prosjektet og de andre personene involvert.

4. Planlegging, oppfølging & rapportering

4.1. Hovedinndeling av prosjektet

4.1.1 Systemutviklingsmodell

Utviklingsgruppen har diskutert hvilke utviklingsmodeller som passer for vårt prosjekt. Vi har sett på tre forskjellige modeller som vi føler at kan passe for vårt prosjekt: Fossefall, Inkrementell utvikling og Scrum.

Fossefall er en utviklingsmodell som gjennomfører flere steg, der en kravspesifikasjon ligger i bunn. Man utvikler så produktet steg for steg, gjennom de videre stegene Analyse, Design, Implementering og integrasjon, testing og vedlikehold. Utfordringen med denne modellen er at den er låst, det vil si at man i utgangspunktet ikke skal gå tilbake til steg som er gjennomført.

Inkrementell modell er en utviklingsmodell der kravspesifikasjonen ligger i bunn, men utvikles i inkremitter. Hvert inkrement i prosjektet følger de samme stegene som i fossefallsmodellen. Dette gjør at utviklingsgruppen raskt får se resultater, og får tidlig vite om man er på rett vei.

Scrum er en utviklingsmodell som også baserer seg på inkremitter, men utvikles med sprinter. En sprint er en fast avgrenset tidsperiode i den fastsatte utviklingstiden, det er mest vanlig med to til fire uker. Hver sprint skal resultere i et inkrement av applikasjonen. Hva som skal utvikles i en sprint er fastsatt av en product backlog. Denne product backlogen er en liste med oppdelte biter av hva som skal utføres i prosjektet. I Scrum har utviklingsgruppen daglige møter hvor det diskuteres hva som skal gjøres. Etter hver sprint har utviklingsgruppen en sprint gjennomgang, der man ser tilbake på en sprint. Mengden dokumentasjon i Scrum er opp til utviklerne å bestemme.

Utviklingsgruppen har tilgang til kravspesifikasjonen som ble laget i fjor, og oppdragsgiver ønsker at vi skal bruke denne kravspesifikasjonen som utgangspunkt i vår applikasjon. Vi ser på det som svært sannsynlig at det vil bli gjort endringer i løpet av prosjektet og siden fossefallsmetoden hindrer tilbakefall er ikke denne modellen aktuell

Inkrementell modell har vi vurdert som en meget aktuell modell i vårt prosjekt. Vi har kravspesifikasjonen i bunn, men føler at hvert inkrement burde ha en fast tidsplan. Noe inkrementell modellen ikke benytter seg av.

Scrum som utviklingsmodell gir en god oversikt over vårt prosjekt. Det å legge opp til en fast tidsplan på hva som implementeres underveis er en viktig faktor, da vi ønsker å bruke like lang tid på alle inkrementene i utviklingsprosjektet. Scrum legger også opp til jevnlig møter med oppdragsgiver, noe som vil være gunstig for utviklingsgruppen. Siden vi også ønsker noe dokumentasjon, falt det endelige valget på Scrum.

4.1.2 Prosessrammeverk/utviklingsmiljø

Utviklingsgruppen har testet ut forskjellige prosessrammeverk for utvikling av applikasjonen. Ettersom både oppdragsgiver og utviklingsgruppen ønsker å lage programmet for flere plattformer, hovedsakelig Android og iPhone, vil vi finne et utviklingsmiljø som benytter seg av felles kildekode. De forskjellige prosessrammeverk/utviklingsmiljøene vi har sett på følger:

Titanium Studio

Er et utviklingsmiljø som baserer seg på javascript som kildekode. For å bruke dette utviklingsmiljøet må vi programmere både på Mac OS for iPhone og Windows PC for Android siden man ikke kan lage en felles kildekode som vil fungere på begge plattformer.

Rhomobile

Er et utviklingsmiljø som bruker felles kildekode i programmeringsspråket Ruby. Utviklingsmiljøet fungerer med de fleste mobil-operativsystemene.

MoSync

Er et utviklingsmiljø som baserer seg på kildekode i C/C++ og HTML 5.

Sencha Tools og Phone Gap

Er et utviklingsmiljø som Gitek a/s bruker til å utvikle sine applikasjoner. Sencha Tools har ingen mulighet for debugging, noe som gjør valget av denne litt mer usikkert.

Noen endelig avgjørelse på hvilket av utviklingsmiljøene vi vil benytte, har ikke utviklingsgruppen tatt enda.

4.2. Plan for statusmøter og beslutningspunkter

Statusmøter med veileder arrangeres annenhver uke, og avtales via epost, hvor tidspunktet for neste møte avtales på statusmøtene. Statusmøter med oppdragsgiver arrangeres da det blir behov, og avtales via epost og/eller telefon, minimum hver 3. uke.

Kontaktperson har ansvar for å organisere og å avtale møtetidspunktene med veileder og oppdragsgiver. Kontaktperson har også det overordnede ansvaret for å skrive møtereferater fra hvert enkelt møte.

Beslutninger vil bli tatt fortløpende igjennom prosjektet, både i samhandling med veileder og oppdragsgiver. Ved uenigheter innad i utviklingsgruppen hvor hverken veileder eller oppdragsgiver må involveres, blir beslutninger tatt av prosjektleder, [ref. punkt 3.2.1.](#)

5. Organisering av kvalitetssikring

5.1. Dokumentasjon, standardbruk og kildekode

Scrum som utviklingsmodell, [ref. 4.1.1](#) legger opp til så mye dokumentasjon som utviklingsgruppen ønsker, og siden prosjektet berører flere medisinske aspekter og skal benyttes av barn fra 7 års alder vil vi sette fokus på å lage en god dokumentasjon/brukerveiledning av applikasjonen. Det vil parallelt med utviklingen av kode, lages dokumentasjon av applikasjonen.. Dette for å unngå at all dokumentasjon må skrives etter endt utvikling.

Alle møter blir dokumentert igjennom møtereferater som kontaktpersonen skriver etter hvert møte. Dette sikrer dokumentasjon av videre arbeid, samt meninger rundt prosjektet.

Hvordan kildekode blir behandlet etter prosjektets slutt er bestemt ut ifra regler satt av Høgskolen i Gjøvik, [ref. vedlegg 3](#).

5.3. Risikoanalyse (identifisere, analysere, tiltak, oppfølging)

Igjennom prosjektets gang vil det være flere risikomomenter rundt utvikling, testing og levering av prosjektet. Under utarbeidelsen av risikoanalysen har vi forsøkt etter beste måte å identifisere, analysere og planlegge mot visse hendelser som kan finne sted i løpet av utviklingsperioden. De forskjellige risikoaspektene er forsøkt definert godt i tabellen som følger:

Nr	Hendelse	Forebyggende Tiltak
1	Kostnadsrammene blir ikke overholdt	Holde god kontakt med oppdragsgiver for å til høyest mulig grad forebygge påløpende kostnader.
2	Lite kompetanse innen mobilarkitektur	Utviklingsgruppen vil benytte gode kontakter og ha møter med veileder samt oppdragsgiver for å samle informasjon til selvstudie igjennom hele perioden.
3	Fravær under prosjektet	Det vil under prosjektets periode bli løpende gitt oppgaver knyttet til fravær og for å forhindre fravær. Prosjektets utviklingsgruppe som har mest tyngde vil under prosjektets periode bli oppfordret til å holde seg i form for å hindre sykdom.
4	Kunde går konkurs og prosjekt stoppes	Her vil det bli tatt en konklusjon i samarbeid med Høgskolen i Gjøvik og oppdragsgiver vedrørende videre arbeid.
5	En tilsvarende applikasjon dukker opp på markedet	Prosjektet vil fortsette sin gang, da prosjektet er en viktig kilde for læring både for oppdragsgiver og utviklingsgruppen.
6	Applikasjonen har grensesnitt som ikke passer brukergruppen	Det settes opp møter med oppdragsgiver og diskuteres fortløpende hvilke tiltak som kan gjøres for å forbedre grensesnittet.
7	Tidsfrister blir ikke overholdt	Ved å følge gantt skjemaet, samt ha god kommunikasjon og ha gode jevnlig statusmøter.
8	At programvaren ikke er kompitabel med både android og apple	Ved å sørge for jevnlig testing mot begge plattformer, og samråd med veileder
9	Tap av data	Forebyggende tiltak ved å kontinuerlig ta backup

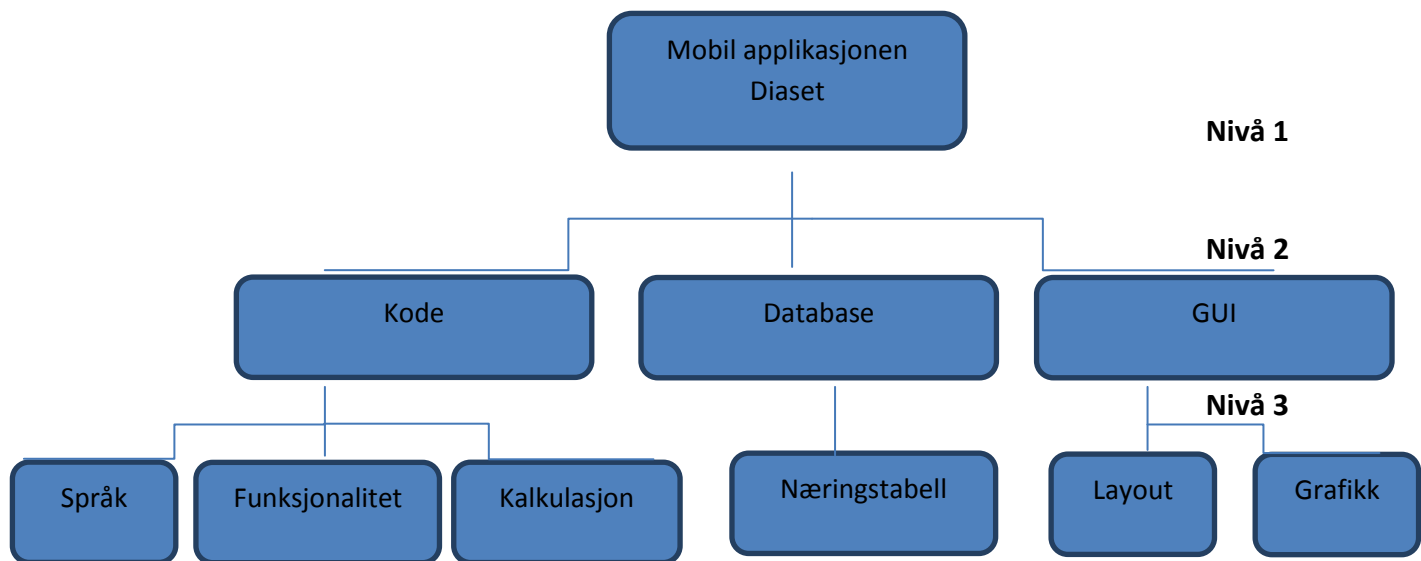
Det er videre ut fra tabellen foretatt en nærmere risikoanalyse som definerer hvilke hendelser som ansees som mest sannsynlig å inntreffe.

