

Kolonihageprosjektet

Kandidatene 60, 142, 265, 317 og 381
<http://heim.if.uio.no/sigvei/kolonistbasen/>

23. mai 2005

Innhold

| | | |
|----------|-------------------------------------------------------------|-----------|
| I | Prosjektbeskrivelse | 3 |
| 1 | Prosjektbeskrivelse | 4 |
| 1.1 | Virksomheten – Norsk Kolonihageforbund | 4 |
| 1.2 | Systemet | 4 |
| 1.3 | Formålet med prosjektet | 5 |
| 1.4 | Interesseområdet | 5 |
| 1.5 | Konsekvenser | 5 |
| 1.6 | Brukere og funksjonalitet | 6 |
| 1.7 | Fremtidige utvidelser | 6 |
| 2 | Bruksmønstre | 8 |
| 2.1 | Generelt om bruksmønstrene | 8 |
| 2.2 | Bruksmønstrene | 9 |
| 2.2.1 | Registrer nytt medlem | 9 |
| 2.2.2 | Registrer ny parsell | 9 |
| 2.2.3 | Endre medlem | 10 |
| 2.2.4 | Endre parsell | 10 |
| 2.2.5 | Vis medlemmer | 11 |
| 2.2.6 | Vis parseller | 11 |
| 2.2.7 | Vis søkerliste | 12 |
| 2.2.8 | Vis informasjon om seg selv | 12 |
| 2.3 | Analyse av et bruksmønster | 14 |
| 2.3.1 | Differanse fra det opprinnelige klassesdiagrammet | 14 |
| 3 | Realisering i databasen | 15 |
| 3.1 | Avklaringer av mulige uklarheter | 15 |
| 3.2 | SQL-spørringer | 16 |
| 3.2.1 | Spørringer for å opprette databasen | 17 |
| 3.2.2 | SELECT * på alle tabellene i databasen | 17 |
| 3.2.3 | Oppslag som viser primitiv funksjonalitet | 18 |
| 4 | Brukerdokumentasjon | 19 |
| 4.1 | Innsyn for vanlige medlemmer | 19 |
| 4.2 | Pålogging for administratorer | 20 |
| 4.3 | Hovedmenyen | 20 |
| 4.3.1 | Medlemmer | 20 |
| 4.4 | Parseller | 24 |
| 4.4.1 | Ny parsell | 24 |
| 4.4.2 | Endre parsell | 24 |
| 4.4.3 | Vis parseller | 24 |
| 4.5 | Oversikter | 24 |
| 4.5.1 | Statistikk | 24 |

| | | |
|------------|----------------------------------------------------------------|-----------|
| 4.5.2 | Vis ventelister | 24 |
| II | Utviklingsprosessen | 30 |
| 5 | Prosessen | 31 |
| 6 | Jus og etikk | 33 |
| 7 | Gruppedynamikk | 35 |
| 8 | Endringer som følge av evalueringen | 36 |
| III | Evaluering av PIMEDAB | 40 |
| 9 | Evaluering av prosjektrapporten | 42 |
| 9.1 | Konsistens mellom gruppert og ugruppert UML -diagram. | 42 |
| 9.1.1 | Den ugrupperte modellen | 42 |
| 9.1.2 | Konsistens med gruppert modell | 42 |
| 9.2 | Konsistens mellom UML-diagram og SQL-statements | 43 |
| 9.2.1 | Piktogrammer | 43 |
| 9.2.2 | Bildefiler | 43 |
| 9.2.3 | Kategorier | 43 |
| 9.2.4 | Hovedkategorier | 43 |
| 9.2.5 | Kobling | 43 |
| 9.2.6 | Om assosiasjonsmultiplisiteter | 44 |
| 9.3 | Sammenheng mellom bruksmønstre og de overordnede mål | 44 |
| 9.4 | Sekvensdiagram og objektorientert klassediagram | 44 |
| 9.5 | Brukerveiledningen | 45 |
| 10 | Evaluering av systemet | 47 |
| 10.1 | Hente frem språktegn som omhandler mat | 47 |
| 10.1.1 | Utførelse | 47 |
| 10.1.2 | Funksjonalitet | 47 |
| 10.1.3 | Kontroll | 48 |
| 10.1.4 | Konklusjon | 48 |
| 10.2 | Legge inn et nytt piktogram | 48 |
| 10.2.1 | Utførelse | 48 |
| 10.2.2 | Funksjonalitet | 48 |
| 10.2.3 | Kontroll | 48 |
| 10.3 | Endre et piktogram | 49 |
| 10.3.1 | Utførelse | 49 |
| 10.3.2 | Funksjonalitet | 49 |
| 10.3.3 | Kontroll | 49 |
| 10.4 | Konsistens i brukergrensesnitt | 49 |
| 10.5 | Konsistens mellom brukerveiledning og systemet | 50 |

URL: <http://heim.ifi.uio.no/sigvei/kolonistbasen/>

Innlogging: Brukernavn 'admin', passord 'sikkerStreng'

Kildekode: Kan vises gjennom <http://eksploder.mine.nu/websvn/wsvn/kolonistbasen/tags/release/>, eller lastes ned som en zip-fil fra <http://heim.ifi.uio.no/sigvei/kolonistbasen/kildekode.zip>

Del I

Prosjektbeskrivelse

Kapittel 1

Prosjektbeskrivelse

1.1 Virksomheten – Norsk Kolonihageforbund

Norsk Kolonihageforbund er en ikke-kommersiell organisasjon, opprettet for å tjene medlemshagenes interesser i forhold til offentlige myndigheter, samt fungere som et bindeledd mellom hagene og publikum. Forvaltningsansvaret påligger imidlertid den enkelte hage, innenfor de retningslinjer som er fastsatt i samarbeid med myndighetene og forbundet sentralt. Hagene er stort sett organisert med et styre, samt ulike komiteer som rapporterer til styret. Komiteene har avgrensede ansvarsområder. Det er med andre ord en relativt stor del av hagens beboere som er engasjert i ulike tillitsverv. Alle verv besettes gjennom årsmøtets valg, og alle hagens beboere har møte- og stemmerett på årsmøtet.

Forbundets og hagenes aktiviteter er helfinansiert gjennom medlemskontingent, og midler til bruk av kommersielle systemutviklere er svært begrenset.

1.2 Systemet

En av prosjektgruppens medlemmer sitter for tiden i styret i en av Oslos kolonihager. Prosjektets målsetning er basert på hans kjennskap til kolonihagens rutiner for fordeling av ledige parseller, samt utfordringer som har kommet frem i møte med en av hagestyrene og den sentrale administrasjonen.

Tildeling av parseller til nye medlemmer administreres lokalt av den enkelte kolonihage, basert på ventelister. Imidlertid skjer tilmelding til ventelisten via kolonihageforbundet sentralt. Nye interessenter sender sin søknad inn til forbundet, og forbundet sender så ut (på papir) årlige lister med nye søkere til den enkelte hage. Det er da opp til hagene selv å oppdatere og administrere den lokale databasen som utgjør grunnlaget for tildeling av parseller til nye medlemmer. For dette formål benyttes i dag et system som er 100% proprietært, fullstendig uten mulighet for import eller eksport av elektroniske lister. Det finnes heller ingen mulighet for å hente ut eller legge inn data fra andre maskiner enn den databasen ligger på.

Når personer på venteliste ønsker å få rede på sin status (om de står på listen, hvor lang ansiennitet de har osv.) ringer de gjerne til styrets leder eller sekretær, som så må til styrets kontor for å gjøre et manuelt oppslag som så formidles til den håpefulle. Med andre ord et meget arbeidskrevende, tungvindt og lite fleksibelt system.

I kontakt med kolonihageforbundet ga de sterkt uttrykk for at systemet måtte være enkelt i bruk og kreve minimal opplæring, ettersom brukerne i stor grad vil være frivillige med lav datakompetanse.

1.3 Formålet med prosjektet

Prosjektets formål er å erstatte et eksisterende utdatert, upraktisk og lite fleksibelt system med et åpent system basert på PHP-skript og en SQL-database. For å gjøre prosjektet skalerbart vil det være en fordel å definere følgende målsetninger i prioritert orden:

- Erstatte dagens system med en åpen og fleksibel klient/tjener-basert løsning som kan utvides i takt med nye behov.
- Opprette ny funksjonalitet som kan forenkle registrering, behandling, oppslag og utskrift av data for kolonihagene.
- Opprette et offentlig grensesnitt som kan gjøre informasjonen tilgjengelig for allmennheten.
- Opprette grensesnitt mot andre, sentrale IT-systemer av relevans, for slik å automatisere vedlikehold av databasen.