	Ufarlig	Farlig	Kritisk	Katastrofalt
Meget sannsynlig				
Sannsynlig	#5			#2
Lite sannsynlig		#3	#6, #7	#8, #9
Usannsynlig	#1			#4

6. Plan for gjennomføring

6.1 Work Breakdown Structure

Prosjektet er bygget opp etter visse aktiviteter som man kan se under, her ser man de forskjellige nivåer i prosjektet, samt hvilke aktiviteter som ligger under under disse ulike nivåene.



6.2 Milepæler

Milepæler i prosjektet vil man se ut av gantt – skjemaet, men under ser man de fastsatte datoene for enkelte milepæler i prosjektet.

- Statusrapporter følgende datoer - 27.2, 25.3 og 30.4.
- Betaversjon, 25.3.
- Prototypversjon, 16.4.
- Ferdigstilling, 23.5.

6.3 Tids- og ressursplan

Aktivitet	Tidsrom
Oppstart av prosjekt	Uke 48, 2011
Prosjektplan	Uke 1 – 4, 2012
Etablering av Hjemmeside	Uke 3 – 5, 2012
Utarbeidelse av teori	Uke 5 – 7, 2012
Utarbeidelse av kode, database og GUI (beta)	Uke 6 – 12, 2012
Testing av kode	Uke 6 – 12, 2012
Utarbeidelse av kode, database, GUI (prototyp)	Uke 12 – 16, 2012
Testing av kode	Uke 12 – 16, 2012
Analyse av applikasjon	Uke 16 – 18, 2012
Ferdigstilling av prosjekt	Uke 18 – 21 (23.05.2012), 2012
Arbeid med fremføring	Uke 21

6.4 Gantt-skjema

ID	Task Mode	Task Name	Duration	Start	Finish	Gantt Chart											
						Decem	Janu	Febru	March	April	May	E	B	M	E	B	M
1	Task Mode	Diaset	128 days	Mon 28.11.11	Wed 23.05.12	[Gantt bar for Diaset]											
2	Task Mode	Jobb med prosjektplan	20 days	Mon 02.01.12	Fri 27.01.12	[Gantt bar for Jobb med prosjektplan]											
3	Task Mode	Innløring av prosjektplan	1 day	Fri 27.01.12	Fri 27.01.12	[Gantt bar for Innløring av prosjektplan]											
4	Task Mode	Innløring av prosjektavtale	1 day	Fri 27.01.12	Fri 27.01.12	[Gantt bar for Innløring av prosjektavtale]											
5	Task Mode	Webseite	102 days	Tue 10.01.12	Wed 30.05.12	[Gantt bar for Webseite]											
6	Task Mode	Webseite oppdatert	1 day	Fri 03.02.12	Fri 03.02.12	[Gantt bar for Webseite oppdatert]											
7	Task Mode	Utprøving av utviklingsmiljø og samling av informasjon	26 days	Mon 02.01.12	Sun 05.02.12	[Gantt bar for Utprøving av utviklingsmiljø og samling av informasjon]											
8	Task Mode	SCRUM utvikling:				[Gantt bar for SCRUM utvikling:]											
9	Task Mode	Beta versjon	40 days	Mon 06.02.12	Thu 29.03.12	[Gantt bar for Beta versjon]											
10	Task Mode	Sprint 1 - Opprette applikasjon og database	14 days	Mon 06.02.12	Thu 23.02.12	[Gantt bar for Sprint 1 - Opprette applikasjon og database]											
11	Task Mode	Lage "skal" til applikasjonen for begge plattformen	6 days	Mon 06.02.12	Mon 13.02.12	[Gantt bar for Lage "skal" til applikasjonen for begge plattformen]											
12	Task Mode	Implementering av matvaretabell	8 days	Mon 13.02.12	Wed 22.02.12	[Gantt bar for Implementering av matvaretabell]											
13	Task Mode	Sprint 2 - Funksjonalitet	14 days	Wed 22.02.12	Mon 12.03.12	[Gantt bar for Sprint 2 - Funksjonalitet]											
14	Task Mode	Kalkulering, koblinger, registrerings	14 days	Wed 22.02.12	Mon 12.03.12	[Gantt bar for Kalkulering, koblinger, registrerings]											
15	Task Mode	Statusrapport	1 day	Mon 27.02.12	Mon 27.02.12	[Gantt bar for Statusrapport]											
16	Task Mode	Sprint 3 - Design, feilretting og testing	14 days	Mon 12.03.12	Wed 28.03.12	[Gantt bar for Sprint 3 - Design, feilretting og testing]											
17	Task Mode	Lage Hoveddesign for applikasjonen	6 days	Mon 12.03.12	Mon 19.03.12	[Gantt bar for Lage Hoveddesign for applikasjonen]											
18	Task Mode	Statusrapport	1 day	Sun 25.03.12	Sun 25.03.12	[Gantt bar for Statusrapport]											
19	Task Mode	Feilretting og test av betaversjon	8 days	Tue 20.03.12	Thu 29.03.12	[Gantt bar for Feilretting og test av betaversjon]											
20	Task Mode	Prototyp versjon	38 days	Thu 29.03.12	Mon 21.05.12	[Gantt bar for Prototyp versjon]											
21	Task Mode	Sprint 4 - Ferdigstille kode	14 days	Thu 29.03.12	Tue 17.04.12	[Gantt bar for Sprint 4 - Ferdigstille kode]											
22	Task Mode	Ferdigstille kode	7 days	Thu 29.03.12	Fri 06.04.12	[Gantt bar for Ferdigstille kode]											
23	Task Mode	Ferdigstille design	7 days	Fri 06.04.12	Mon 16.04.12	[Gantt bar for Ferdigstille design]											
24	Task Mode	Sprint 5 - Utprøving og vurdering av applikasjonen	14 days	Mon 16.04.12	Thu 03.05.12	[Gantt bar for Sprint 5 - Utprøving og vurdering av applikasjonen]											
25	Task Mode	Utprøving, testing og evaluering av applikasjonen	14 days	Mon 16.04.12	Thu 03.05.12	[Gantt bar for Utprøving, testing og evaluering av applikasjonen]											
26	Task Mode	Statusrapport	1 day	Mon 30.04.12	Mon 30.04.12	[Gantt bar for Statusrapport]											
27	Task Mode	Ferdigstilling av prosjekt	18 days	Mon 30.04.12	Wed 23.05.12	[Gantt bar for Ferdigstilling av prosjekt]											
28	Task Mode	Innløring av prosjekt	1 day	Wed 23.05.12	Wed 23.05.12	[Gantt bar for Innløring av prosjekt]											

7. Vedlegg

1. DiaSet - Del 1-Prosjektbeskrivelse-systemutvikling
2. Oppgave 13 - DiaSet - insulinkalkulator for Smarttelefon
3. Prosjektavtale

10.8 Product backlog

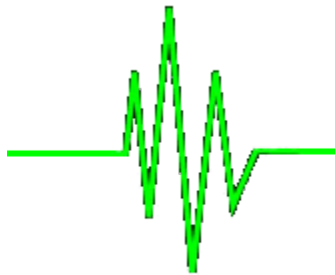
Product Backlog - Grov oversikt

Nr	Beskrivelse	Sprint nr
1	Opprette prosjekt	1
2	Lage et kjørbart " skall" for applikasjonen	1
3	Implementere database	1
4	Lage formler og funksjoner for kalkulasjon	2/4
5	Implementere formler og funksjoner i applikasjonen	2/4
6	"Favoritt"/ Lagre måltider	2/4
7	SMS til pårørende	2/4
8	Design	3

10.9 Prosjekt 2011: DiaSet - Del 1-Prosjektbeskrivelse-systemutvikling

Prosjekt “DiaSet”

Mobil insulin-dosekalkulator



Hans Åge Martinsen (090440)

Erik Grina Raassum (090373)

Øyvind Bruvoll (080174)

Endringer fra levering #1:

- Forbedret gannt-skjema

- Risikoanalyse “begrunnelse”

- Effektmål

- Litt på slutten av begrunnelse av systemutviklingsmodell

- Projisering av kostnader gjort mer realistisk

- Valg av UML-verktøy begrunnet under kvalitetsikring

Innholdsfortegnelse

Innholdsfortegnelse

1.1 Bakgrunn

1.2 Prosjekt mål

1.2.1 Resultatmål

1.2.2 Effektmål

1.2.3 Læringsmål

1.3 Målgruppen

1.4 Rammer

2 Omfang

2.1 Oppgavebeskrivelse

2.2 Avgrensninger

2.3 Prosjektorganisering

2.3.1 Ansvarsforhold

2.3.2 Øvrige roller og bemanning

3.1 Hovedinndeling av prosjektet

3.2 Krav til statusmøter

4 Risikoanalyse

5 Kvalitetssikring

5.1 Organisering av kvalitetssikring

5.2 Kvalitetssikring

5.2.1 Valg av UML-verktøy

6 Gjennomføring

6.1 Gantt-skjema

Kilder

1 Prosjektmål og rammer

1.1 Bakgrunn

Diabetes type 1 er en kronisk autoimmun sykdom hvor personen har høyt blodsukkernivå grunnet relativ eller absolutt insulinmangel. Ved høyt blodsukker eller etter måltider må en type 1 diabetiker injisere insulin for å bringe blodsukkeret tilbake til normale nivåer.