1.4 Interesseområdet

Systemet vil befatte seg med og modellere følgende deler av virkeligheten:

- Personer som vurderer å sette sitt navn på ventelisten.
- Personer som allerede står på ventelisten.
- Den sentrale administrasjon med ansvar for kretsens/forbundets rutiner og IT-ressurser.
- Kolonihager.
- Parseller i kolonihager.

Den lokale hagen må differensieres i styrets sekretær (som har ansvar for vedlikehold av databasen) og salgskomiteen (som benytter databasen som beslutningsgrunnlag ved salg av ledige parseller). Funksjonaliteten skal være den samme for begge aktørene, og systemet gjør derfor ikke forskjell på sekretær og salgskomite i denne omgang.

1.5 Konsekvenser

Hvilke konsekvenser systemet vil kunne få hvis det realiseres og settes i drift avhenger av hvilket omfang prosjektet vil vise seg å gi rom for.

Først og fremst vil det kunne erstatte et utdatert og lite fleksibelt system med en utvidbar løsning basert på åpne systemer. Dette vil uansett være en fordel med tanke på fremtidig bruk av systemet.

Videre vil det åpne for en mer fleksibel tilgang til systemet, ved at det kan aksesserer via en webløsning. Dette vil gi brukerne økt fleksibilitet med hensyn til lokasjon, samt muliggjøre sentralisert administrasjon og vedlikehold.

Registrering av nye data i databasen vil forenkles med funksjonalitet for import av elektroniske lister. Dette medfører en betydelig arbeidsbesparelse for de tillitsvalgte som har ansvar for denne oppgaven.

Med et offentlig grensesnitt mot databasen vil publikum selv kunne sjekke sin egen og andres status på ventelisten, uten å måtte gå via hagens sekretær eller styreleder. Dette gir enklere tilgang til informasjonen for publikum, samt en arbeidsbesparelse for de tillitsvalgte.

Med et sanntidsgrensesnitt mot de sentrale medlemsregistre vil informasjonen kunne oppdateres fortløpende, i takt med endringer i den sentrale databasen. Slik vil dagens til dels lange latenstid mellom tilmelding og oppdatering bli historie.

Dersom systemet blir vellykket, storstilt og tas i bruk i alle Oslos kolonihager, vil det derfor kunne redusere tiden de tillitsvalgte bruker på papirarbeid, og øke tiden de bruker på hagene. Tilsvarende kan kortere latenstid for tilmelding og oppdatering gjøre at det blir enklere, raskere og mer attraktivt å skaffe seg en kolonihage.

1.6 Brukere og funksjonalitet

Brukergruppene vil i en stor grad være sammenfallende med listen under Interesseområdet. Funksjonaliteten som tilbys hver gruppe vil imidlertid være ulik.

Personer som vurderer å sette sitt navn på ventelisten vil primært være interessert i å se hvor mange som allerede står på den enkelte venteliste, samt hvilke hager som har kortest venteliste. I tillegg vil det kunne være av interesse å se hvor lenge interessentene til de ulike hagene har stått på ventelisten.

Personer som allerede står på ventelisten vil kunne benytte systemet til å verifisere at de faktisk fortsatt står på ventelisten, hvilken ansiennitet de er kreditert med, samt hvor lang ansiennitet deres konkurrenter har.

Den sentrale administrasjon vil benytte systemet til å legge inn nye innmeldinger og slette gamle, samt hente ut informasjon fra systemet til fakturagrunnlag etc.

Den lokale hage vil ha et todelt behov. Styrets sekretær vil primært benytte systemet til registrering av nye medlemmer på ventelisten, samt sletting av dem som ikke lenger ønsker å stå på listen. Salgskomiteen vil bruke systemet til å hente ut informasjon om aktuelle kandidater ved tildeling av ledige parseller.

1.7 Fremtidige utvidelser

Det er en rekke muligheter for utvidelser av systemet, men som innenfor rammene av dette prosjektet ble for omfattende. Vi nevner noen mulige utvidelser her.

Funksjon for å slette medlemmer. I fare for at brukere som er logget inn som administrator i systemet foretar en sletting av et medlem som ikke var med hensikt, har vi valgt å ikke implementere denne funksjonaliteten. Det er derimot fullt mulig å foreta en sletting av et medlem med en SQL kommando direkte til databasen. Ved en utvidelse av systemet vil en

slik funksjonalitet med begrenset brukertilgang være nærliggende. Dette vil også være et juridisk/etisk krav.

Salgskomiteen. Vi har som nevnt ikke gjort forskjell på funksjonene som er tilgjengelig for sekretærene og salgskomiteen. Det kan imidlertid være et ønske om å skille funksjonaliteten for disse gruppene i fremtiden.

Medlemsnummer. Det er en potensiell utvidelse at medlemsnummer genereres av systemet. I øyeblikket har vi tatt utgangspunkt i at format og tildeling av medlemsnummer ligger utenfor systemet (skjer manuelt). Oppdragsgiver ønsket heller ikke en slik funksjonalitet i første omgang.

“Vis venteliste”-funksjon for allmennheten. Det må tas hensyn til personvernloven når det gjelder å gjøre slike data tilgjengelig for allmennheten. Men det er en mulig utvidelse av systemet å gjøre mer informasjon av verdi for potensielle søkere tilgjengelig. I denne omgang tar vi sikte på at systemet kan tilby aggregert og generelle statistikk-opplysninger.

Validering. Innenfor rammene av dette prosjektet vil vi ikke ha kapasitet til å ta oss av en del forbedringer knyttet til utseende og validering av html-sider. Det må derimot gjøres en innsats på dette området ved en fremtidig anledning.

Oppdelt informasjon. Det kan på et senere tidspunkt bli aktuelt å dele opp "info"-feltet om medlemmer, slik at registreringsskjemaet i systemet likner søknadsskjemaet med mer detaljerte tekstfelder. Hva en slik utvidelse eventuelt skal inneholde blir helt opp til oppdragsgiver.

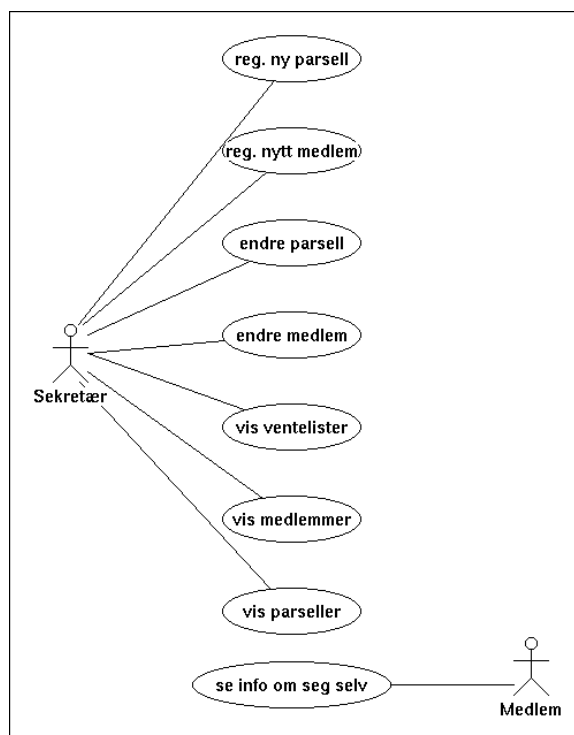
Forøvrig kan det nevnes at vi i dette systemet legger opp til å bruke mest mulig variabler fremfor konstanter, for å sikre rom for fremtidige endringer av sentrale begreper i systemet.

Kapittel 2

Bruksmønstre

2.1 Generelt om bruksmønstrene

Figur 2.1: Bruksmønsterdiagram (use-case)



Vi har valgt ut åtte bruksmønstre (jf. figur 2.1), som til sammen tilfredsstiller formålet definert i kapittel 1. Den primære aktøren, Sekretær, utgjør både den sentrale administrasjon og den lokale hage. Selv om det er mulig å gjøre et teoretisk skille på disse to aktørene, ønsket ikke kolonihageforbundet et slikt skille på det nåværende tidspunkt.

Gjennom å kunne behandle og vise (søke etter) medlemmer og parseller, samt vise ventelister, oppnår systemet den funksjonaliteten som kreves for å oppnå punktene 1 og 3 i formålet med prosjektet (seksjon 1.3 på side 5).