Bakgrunnen for dette prosjektet er at det i dag er veldig mange diabetikere som før hver insulin-injeksjon, manuelt må regne ut hvor mange internasjonale enheter insulin de må injisere vha insulinpumpe eller sprøyter for å korrigere blodsukkeret. Dette utføres fordi det kan være vanskelig for personen å vite med noenlunde sikkerhet hvor mange kilokalorier det var i måltidet de nettopp spiste. Utrekningen blir derfor til tider upresis, og man tar enten for få eller for mange enheter insulin. Denne usikkerheten, i tillegg til selve arbeidet med å regne ut dosene, fører til en ekstra belastning for personen og de pårørende.

I takt med at smarttelefoner er blitt allment utbredt, er det således blitt ønskelig med en løsning som tar i bruk den nye teknologien i både telefon og medisinsk utstyr for å gjøre livet til type 1-diabetikere lettere.

En hensiktsmessig løsning for å avlaste de rammede bør kunne gjøre følgende:

- Hjelp til med å gi mer korrekte injiseringer.
- Skape trygghet for kjente, kjære og den syke.

Da denne typen diabetikere er helt avhengige av insulin til gitte tider, vil sikkerhet og etterfølgelse av internasjonale standarder være i absolutt høysete.

1.2 Prosjektmål

1.2.1 Resultatmål

Resultatmålet er at vi skal kunne levere en fungerende applikasjonen som er i stand til å ta imot kontinuerlige målinger av blodsukkernivået, og i forlengelse av denne dataen, skal brukeren kunne registrere egne matvarer og således også måltider, hvis disse ikke allerede finnes på ekstern server, for å bruke mengden kilokalorier i disse til å beregne antall internasjonale enheter insulin som skal injiseres.

I tillegg så skal applikasjonen kunne lagre relevant statistikk. Man skal også kunne ha mulighet til å bruke mobilens SMS-system til å kommunisere med personens formynder ved krisetilfeller, som ved farlig høyt eller lavt blodsukker. Støtte for flere språk og kulturspesifikke måltider er vesentlig.

1.2.2 Effektmål

- Tilfredsheten til brukerne av prototypen i en test-gruppe skal være så stor at de vil bruke

applikasjonen i hverdagen.

- Tilstrebe en optimalisering av mengden insulin brukt av brukerne med 25%
- Redusere tiden den gjennomsnittlige diabetespasient bruker på utregninger med 50%.
- I forlengelse av overnevnte punkter, øke livskvaliteten til brukeren ved å fjerne behovet for manuelle, usikre utregninger.

1.2.3 Læringsmål

For å vurdere hvorvidt prosjektet er gjennomførbart, og til hvilken grad, er vi nødt til å konferere med diabetikere, pårørende og eksperter for å tilegne oss relevant kunnskap om hvordan diabetes fungerer og hvilke sykdomsspesifikke omstendigheter som foreligger. Bedriften må også sette seg inn i hvordan utviklingsmiljøet, standarder er på nye smart-telefoner og kontakte produsenter av insulinpumper for å få innsikt i den trådløse standarden de har valgt i sitt produkt for å vurdere hvorvidt trådløs måling av blodsukker vha. sensor (f. eks i en insulinpumpe, men det kan også være en selvstendig sensor) er realistisk å kunne implementere.

1.3 Målgruppen

Målgruppen for prosjektet er type 1-diabetikere, unge og eldre som egenhendig må injisere insulin ved hjelp av insulinpumpe basert på manuelle utregninger før og etter måltider. Disse vil benytte vår applikasjon til å lette arbeidet med å dosere nødvendige enheter insulin.

1.4 Rammer

Vi forespeiler at en fornuftig tidsramme for utviklingen av dette systemet ligger på omkring 6 måneder. Da vi ser for oss at utviklingen kan gjennomføres innenfor tidsrammen ovenfor med tre ansatte, vil prosjektet tilsvare noe mellom 1,5 og 2 årsverk. Med en grunnlønn på 400,000,- gir dette lønnskostnader i området 900,000 og 1,000,000,-.

Vi behøver også å ta utviklingskostnader for platformene, kjøp av nødvendig teknisk utstyr for å lage prototyper, leie av server inn i betraktningen. Et edruelig overslag, som samtidig gir oss litt slingsringsmonn ifht potensielle tidsrammeoverskridelser, ligger da på 2,500,000,-.

2 Omfang

2.1 Oppgavebeskrivelse

Dette programmet utvikles for å avlaste diabetikere med behov for å regne ut antall enheter insulin som må injiseres før og etter måltider. Måten dette forespeiles utført er at personen velger matvarer og/eller hele måltider fra et grafisk grensesnitt eller en form for liste/tabell og deretter får utregnet det nødvendige antall enheter insulin personen er nødt til å injisere for å få blodsukkernivået tilbake til et gunstig/normalt nivå.

Instillingene som brukeren er nødt til å defineres ved første kjøring av programmet. er blant annet en unik konstant som brukeren har fått fra sin lege og som brukes under utregning av insulinenheter.

Han kan også registrere egne preferanser hva gjelder måltider, matvarer og lagring av statistikk av antall doser injisert. Det vil være mulig å sette passord på innstillingsmenyen slik at uvedkommende ikke vil kunne endre den unike konstanten til brukeren.

På en ekstern server ligger det til enhver tid oppdaterte næringstabeller for de mest kjente matvarene, og her vil brukeren kunne filtrere på mange forskjellige variable. Støtte for flere språk skal være støttet, slik at programmet er egnet for mennesker fra mange forskjellige kulturer.

Brukeren vil møte et simpelt grafisk grensesnitt med ulike muligheter videre, herunder statistikk, innstillinger, utregninger og brukerveiledning. Fra innstillingene kan utregnings-, statistikk og måltidsrelevante innstillinger endres; man skal kunne vise og eksportere statistikk; man skal kunne registrere og slette måltider og beregne insulindoser; og det skal være en brukerveiledning hvor brukeren kan få hjelp til å forstå hvordan applikasjonen opererer.

Personens blodsukker skal enten bli hentet ved behov fra en eller annen form for ekstern sensor (for eksempel en insulinpumpes), eller i tilfeller hvor dette ikke foreligger må brukeren før hvert måltid legge inn nivået målt manuelt, slik at korreksjonsdosen kan utregnes før man regner ut dosen som skal injiseres etter måltid.

Det er vesentlig at løsningen for utregning av insulinheter er enkel å bruke, samtidig som den ivaretar funksjonaliteten beskrevet ovenfor i grove trekk. Dette poengteres slik at programmet er like brukbart for alle grupper mennesker, ung som gammel.

Både for sensorløsningen og måltidsfunksjonaliteten vil vi undersøke eksisterende løsninger for å eventuelt redusere utviklingskostnader og holde oss innenfor gjeldende tidsrammer.

2.2 Avgrensninger

Det er fornuftig å avgrense prosjektet, slik at man får en presis oversikt over hvilken funksjonalitet som faktisk skal implementeres.

Det presiseres derfor at vår applikasjon ikke på noen måte skal:

- utvikles for å automatisk injisere insulin, hverken direkte eller indirekte
- fremme kostholdsrad for brukerne
- legge til rette for direkteinntekter fra reklame

2.3 Prosjektorganisering

Prosjektet vil innledes med et møte med oppdragsgiver, hvor vi i etterkant bestemmer oss for prosjektleder. Denne personen vil ha en lite formell rolle i hverdagen under utviklingen, men vil styre ordet i møter og ivareta overordnet kontroll over prosjektet. Vedkommende kan dra prosjektet i ønsket retning ved uforutsette behov.

Følgende mennesker vil være involvert i prosjektet:

- Hans Åge Martinsen - 09HING - prosjektleder

- Erik Grina Raassum - 09HBDRA - utvikler
- Øyvind Bruvoll - 09HBDRA - utvikler
- Steinar Dalby - oppdragsgiver

2.3.1 Ansvarsforhold

Da vi er liten gruppe å tre utviklere hvor ansvarsforholdene er transparente og veldefinerte, er det ikke behov for å fastsette roller. Alle vil kunne bidra likt i alle oppgaver på en demokratisk måte, men prosjektleder sitter med vetorett for situasjoner hvor behovet for eksplisitt lederskap blir presserende.

2.3.2 Øvrige roller og bemanning

Bemanningen vil bestå av overnevnte tre studenter, samt oppdragsgiver, i tillegg til eventuelt innleid arbeidskraft som tiltrer etter behov. Etter at grunnmodulen er implementert, settes det parallelt med utviklingen i gang en mangfoldig betatesterguppe, hvor vi både har unge og gamle diabetikere som hjelper oss å finne feil, mangler og områder med forbedringspotensiale. De med teknisk innsikt får oppdateringer tidligere enn andre, for å kunne luke ut de mest kritiske og øyensynlige bugs.

3 Planlegging, oppfølging og rapportering

3.1 Hovedinndeling av prosjektet

For å få utviklet systemet, har vi i all hovedsak vurdert fossefallsmetoden, en gjenbruksrettet modell, extreme programming og en inkrementell modell.

I **fossefallsmodellen** så jobber man sekvensielt, hvor hver fase fullføres før arbeidet på den neste tiltar. Denne modellen har en rekke gunstige kvaliteter, blant annet uovertruffent god oversikt, samtidig som den bidrar til presise fremdriftsplaner og dermed prosjektets tidsbruk og budsjettestimater. Den er veldig fornuftig å ta i bruk ved prosjekter hvor man har tidligere erfaring fra lignende prosjekter, men låser deg i form av å ikke tillate endringer i de ferdige fasene.