2.2 Bruksmønstrene

2.2.1 Registrer nytt medlem

Aktør: Sekretær

Prebetingelse: Sekretær har mottatt et korrekt utfylt søknadsskjema og er logget inn på systemet

Postbetingelse: Det nye medlemmet er registrert i systemet

Trigger: Nytt medlem har sendt inn søknadsskjema

Normal hendelsesflyt:

1. Sekretær skriver inn tilordnet medlemsnummer, kontaktadresse og annen relevant informasjon
2. Systemet sjekker at medlemsnummeret ikke eksisterer fra før, lagrer dataene og viser skjermbildet for å tilordne personer til medlemsnummeret. Systemet oppdaterer søkerlisten i henhold til registrerte data.
3. Sekretær legger inn navn, fødselsdato og annen relevant informasjon om personen
4. Systemet sjekker at personen ikke finnes fra før, lagrer dataene og viser skjermbildet for en ny person
5. Sekretær gjentar prosessen fra punkt 4 til alle personer som skal tilordnes medlemsnummeret er registrert

Variasjoner:

2a. Medlemsnummeret eksisterer fra før

1. Systemet viser feilmelding
2. Sekretær tilordner medlemmet et annet nummer eller avslutter

4a Personen finnes fra før

1. Systemet viser feilmelding
2. Sekretær velger en annen person eller avslutter

2.2.2 Registrer ny parsell

Aktør: Sekretær

Prebetingelse: Sekretær er logget inn på systemet

Postbetingelse: Den nye parsellen er registrert i systemet

Trigger: Det eksisterer en parsell som ikke er registrert i databasen

Normal hendelsesflyt:

1. Sekretær skriver inn tilordnet parsellnummer, hagenavn og annen relevant informasjon
2. Systemet sjekker at parsellen ikke eksisterer fra før og lagrer dataene

Variasjoner:

2a. Parsellen eksisterer fra før

1. System viser feilmelding
2. Sekretær sjekker grunnlagsdata

2.2.3 Endre medlem

Aktør: Sekretær

Prebetingelse: Sekretær er logget inn på systemet

Postbetingelse: Endringene er registrert i systemet

Trigger: Sekretær har mottatt endringsinformasjon

Normal hendelsesflyt:

1. Sekretær skriver inn medlemsnummer
2. Systemet henter eksisterende informasjon om medlemmet og fyller disse inn i registreringsbildet
3. Sekretær endrer aktuell informasjon om medlemmet
4. Systemet lagrer oppdatert informasjon

Variasjoner:

2a. Medlemsnummeret eksisterer ikke

1. Systemet viser feilmelding
2. Sekretær skriver inn et annet medlemsnummer eller avslutter

2.2.4 Endre parsell

Aktør: Sekretær

Prebetingelse: Sekretær er logget inn på systemet

Postbetingelse: Endringene er registrert i systemet

Trigger: Sekretær har mottatt endringsinformasjon

Normal hendelsesflyt:

1. Sekretær velger hage og skriver inn parsellnummer
2. Systemet henter eksisterende informasjon om parsellen og fyller disse inn i registreringsbildet
3. Sekretær endrer aktuell informasjon om parsellen
4. Systemet lagrer oppdatert informasjon

Variasjoner:

2a. Parsellen eksisterer ikke

1. System viser feilmelding
2. Sekretær søker etter en annen parsell eller avslutter

2.2.5 Vis medlemmer

Aktør: Sekretær

Prebetingelse: Sekretær er logget inn på systemet

Postbetingelse: N/A

Trigger: Sekretær ønsker å hente ut informasjon om medlemmer fra databasen

Normal hendelsesflyt:

1. Sekretær velger utvalgsriterier
2. Systemet henter informasjon om medlemmer basert på utvalgsriteriene, og viser resultatet i en tabell

Variasjoner:

2a. Sekretær ønsker å se detaljert informasjon om ett medlem

1. Sekretær velger et medlem fra tabellen
2. Systemet henter detaljert informasjon om medlem

2.2.6 Vis parseller

Aktør: Sekretær

Prebetingelse: Sekretær er logget inn på systemet

Postbetingelse: N/A

Trigger: Sekretær ønsker å hente ut informasjon om parseller fra databasen

Normal hendelsesflyt:

1. Sekretær velger utvalgsriterier
2. Systemet henter informasjon om parseller basert på utvalgsriteriene, og viser resultatet i en tabell

Variasjoner:

2a. Sekretær ønsker å se detaljert informasjon om en parsell

1. Sekretær velger en parsell fra tabellen
2. Systemet henter detaljert informasjon om parsell

2.2.7 Vis søkerliste

Aktør: Sekretær

Prebetingelse: Sekretær er logget inn på systemet

Postbetingelse: N/A

Trigger: Sekretær ønsker å hente ut informasjon om medlemmer på en venteliste

Normal hendelsesflyt:

1. Sekretær velger hage og sorteringskriterier
2. Systemet viser søkerlisten basert på sorteringskriteriene i en tabell

Variasjoner:

2a. Sekretær ønsker å se detaljert informasjon om ett medlem på søkerlisten

1. Sekretær velger et medlem fra tabellen
2. Systemet henter detaljert informasjon om medlem

2.2.8 Vis informasjon om seg selv

Aktør: Medlem

Prebetingelse: Medlem kjenner sitt medlemsnummer

Postbetingelse: N/A

Trigger: Medlem ønsker å se hvilke informasjon som er lagret om seg selv i databasen

Normal hendelsesflyt:

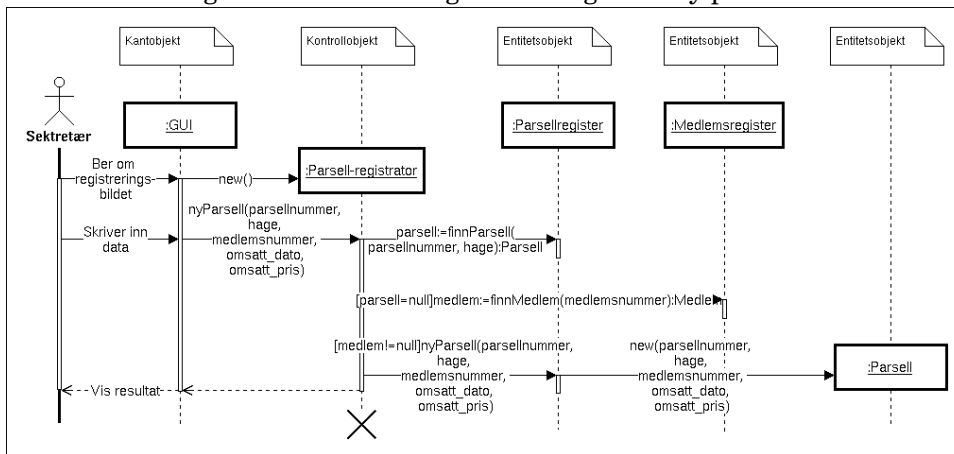
1. Medlem skriver inn sitt navn og medlemsnummer
2. Systemet henter eksisterende informasjon om medlemmet og fyller disse inn i visningsbildet

Variasjoner:

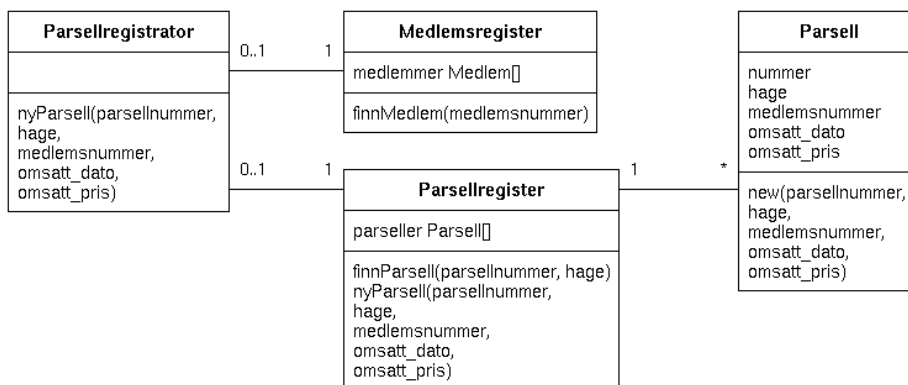
2a. Kombinasjonen av medlemsnummer og navn eksisterer ikke i databasen