Applikasjonen vi skal utvikle, vil starte med en grunnfunksjonalitetsmodul hvor vi legger til ytterligere innholdsmoduler etter hvert som vi får innholdsfunksjonalitet plass. Disse vil mest sannsynligvis føre til at vi må utføre noen endringer i tidligere ferdigstilt programvare, hvilket ikke er anbefalt i fossefallsmetoden. Oppdragsgiver ønsker også hyppig kontakt og involvering i utviklingen. Dette gjør at denne metoden i sin reneste form ikke egner seg for vårt prosjekt.

Vi vurderte også **XtremeProgramming** som systemutviklingsmodell, da den er godt egnet for prosjekter som er tuftet på iterasjoner med små forbedringer, tett kontakt med de faktiske brukerne parprogrammering. Den fører til at man raskt ser framgang i prosjektet, samt får gode tilbakemeldinger på tidlige tidspunkt i utviklingen.

Årsaken til at vi avskrev extreme programming som modell, er at vi ikke planlegger å parprogrammere da det sett i sammenheng med prosjektets størrelse og omfang ikke gjør det lønnsomt, i tillegg til at programmet vil være basert på moduler av annen størrelse og at disse

fremkommer fra lengre inkremitter enn XP foreskriver.

Dette systemet vil i høy grad kunne dra nytte av teknologi som allerede eksisterer, eller kan videreutvikles og skreddersys til våre spesifikke formål. En gjenbruksorientert modell, hvor man i stor stil høster fruktene av en slik tankegang, er et rasjonelt valg til dette prosjektet.

Modellen vi til slutt endte opp med er den **inkrementelle** med innslag fra fossefalls- og gjenbruksmodellen. Denne metoden gir oss godt med slingringsmonn grunnet sin fleksible natur, som vil være gunstig, sett i lys av vårt naive syn på systemutvikling ifht. tidsfrister og de ulike emnespesifikke delenes omfang. Den modulatorienterte jobbingen fører til lavere risiko fordi de viktigste delene av programmet utvikles først, og også blir testet utførlig først. Dermed ender man ikke opp med dårlige implementasjoner. Kunden er også involvert i stor grad, og kan tidlig se og vurdere resultatene som fremkommer.

Denne modellen har også noen ulemper, som blant annet at oversikten over systemet kan være mindre enn ved f.eks fossefallsmetoden, hvor man til enhver tid har store mengder dokumentasjon tilgjengelig. Likevel veier fordelene opp for eventuelle ulemper, ettersom vi tar oss god tid til planlegging i form av forprosjektsplan, kravspesifikasjon og designdel, og således låner litt av fossefallsmetodens styrker. I tillegg står vi sterkere stilt ved å kunne detaljspesifisere kravene til hver enkelt modul før kodingen tiltar.

Vi vil bruke den inkrementelle modellen til å rulle ut få, men store moduler med funksjonalitet som integreres i grunnapplikasjonen, slik at betatestere tidlig kan hjelpe til med tilbakemeldinger. Disse inkrementene forventes å ferdigstilles med ett måneds mellomrom fra hverandre. Dette sikrer at de som faktisk skal benytte applikasjonen i hverdagen får sine ønsker oppfylt. Vi vil også tilpasse en gjenbruksrettet modell til undersøke om det finnes programvare tilgjengelig som vi kan utvikle videre, eller tilpasse til våre behov.

3.2 Krav til statusmøter

Statusmøter bør følge hver gang radikale forandringer inntreffer prosjektes løp, for eksempel ved ferdigstilling av en ny modul. Disse kan også avtales av kunde og utviklerteam ved andre anledninger eller behov. Dersom ikke dette er nok, kan det vurderes om ukentlige møter skal gjennomføres for å diskutere mindre betydelige karakteristika ved prosjektet.

På de lengre statusmøtene skal det også bestemmes hvorvidt moduler er av tilfredsstillende kvalitet, hvorvidt forlenget beta-testing er nødvendig, og når neste modul tenkes påbegynt. Dessuten detaljspesifiseres kravene til funksjonalitet i neste modul.

4 Risikoanalyse

Det er flere risikomomenter som kan utgjøre en fare for at prosjektet ikke blir fullført innenfor spesifisert tidsramme, og med den kvaliteten kunden forventer, blant annet:

- Valg av insulinpumpe/sensor
- Sikkerhet overfor brukere mtp. nøyaktighet i utregninger og sensormålinger

- Korrekt etterfølgelse av lover og standarder

I dette prosjektet vil vi identifisere, analysere og planlegge for hendelser som kan ha en negativ innvirkning på utførelsen av utviklingen, og nivåmarkere graden av sannsynlighet og konsekvens dersom de inntreffer: Grønn for minimal, gul for moderat, orange for kritisk og rød for katastrofal

#	Hendelse	Tiltak
1	Nåværende teknologi støtter ikke tilkobling mellom sensor og mobil.	Det må lyses ut et prosjekt om å utvikle en enhet som fungerer. Eventuell skrinlegging eller omstilling av prosjektets karakter.
2	Kostnadsrammene for prosjektet overskrides	Møte med kunde og de ansatte avtales sporenstreks for å avklare situasjonen.
3	Kompetansen for å utvikle på en mobilarkitektur er ikke tilfredsstillende innad i bedriften.	Innleiing av ekstern arbeidskraft, eventuelt kursing og selvstudium.
4	Sykdom eller fravær innenfor gruppen.	Det settes opp et møte for å diskutere videre arbeidsfordeling, eventuelt leie inn ekstra arbeidskraft.
5	Kravene for applikasjonen endres radikalt	Omstilling til de nye kravene hvis de ikke involverer et uforholdsmessig ekstraarbeid.
6	For liten kapasitet i prosjektgruppa.	Tidsfristen utsettes, eller arbeidskraft innleies.
7	Kunde eller leverandør av kritisk maskin- eller programvare går konkurs.	Oppkjøp av rettigheter til å utvikle applikasjonen egenhendig videre. Diskutér andre leverandører.
8	Kontraktbrudd mellom kunde og utviklerteam	Prosjektet settes umiddelbart på pause inntil en gjensidig løsning er tilstrebet.
9	En konkurrerende bedrift utgir et lignende produkt innen vår ferdigstillelse.	Sette opp et møte med kunde og diskutere videre forløp. Også diskutere om dette forandrer på krav til våres applikasjon.
10	Applikasjon er ikke tilpasset vår brukergruppe godt nok og blir derfor ikke brukt fremfor andre løsninger.	Tett kontakt med betestere, og fortløpende løsninge
11	Utregningene er ikke presise nok, enten grunnet gale sensormålinger eller gal informasjon i næringstabell	Næringstabellene må hele tiden holdes oppdatert; sensorløsningen må testes utførlig opp mot målinger tatt manuelt for å finne ut av presisjon; betestere har hele tiden mulighet til å kontakte oss ifht. feilsituasjoner

Deretter plottet vi de seks mest prosjektkritiske, mulige hendelsene inn i en 4x4-tabell, hvor sannsynlighet og konsekvensen ved inntreffelse er kombinert med en tilhørende (beskrivende) farge.

	Lite farlig	Farlig	Kritisk	Katastrofalt
Svært sannsynlig				
Nokså sannsynlig		#2, #4	#3, #9	
Lite sannsynlig			#1	#11
Usannsynlig				

Av figuren ovenfor kan man lese at det er få risikomomenter av meget stor signifikans. Derimot er det flere av moderat betydelighet, blant annet #3 - at vi er nødt til å bruke tid til å sette oss inn i systemer, programmeringsspråk og grensesnitt som er nødvendige for å få utført oppgaven, og #9 - at et tilsvarende produkt lanseres innen vi er ferdige. Et mottiltak for #3 er å bruke en del tid på selvstudium eller kursing, selv om innleiing av eksternt arbeidskraft for å motvirke mangelen på kompetanse er det beste alternativet, og lar utviklingen skride fremover raskere. #9 får man ikke gjort særlig med, annet enn å utgi et tilfredsstillende produkt som skiller seg ut på én eller flere måter fra de konkurrerende og således kan kapre markedsandeler basert på særegenhet.

Punkt #11 er prekært, og vi vil prioritere mottiltak for å motvirke dets eventuelle innvirkning på prosjektets gang. Dette fordi det er direkte knyttet til hovedfunksjonaliteten i applikasjonen. Ved tilfellene #2 og #4 må det også straks kalles inn til et møte for å avklare situasjonen, og bestemmelser for hvorvidt eller hvilke mottiltak som er nødt til å iverksettes. #1 er også et viktig punkt, men applikasjonen skal også kunne brukes av diabetikere uten blodsukkernivåsensor, slik at en eventuell teknologimangel ikke negerer eksistensgrunnlaget til prosjektet.

5 Kvalitetssikring

5.1 Organisering av kvalitetssikring

Kvalitetssikringen vil i hovedsak bli utført av betatestergrupper som fra hovedfunksjonalitetsmodulen påbegynnes er med i utviklingen. Disse kalles inn til møter regelmessig, og oppfordres også til å egenhendig rapportere feil, mangler eller forbedringsområder.

Det vil bli kjørt betagrupper helt fra andre modul er halvferdig, frem til rett før leveringsdatoen. Halvparten av brukerne i betatestergruppa vil bli skolert i bruk av applikasjonen, mens den siste halvdel vil være vanlige brukere som tester for første gang. Under beta av modul 1 og 2, vil brukerne bli bedt om å kontrollsjekke alle utregninger med vanlig regning.

5.2 Kvalitetssikring

Under betatesting vil vi få utførlig testet hvorvidt de ulike modulene virker tilfredsstillende,

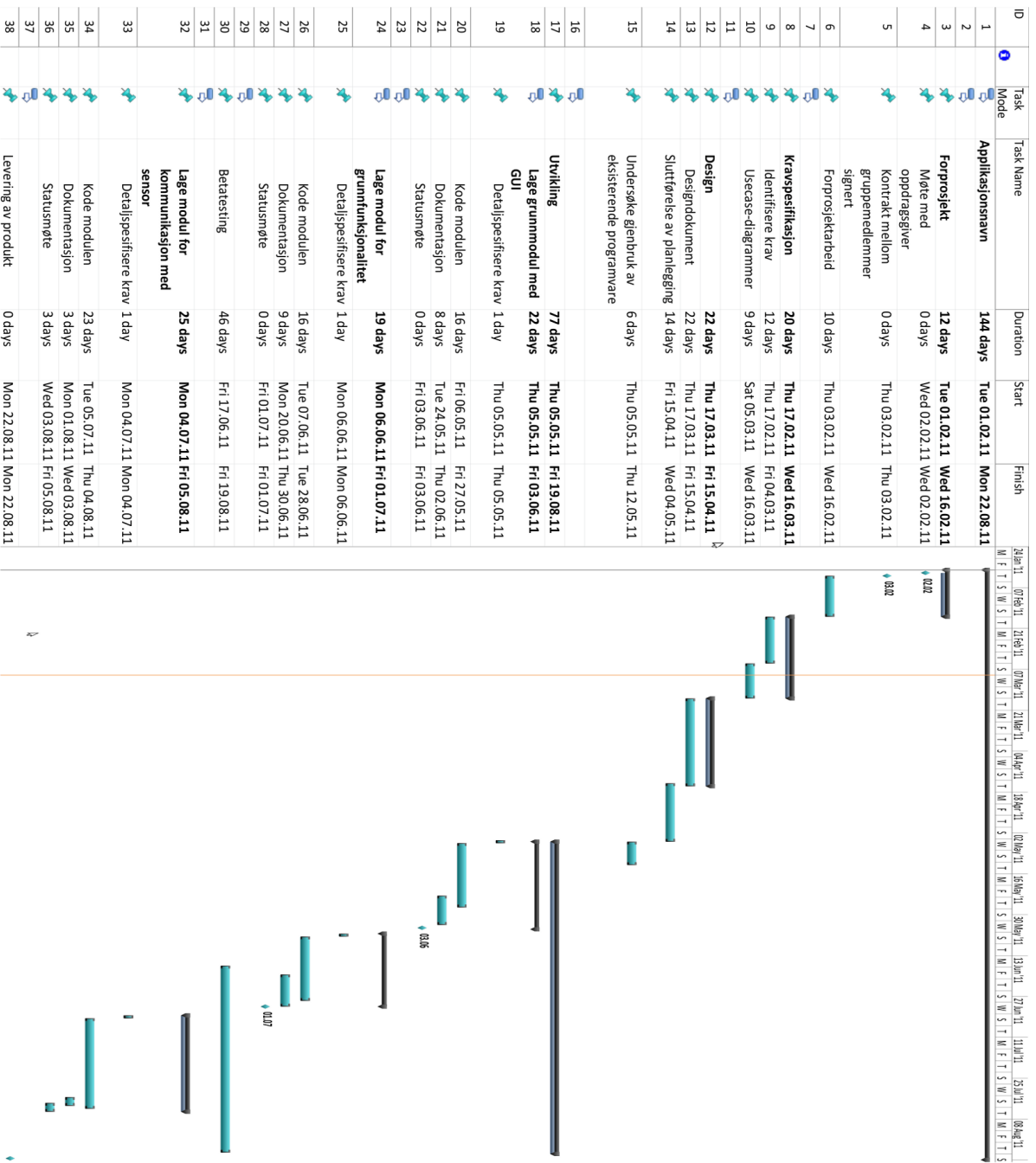
5.2.1 Valg av UML-verktøy

Vi velger å bruke freewareproduktet "Violet UML" for å lage brukstilfellediagram og konseptuelt klassediagram. Dette valget falt vi på etter å ha vurdert Microsofts Visioprodukt og flere andre gratis UML-verktøy. I forhold til pris har vi tusener å spare på å ta en slik avgjørelse, og vi støtter i tillegg opp om åpen kildekode i kommersiell sammenheng.

Et annet argument er at funksjonaliteten i de respektive programmene er mer eller mindre lik. Rent estetisk sett kan kanskje Visio være et sterkere kort, men har dessverre tilkortkommenheter hvis man også tar prisen i betraktning.

6 Gjennomføring

6.1 Gantt-skjema



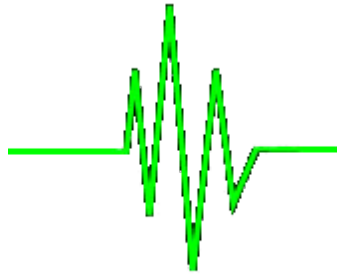
Kilder

1. INF4120. "Systemutviklingsmodeller". Lokalisert 3/2 på Verdensveven: folk.uio.no/sumandek/inf4120/Utviklingsmodeller.doc
2. Sommerville, Ian. (2011). Software Engineering (9. utg.). USA: Pearson Education.
3. Wikipedia. "Diabetes Mellitus". Lokalisert 3/2 på Verdensveven: http://no.wikipedia.org/wiki/Diabetes_mellitus
4. Kunnskapscenteret. "Risikoanalyse". Lokalisert 3/2 på Verdensveven: http://www.ogbedreskaldetbli.no/metoder_verktoy/Verktoykasse/Forankre_og_organisere_forbedringsarbeidet/ROS-analyse/1301

10.10 Prosjekt 2011: DiaSet - Del 2-Kravspesifikasjon

Prosjekt “DiaSet”

Mobil insulin-dosekalkulator



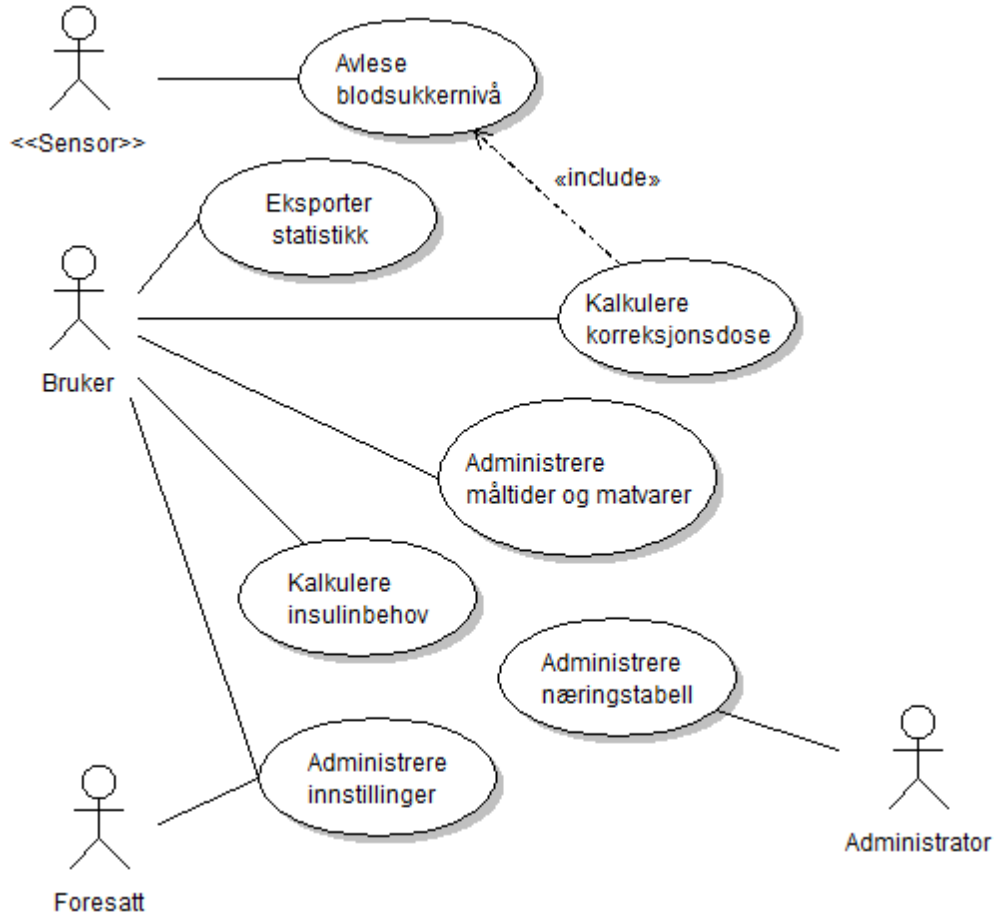
Hans Åge Martinsen (090440)

Erik Grina Raassum (090373)

Øyvind Bruvoll (080174)

Del 2, kravspesifikasjon

7 Brukstilfellediagram



Usecase-diagram laget i Violet UML.

8 Høynivå brukstilfeller

Navn: Administrere næringstabell

Mål: Målet er å vise, legge til, fjerne og redigere matvarer i en næringstabell

Primær aktør: Administrator

Beskrivelse: Næringstabellen er viktig for tilnærmet all funksjonalitet i applikasjonen, og letter arbeidet med å utarbeide måltider og således kalkulere riktige insulindose.

Navn: Administrere innstillinger

Mål: Målet er for brukeren å få applikasjonen funksjonell etter brukerens personlige behov.

Aktører: Bruker (og eventuelt en foresatt)

Beskrivelse: Brukeren skal kunne endre innstillingene som programmet baserer sin funksjonalitet på. En eventuell foresatt skal også ha mulighet til å endre brukerinntstillinger.

Navn: Administrere måltider og matvarer

Mål: Målet er å tilby brukeren en mulighet til å legge til, endre, fjern og vise måltider og matvarer fra profilen.

Primær aktør: Bruker

Beskrivelse: Dette er funksjonaliteten som hovedmodulen baserer seg på. Brukeren skal kunne administrere sine måltids- og matvarepreferanser og bruke disse i kalkuleringsøyemed.

Navn: Kalkulere korreksjonsdose

Mål: Målet er å kalkulere riktig korreksjonsdose med insulin som skal injiseres før et måltid inntas.

Primær aktør: Bruker

Beskrivelse: Brukeren skal kunne få kalkulert en korreksjonsdose basert på en sensormåling, eller ved å selv spesifisere nåværende blodsukkernivå manuelt.

Navn: Kalkulere insulinbehov

Mål: Målet er å kalkulere riktig insulindose som skal injiseres etter at et måltid er inntatt.

Primær aktør: Bruker

Beskrivelse: Brukeren skal kunne få kalkulert riktig mengde insulinheter basert på et utvalg av matvarer eller et måltid.