1. System viser feilmelding
2. Medlem retter navn, medlemsnummer eller avslutter

Figur 2.2: Sekvensdiagram for registrer ny parsell



Figur 2.3: Klassediagram for registrer ny parsell



2.3 Analyse av et bruksmønster

Vi vil nå ta for oss et sekvens- og klassediagram for ett utvalgt bruksmønster. Vi har valgt ut bruksmønsteret beskrevet i punkt 2.2.2, “Registrer ny parsell”. Sekvensdiagrammet (figur 2.2) viser normal hendelsesflyt for dette bruksmønsteret, og det tilsvarende klassediagrammet (figur 2.3) viser klassene som sekvensdiagrammet benytter seg av.

Metoden `New()` med parametere kan virke unødvendig. Dette diagrammet hviler på at det skal opprettes en ny parsell og vi ønsker å vise dette på en slik måte at det ikke foreligger tvil om hva konstruktøren har til hensikt å gjøre, og det er å opprette et nytt objekt (parsell) som skal inneholde relevante data.

2.3.1 Differanse fra det opprinnelige klassediagrammet

Det opprinnelige klassediagrammet som er grunnlag for databasestrukturen (figur 3.2) avviker på flere punkter fra sekvensdiagrammet og dets medfølgende klassediagram. Det mest åpenbare er at det finnes to klasser i de nye diagrammene som ikke finnes i klassediagrammet; nemlig “Medlemsregister” og “Parsellregister”.

Mens det i ren objektorientert programmering ville være naturlig å ha slike klasser og la de inneholde aggregater av medlemmer og parseller, blir dette lite meningsfylt i en databaseorientert struktur. Tabellen `Medlem` inneholder nettopp alle medlem-objekter i systemet, og det er ingen grunn til å (i databasestrukturen) innføre noen utenforliggende struktur ut over denne tabellen. En slik struktur vil imidlertid kunne innføres i PHP-programmeringen, som en abstraksjon for tabellen `medlem`. Det samme gjelder `parsell`. `Medlemsregister`-objektet i sekvensdiagrammet representerer derfor et logisk lag som i implementasjonen vil bestå av databasemotoren og applikasjonslaget i fellesskap.

Hage er ikke tatt med i vår sekvensmodell. Dette er også en klassisk forskjell på databaseorientert og objektorientert programmering. Mens det i objektorientert programmering vil være naturlig at hvert hage-objekt inneholder et aggregat av tilhørende parseller, er det stikk motsatt i databasemodellen. I databasemodellen er det hver parsell som gjennom fremmednøkkelen holder styr på hvilken hage den tilhører. Dermed er det ikke noe behov for å spørre et hage-objekt hvilke parseller det inneholder. Vi kan gå rett til parsellene og spørre hvem av dem som tilhører en bestemt hage.

`Søkerliste`, `person` og `medlem` er ikke tatt med i sekvens- eller klassediagrammet, fordi de ikke spiller noen aktiv rolle i sekvensdiagrammet. De er langt på vei erstattet av “`Medlemsregister`”.

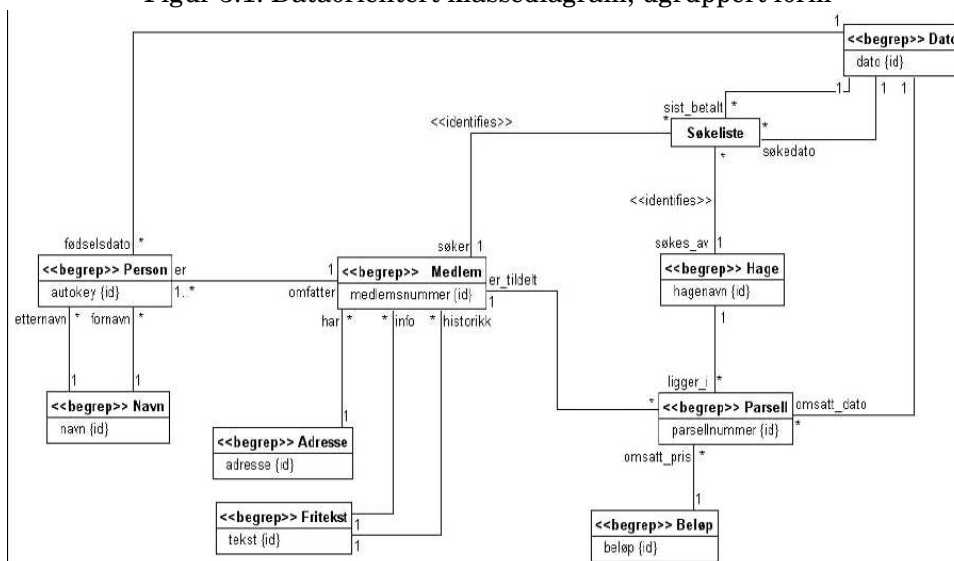
`Parsellregistrator` er den siste klassen vi har inkludert. Den er ikke et datastrukturobjekt i det hele tatt, og vil bli implementert på applikasjonslaget.