Navn: Avlese blodsukkernivå

Mål: Målet er å få blodsukkernivået avlest fra en ekstern sensor, være seg insulinpumpe eller en selvstendig enhet, vha. trådløs teknologi.

Primær aktør: Bruker/Sensor

Beskrivelse: Ved kalkulering av korreksjonsdose kan brukeren få applikasjonen til å hente blodsukkernivået og benytte dette i sin utregning, samt vise hva det nåværende nivået er.

Navn: Eksportere statistikk

Mål: Målet er å kunne eksportere statistikken som applikasjonen generere til egnet format og mål.

Primær aktør: Bruker

Beskrivelse: Programmet har mulighet for å loggføre antall doser tatt og mengden makronæringsstoffer. Loggen vil kunne bli sendt på epost eller bare lagres på mobilens lagringsområde.

9 Detaljerte brukstilfeller

Navn: Administrere næringstabell

Mål: Målet er å vise, legge til, fjerne og redigere matvarer i en næringstabell.

Aktør: Administrator

Prebetingelser: Databaseserver må være tilgjengelig.

Postbetingelser: Databasen er oppdatert med nye/endrede/slettede data, eller forblir uendret fordi det var et ønske om å kun se på de data som foreligger.

Trigger: Nye eller endrede data fra myndighetene foreligger, eller brukere har fremmet spesielle ønsker.

Normal hendelsesflyt (“Happy Day Scenario”):

1. Dette use-caset begynner når det finnes et behov for å administrere næringstabellen.
2. Administratoren logger seg på databaseserveren.
3. Administrator ønsker å vise matvarene i næringstabellen, og trykker på denne knappen.
4. Deretter vil vedkommende få mulighet til å filtrere på kategori, navn, innhold av makronæringsstoffer og mengde energi (kcal).
5. Når administrator er fornøyd med filtrering, trykker han søk og får returnert data i tabellform.
6. Steg 5 repeteres inntil administrator er tilfredsstilt, og logger deretter av.

Variasjoner (feilsituasjoner og alternativ bruk):

- 2a. Feil ved innlogging - database nede eller login ugyldig.
- 3b. Administrator ønsker å legge til en eller flere nye matvarer i næringstabellen og trykker på denne knappen.
- 4b. Administrator får nå valg om å enten importere inn fra fil, eller legge til manuelt.
- 5b) Administrator får deretter mulighet til å hhv. velge riktig fil fra disk, eller får en rekke tekstbokser for de ulike attributtene for en matvare opp.
- 3c. Administrator ønsker å endre eller slette matvarer i næringstabell, og trykker på denne knappen.
- 5c) 1) Administrator har funnet matvare(r) han vil fjerne eller endre, og velger disse fra en liste
- 5c) 2) Administrator fjerner eller endrer valgte matvare(r): ved endring vil det dukke opp muligheter for å angi nye data.

Navn: Administrere innstillinger

Mål: Målet er for brukeren å få applikasjonen funksjonell etter brukerens personlige behov.

Aktører: Bruker/Foresatt

Prebetingelser: Ingen spesielle.

Postbetingelser: Applikasjonen laster inn innstillingene på nytt.

Trigger: Brukeren ønsker å administrere innstillingene (eventuelt første gangs kjøring av applikasjonen).

Normal hendelsesflyt (“Happy Day Scenario”):

1. Brukeren har behov for å endre sine innstillinger og velger dette fra meny.
2. Brukeren får deretter anledning til å endre korreksjonsverdi og ettermåltidsverdi som brukeren har fått av sin lege. Bruker trykker neste.
3. Neste skjerm viser en liste over språk. Bruker velger etter sine preferanser og trykker neste.
4. I tredje skjerm får brukeren mulighet til å velge hvilken informasjon som skal taes vare på i forbindelse med statistikkgenerering:
makronæringsstoffinnhold i måltider inntatt samt når de spiste, blodsukkernivået når korreksjonsdosene ble utregnet, og det totale antall insulindoser utregnet er valgene (og disse velges vha. avkrysningsbokser). Brukeren velger også her om logging(statistikk) skal være av eller på.
5. Brukeren trykker “Lagre” etter at riktige verdier, språk og statistikkpreferanser er korrekt inntastet.

Variasjoner (feilsituasjoner og alternativ bruk):

- 2, 3 eller 5b. Bruker angres seg og velger “avbryt”. Nye innstillinger forkastes.
- 1c. Bruker har nettopp startet programmet for første gang, og får med ett beskjed om å endre korreksjonsverdien sin. Etter at denne er inntastet og lagret, starter applikasjonen med standardspråk og statistikkgenerering avslått.

Navn: Administrere måltider og matvarer

Mål: Målet er å tilby brukeren en mulighet til å legge til, endre, slette og vise personlige måltider og matvarer.

Aktører: Bruker

Prebetingelser: Ingen spesielle.

Postbetingelser: Bruker er fornøyd med listene.

Trigger: Bruker skal spise noe som ikke ligger i listene, eller ønsker å endre, slette eller vise hvilke måltider og matvarer som er lagret.

Normal hendelsesflyt (“Happy Day Scenario ved ny matvare”):

Bruker vil legge til matvarer

1. Bruker spesifiserer om han vil legge til manuelt eller hente en eller flere matvarer fra databasen med næringstabellen.
- 2.1 Ved manuell innlegging får bruker en rekke tekstbokser å fylle inn, blant annet for navn og mengde energi (kcal) og gram av makronæringsstoffer, og velger kategori fra en liste. Også hvor mye en porsjon tilsvarer må tastes inn.
- 2.2 Velges det å hente fra næringstabellen, får brukeren mulighet til å filtrere på navn og kategori. Resultatet returneres umiddelbart eksternt, og brukeren kan legge til matvaren som er funnet ved å merke den, definere hvor mye som tilsvarer en porsjon, og bruke knappen “legg til”.
3. Steg 2.1 og/eller 2.2 gjentas inntil ønsket resultat er oppnådd, hvorpå brukeren trykker “Lagre”.

Variasjoner (feilsituasjoner og alternativ bruk):

Bruker vil legge til måltider

- 1b. Bruker spesifiserer “legg til måltider”
- 2b. Bruker velger deretter “hent matvarer”, og får opp filtreringsmuligheter på navn og kategori.
- 3b. Bruker velger matvare fra returnert liste, angir hvor mye som tilsvarer en porsjon, og gjentar steg 2b inntil han er tilfredsstillt.

Bruker vil slette eller endre matvarer eller måltider

- 1c. Bruker får mulighet til å fylle ut navn og velge kategori fra en liste for å filtrere.
- 2c1) Bruker markerer matvarer og eller måltider som ønskes slettet, ved å gjenta søk i steg 2c om ønskelig.
- 2c2) Bruker markerer en matvare eller et måltid returnert fra filtreringsprosessen og trykker “endre”. Derfra kan alle attributtene endres. Når ønsket
- 3c. Steg 2c1 og 2c2 gjentas inntil ønsket resultat er oppnådd, hvorpå brukeren trykker “Lagre”

Feilsituasjoner:

- 3abc. Brukeren angre seg, og trykker avbryt. Ingen av endringene lagres.

Navn: Avlese blodsukkernivå

Mål: Målet er å få blodsukkernivået avlest fra en ekstern sensor, være seg insulinpumpe eller en selvstendig enhet, vha. trådløs teknologi.

Primær aktør: Bruker/Sensor

Prebetingelser: Sensor er tilstede.

Postbetingelser: Sensor hviler inntil neste korreksjonsdose.

Trigger: Bruker trykker kalkulér i korreksjonsdosedelen av programmet.

Normal hendelsesflyt (“Happy Day Scenario”):

1. Programmet ber sensoren om å få nåværende blodsukkernivå.
2. Sensor sender verdien trådløst til mobilen(applikasjonen).
3. Verdien blir brukt i korreksjonsdoseformelen.

Variasjoner (feilsituasjoner og alternativ bruk):

- 1b. Ingen sensor svarer på forespørselen.
- 2b. Tilkoblingen er ikke fungerende og ingen verdi blir returnert.

Navn: Kalkulere korreksjonsdose

Mål: Målet er å kalkulere riktig korreksjonsdose med insulin som skal injiseres før et måltid inntas.

Primær aktør: Bruker

Prebetingelser: Forbindelse mellom telefon og sensor må være på plass.

Postbetingelser: Sensor hviler inntil neste korreksjonsdose.

Trigger: Bruker skal beregne sin korreksjonsdose før et måltid inntas.

Normal hendelsesflyt (“Happy Day Scenario”):

1. Bruker navigerer seg frem til “Kalkulér korreksjonsdose”
2. Deretter velges “Kalkulér”, som får sensoren til å avlese blodsukkernivået og returnere dette, og viser det i en tekstboks, og benytter formelen:
“Blodsukkernivå MINUS 5 (konstant) DELT PÅ korreksjonsverdien satt i innstillinger”.
3. Resultattekstboksen oppdateres med riktig antall insulindoser som skal inntas for å korrigere blodsukkeret etter

Variasjoner (feilsituasjoner og alternativ bruk):

2b. Sensoren er ikke aktiv eller tilkoblet eller ikke fungerende, og brukeren får derfor mulighet til å legge inn nåværende blodsukkernivå manuelt. Hvis dette ikke foreligger, trykkes “avbryt”.

Relatert informasjon: Eksempel på utregning: $25 \text{ mmo} / 1 \text{ MINUS } 5 \text{ DELT PÅ } 10$ er lik 2 korreksjonsdoser.

Navn: Kalkulering av insulin doser

Mål: Målet er å kalkulere riktig insulin dose som skal injiseres etter at et måltid er inntatt.

Aktører: Bruker

Prebetingelser: Bruker må ha registrert ettermåltidsverdien.

Postbetingelser: Antallet insulin enheter samt blodsukkeret og makronæringsinnhold havner i statistikken hvis dette er angitt.

Trigger: Bruker skal innta (eller har allerede inntatt) et måltid og skal kompensere for energiinnholdet i maten.

Normal hendelsesflyt (“Happy Day Scenario”):

1. Bruker henvender seg til utregningsdelen av applikasjonen.
2. Bruker legger til eksisterende matvarer eller måltider v.h.a. knapper ved listen. Han kan filtrere etter navn og kategori. Brukeren skal kunne spesifisere selv hvor mange egendefinerte porsjoner av matvaren de skal spise.
3. Når bruker er fornøyd med listen trykker han/hun på “kalkulér”, og antall internasjonale doser vil vises på skjermen. Formelen som brukes er “energiinnhold i måltidet DELT PÅ ettermåltidsverdi (satt i innstillinger)”.

Variasjoner (feilsituasjoner og alternativ bruk):

- 2b. Ingen personlige matvarer eller måltider finnes: Alle knapper er grået ut.
- 3b. Bruker får opplyst feil antall doser og en alvorlig medisinsk situasjon kan inntreffe.

Relatert informasjon:

Det er ikke mulig å trykke på “kalkulér” med mindre det er lagt til matvarer eller måltider i listen. Eksempel på utregning: $170 \text{ (kcal)} / 50 = 3.4 = 3$ doser.

Navn: Eksportere statistikk

Mål: Målet er å kunne eksportere statistikken som applikasjonen generere til egnet format og mål.

Primær aktør: Bruker

Prebetingelser: Loggføring er påslått.

Postbetingelser: Statistikk har blitt sendt til korrekt epost eller rett sted på mobil.

Trigger: Bruker ønsker å importere statistikk i et egnet statistikkprogram (f.eks Excel, OOo Calc).

Normal hendelsesflyt (“Happy Day Scenario”):

1. Bruker entrer “Eksporter statistikk”-menyen.
2. Bruker velger enten å lagre på et bestemt sted på mobilen, eller å få statistikken tilsendt på epost.
- 3a. Bruker navigerer mappestrukturen på brukerens mobil og velger hvor den skal lagres. Loggen lagres.
- 3b. Epostklienten på brukerens mobil åpnes, med en ny blank epost og loggen lagt som vedlegg.

Variasjoner (feilsituasjoner og alternativ bruk):

- 3a. Det er ikke plass på brukerens mobil til å lagre loggen. Feilmelding gis.
- 3c1. Brukeren har ikke noe egnet epostprogram på mobilen, og vil dermed ikke få tilsendt logg.
- 3c2. Brukerens epostoppsett er feil, og brukeren vil dermed ikke få sendt logg.

Relatert informasjon:

At brukeren får tilsendt epost er avhengig av at brukeren på forhånd har satt opp epostklienten på sin egen mobil.

10 Mock-ups

Nedenfor følger mock-ups av hvordan vi ser for oss at utregning av insulindoser etter måltid, og innlegging av matvarer fra næringstabellen kan se ut. Disse er laget vha. programmet *Pencil*.

The screenshot shows a software window titled "DiaSet -> Utregning". It contains a table with the following data:

Navn	Karbohydrater per porsjon	Antall
Tacogryte	140 gram	2
Eple	35 gram	1

Below the table are two buttons: "Legg til matvare" and "Legg til måltid".

Below the buttons is a text input field containing "Blodsukkernivå: 12mmo/l".

Below the input field is a "RESULTAT" label followed by a text box containing "Antall insulinenheter: 4 (FIRE)".

At the bottom is a large button labeled "Kalkulér antall insulinenheter".

Diaset -> Hent matvare

Navn

Kategori

Hent resultater

Hva tilsvare 1 porsjon for deg (gram):

Navn	Karb...	Protein	Fett	Energi
Hel kylling	2	23	30	250
Kyllingvinger	2	20	25	200

Legg til matvare(r)

Navn	Gram per porsjon
Hel kylling	200 gram
Fullkornspasta	50 gram

Lagre valg Avbryt sekvens

11 Kvalitetsmessige og operasjonelle krav

Ytelse og lagring

- Det skal ikke ta lenger enn to sekunder å returnere matvaresøk fra næringstabellen.
- Systemet skal kunne håndtere at 1000 parallelle brukere søker i næringstabellen etter matvarer samtidig
- Næringstabellen skal kunne lagre alle kjente matvarer
- Programmet skal ikke ta uforholdsmessig stor plass (ikke mer enn 10MB)
- Utregningsfunksjonaliteten i programmet skal ha resultatet klart innen fem sekunder etter at bruker

Sikkerhet og personvern

- Det skal ikke tas vare på noen personopplysninger av programmet, hverken under kjøring eller når brukeren henter ned matvarer fra næringstabellen.

Brukervennlighet

- Det skal være støtte for 10 av de mest kjente språkene i verden.
- Programmet skal være så enkelt at 75% av brukerne skal intuitivt eller i løpet av 2 timer være i stand til å benytte seg av all den funksjonalitet som tilbys.

Pålitelighet

- Vi forventer en garantert oppetidsprosent på næringstabelldatabasen på 95%.
- Ved nedetid på næringstabelldatabasen skal denne være online igjen innen ett døgn.

Livssyklus

- Prosjektet skal vedlikeholdes i minst to år etter ferdigstillelse, med mulighet for forlengelse eller reduksjon av denne perioden
- Gitt den inkrementelle oppbyggingen av programmet, skal det gis mulighet til å implementere ny funksjonalitet/nye komponenter med letthet i fremtiden

Nøyaktighet

- Resultatene fra systemets utregninger skal kunne garanteres å være 100% riktige i henhold til brukerens innstillinger og de formler som er definert.
- Minst 98% av verdiene i næringstabellen skal være korrekte
- Programmet skal benytte seg av standardiserte, internasjonale formler for utregningsfunksjonaliteten som tilbys

Portabilitet

- Programmet skal kodes i et programmeringsspråk som gir bred mulighet til å senere porte det til en annen plattform, arkitektur eller system ved behov.
- Det skal uten problemer kunne portes til Android, iPhone og nettbrett

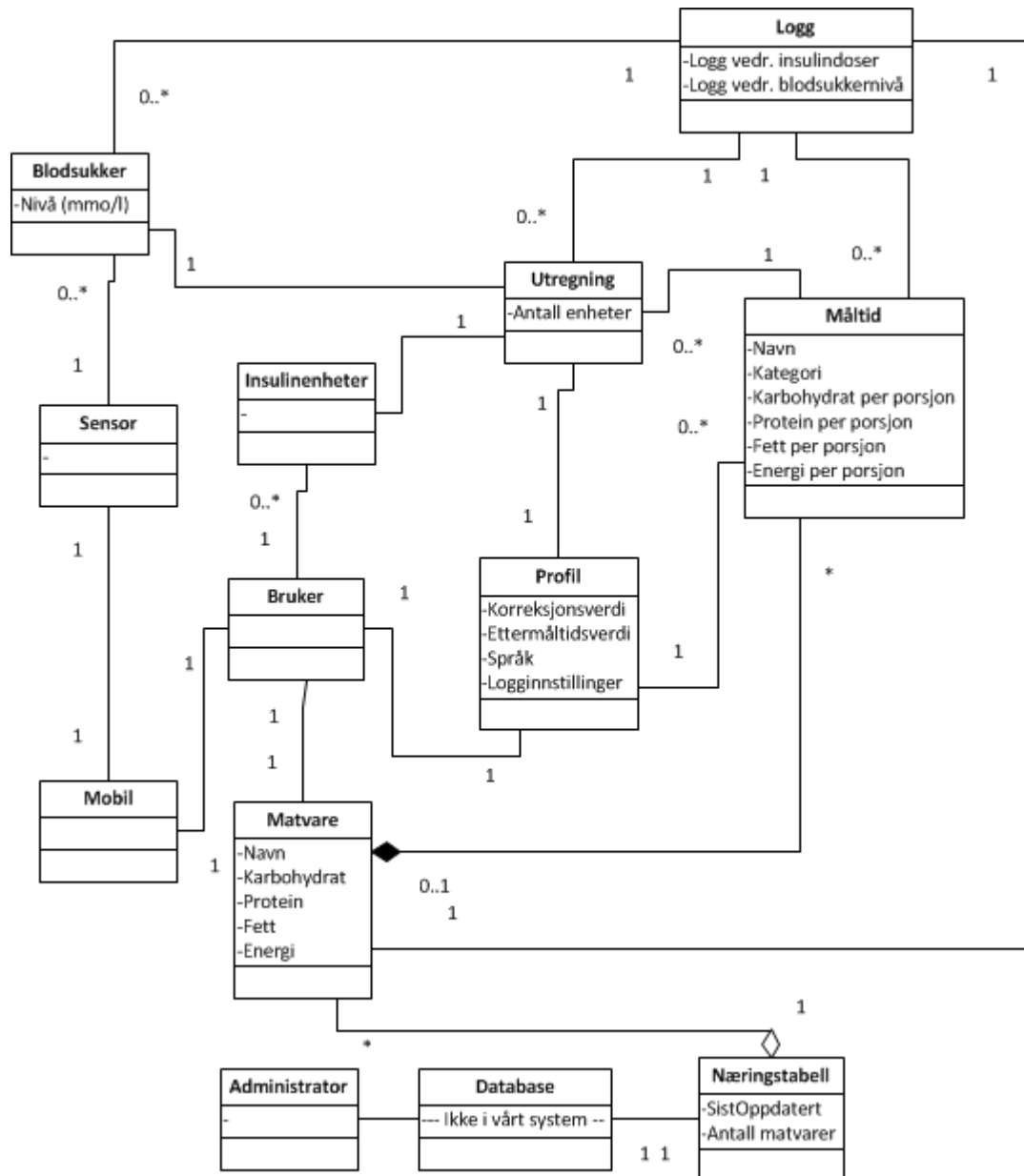
Gjenbruk

- Programmet skal kodes med tanke på fremtiden, slik at noe av, eller hele moduler, med enkelhet kan tas i bruk i annen programvare. Minst 30% av koden skal være mulig å gjenbruke uten større endringer.

Interoperability

- Programmet skal ta i bruk eksterne systemers (sensorers) grensesnitt for å hente data (blodsukkernivå), og vil ikke kunne tilby noen slik funksjonalitet selv.

12 Konseptuelt klassediagram



10.11 Oppgavebeskrivelse: Oppgave 13 - Diaset - insulinkalkulator for Smarttelefon

HiG – Forslag til studentprosjekt 2011-12 - DiaSet - mobil insulinose kalkulator

Oppdragsgiver: Steinar Dalby, GITEK AS
Kontaktperson: Steinar Dalby
Adresse: Teknologiveien 22, 2815 Gjøvik
Telefon: **61178401/95787457**
E-post: sd@gitek.no

Bakgrunnen for prosjektet

Det finnes i dag en del gode hjelpemidler for å måle blodsukkernivået med automatisk overføring av måledata til bla. insulinpumpe, pc og App på mobiltelefon og for visning av graf for utvikling av blodsukkernivået over tid.

Det som er utfordringen er at alle må selv beregne karbohydratmengden og taste resultatet inn på insulinpumpa. Eks. 1 brødskeiv = 15 kcal. Denne informasjonen finnes i dag stort sett i skriftlige tabeller som er tilpasset diabetikere. Det finnes også noen App'er tilgjengelig på App Store. Disse dekker ikke det aktuelle behovet på en tilfredsstillende måte men kan være eksempler på funksjon og grensesnitt.

Oppgaven

Utvikling av App for beregning av karbohydrater i vanlig mat, som kan betjenes av barn fra ca. 7 års alder.

Utviklingen baserer seg på kravspesifikasjon fra Prosjekt "DiaSet" - mobil insulinosekalkulator.

En viktig del av oppgaven vil være å utforme og utvikle et grafisk grensesnitt som er tilpasset en brukergruppe: barn fra 7 års alder på en slik måte at det er "kult" og motiverende.

Grunnlag for utviklingsarbeidet - Studentprosjekt 2010-11

Ei studentgruppe bestående av Hans Åge Martinsen, Erik Grina Raassum og Øyvind Bruvoll utarbeidet en komplett dokumentasjon av en App med de funksjoner som jeg ga innspill til.

Diabetes type 1 er en kronisk autoimmun sykdom hvor personen har høyt blodsukkernivå grunnet relativ eller absolutt insulinmangel. Ved høyt blodsukker eller etter måltider må en type 1 diabetiker injisere insulin for å bringe blodsukkeret tilbake til normale nivåer.

Bakgrunnen for dette prosjektet er at det i dag er veldig mange diabetikere som før hver insulininjeksjon, manuelt må regne ut hvor mange internasjonale enheter insulin de må injisere vha insulinpumpe eller sprøyter for å korrigere blodsukkeret.

Det er ønskelig med en løsning som tar i bruk den nye teknologien i både telefon og medisinsk utstyr for å gjøre livet til type 1-diabetikere lettere.

En hensiktsmessig løsning for å avlaste de rammede bør kunne gjøre følgende:

- Hjelp til med å gi mer korrekte injiseringer.
- Skape trygghet for kjente, kjære og den syke.

Da denne typen diabetikere er helt avhengige av insulin til gitte tider, vil sikkerhet og etterfølgelse av internasjonale standarder være i absolutt høysete.

Programmering

Studentene vil ha innflytelse på valg av teknologi og grensesnitt for den aktuelle modulen.

Om oppdragsgiveren

Jeg er ansatt/medeier i GITEK AS som holder til i lokaler på HiG (over Hig-administrasjonen).

Jeg er far til 2 voksne barn (gutt 36 år og datter 39 år) som begge har hatt Diabetes 1 fra 4 års alderen. Min datter har en sønn som nå er 8 år som også har diabetes 1.

Jeg har altså ca 40 års erfaring med å forvalte et regime sammen med barn under oppvekst og ser derfor både praktiske og psykologiske utfordringer for både foreldre og barn.

Jeg ser for meg at jeg kan bidra med å skaffe testbrukere og eventuelt kontakt med medisinsk ekspertise.

Oppgaven vil passe for en gruppe på 2-3 personer

Jeg har som ansatt i Gitek bistått med å få gjennomført bacheloroppgaver ved HiG fra 1998. De fleste oppgavene har gitt både studentene og oss viktig kunnskap om ny teknologi. Vi har også opparbeidet oss god kunnskap i å samhandle med studentene i prosessen. Det er også kort vei for å besøke oss da vi holder til på Campus.

10.12 Prosjektavtale



HØGSKOLEN I GJØVIK

PROSJEKTAVTALE

mellom Høgskolen i Gjøvik (HiG) (utdanningsinstitusjon),

Steinar Dalby, Vikemøgen 10, 2818 Gjøvik (Gilek 4/5)

(oppdragsgiver), og

Gudbrand Olimb

Michael Urrang

Trond Hagehaugen

(student(er))

Avtalen angir avtalepartenes plikter vedrørende gjennomføring av prosjektet og rettigheter til anvendelse av de resultater som prosjektet frembringer:

1. Studenten(e) skal gjennomføre prosjektet i perioden fra 28/11 til 23/5.

Studentene skal i denne perioden følge en oppsatt fremdriftsplan der HiG yter veiledning. Oppdragsgiver yter avtalt prosjektbistand til fastsatte tider. Oppdragsgiver stiller til rådighet kunnskap og materiale som er nødvendig for å få gjennomført prosjektet. Det forutsettes at de gitte problemstillinger det arbeides med er aktuelle og på et nivå tilpasset studentenes faglige kunnskaper. Oppdragsgiver plikter på forespørsel fra HiG å gi en vurdering av prosjektet vederlagsfritt.

2. Kostnadene ved gjennomføringen av prosjektet dekkes på følgende måte:
 - Oppdragsgiver dekker selv gjennomføring av prosjektet når det gjelder f.eks. materiell, telefon/fax, reiser og nødvendig overnatting på steder langt fra HiG. Studentene dekker utgifter for trykking og ferdigstillelse av den skriftlige besvarelsen vedrørende prosjektet.
 - Eiendomsretten til eventuell prototyp tilfaller den som har betalt komponenter og materiell mv. som er brukt til prototypen. Dersom det er nødvendig med større og/eller spesielle investeringer for å få gjennomført prosjektet, må det gjøres en egen avtale mellom partene om eventuell kostnadsfordeling og eiendomsrett.
3. HiG står ikke som garantist for at det oppdragsgiver har bestilt fungerer etter hensikten, ei heller at prosjektet blir fullført. Prosjektet må anses som en eksamensrelatert oppgave som blir bedømt av faglærer/veileder og sensor. Likevel er det en forpliktelse for utøverne av prosjektet å fullføre dette til avtalte spesifikasjoner, funksjonsnivå og tider.
4. Den totale besvarelsen med tegninger, modeller og apparatur så vel som programlisting, kildekode, disketter, taper mv. som inngår som del av eller vedlegg til besvarelsen, gis det en kopi av til HiG, som vederlagsfritt kan benyttes til undervisnings- og forskningsformål. Besvarelsen, eller vedlegg til den, må ikke nyttes av HiG til andre formål, og ikke overlates til utenforstående uten etter avtale med de øvrige parter i denne avtalen. Dette gjelder også firmaer hvor ansatte ved HiG og/eller studenter har interesser.

Besvarelser med karakter C eller bedre registreres og plasseres i skolens bibliotek. Det legges også ut en elektronisk prosjektbesvarelse uten vedlegg på bibliotekets del av skolens Internett-sider. Dette avhenger av at studentene skriver under på en egen avtale hvor de gir biblioteket tillatelse til at deres hovedprosjekt blir gjort tilgjengelig i papir og nettutgave (jfr. Lov om opphavsrett). Oppdragsgiver og veileder godtar slik

offentliggjøring når de signerer denne prosjektavtalen, og må evt. gi skriftlig melding til studenter og dekan om de i løpet av prosjektet endrer syn på slik offentliggjøring.

5. Besvarelsens spesifikasjoner og resultat kan anvendes i oppdragsgivers egen virksomhet. Gjør studenten(e) i sin besvarelse, eller under arbeidet med den, en patentbar oppfinnelse, gjelder i forholdet mellom oppdragsgiver og student(er) bestemmelsene i Lov om retten til oppfinnelser av 17. april 1970, §§ 4-10.
6. Ut over den offentliggjøring som er nevnt i punkt 4 har studenten(e) ikke rett til å publisere sin besvarelse, det være seg helt eller delvis eller som del i annet arbeide, uten samtykke fra oppdragsgiver. Tilsvarende samtykke må foreligge i forholdet mellom student(er) og faglærer/veileder for det materialet som faglærer/veileder stiller til disposisjon.
7. Studenten(e) leverer oppgavebesvarelsen med vedlegg (pdf) i Fronter. I tillegg leveres et eksemplar til oppdragsgiver.
8. Denne avtalen utferdiges med et eksemplar til hver av partene. På vegne av HiG er det dekan/prodekan som godkjenner avtalen.
9. I det enkelte tilfelle kan det inngås egen avtale mellom oppdragsgiver, student(er) og HiG som nærmere regulerer forhold vedrørende bl.a. eiendomsrett, videre bruk, konfidensialitet, kostnadsdekning og økonomisk utnyttelse av resultatene.

Dersom oppdragsgiver og student(er) ønsker en videre eller ny avtale, skjer dette uten HiG som partner.

10. Når HiG også opptrer som oppdragsgiver trer HiG inn i kontrakten både som utdanningsinstitusjon og som oppdragsgiver.
11. Eventuell uenighet vedrørende forståelse av denne avtale løses ved forhandlinger avtalepartene i mellom. Dersom det ikke oppnås enighet, er partene enige om at tvisten løses av voldgift, etter bestemmelsene i tvistemålsloven av 13.8.1915 nr. 6, kapittel 32.
12. Deltakende personer ved prosjektgjennomføringen:

HiGs veileder (navn): Harald Liødden

Oppdragsgivers
kontaktperson (navn): Steinar Dalby

Student(er) (signatur): Tomil Flagehaugen dato 24/1-12

Michael Urang dato 24/1-12

Guðmund G. Ólafur dato 24/1-12

dato _____

Oppdragsgiver (signatur): Steinar Dalby dato 24/1-12

IMT Dekan/prodekan (signatur): [Signature] dato 15/2-2012