Kapittel 3

Realisering i databasen

Disse bruksmønstrene og prosjektets interesseområde utgjør grunnlaget for den databaseorienterte implementasjonen vi har valgt. Klassediagrammet på ugruppert form vises i figur 3.1, mens det grupperte vises i figur 3.2.

Figur 3.1: Dataorientert klassediagram, ugruppert form



3.1 Avklaringer av mulige uklartheter

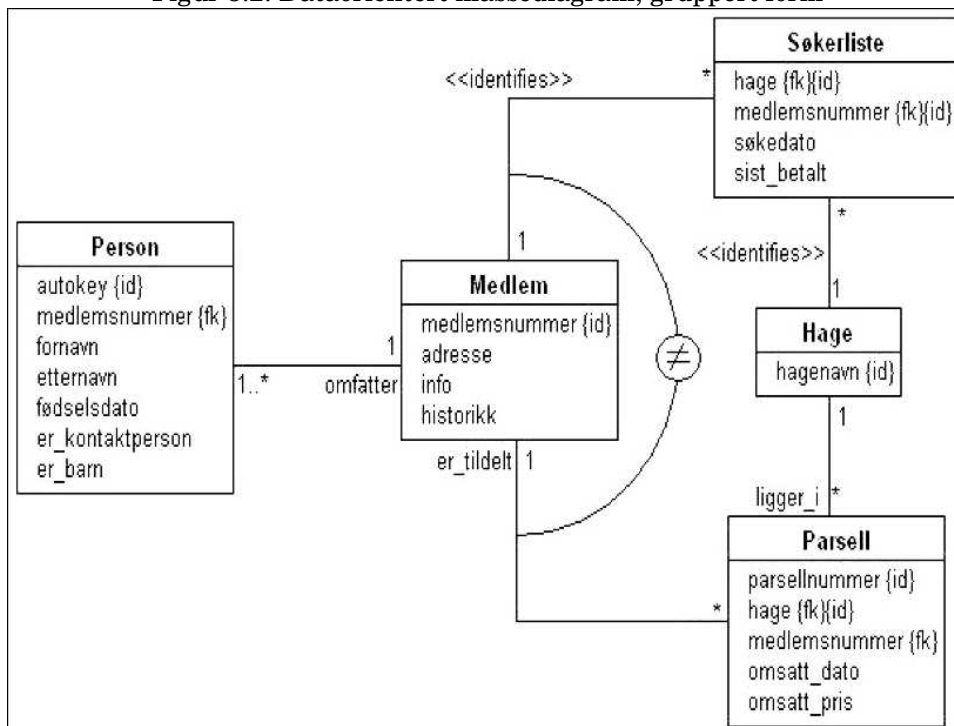
Dette er klargjøringer av mulige uklare feltnavn i databasemodellen.

Person.autokey: Automatisk generert nøkkel for å skille person objekter fra hverandre.

Person.er_kontaktperson: Kun en person av husholdningen kan være listet opp som kontaktperson.

Person.er_barn: Søkere med barn under 16 år prioriteres når parseller skal tildeles. Barn av husstanden kan være i alle aldre (også voksne), vi har derfor valgt å lagre dette som egen post.

Figur 3.2: Dataorientert klassediagram, gruppert form



Medlem.medlemsnummer: Et medlemsnummer inneholder hele husholdningen (Dvs. Fra en til flere personer).

Medlem.info: Inneholder annen informasjon som kan være relevant. Her vil spørsmålene på nedre del av søknadsskjemaet, og eventuelt annen informasjon av relevans legges. Se søknadsskjemaet. <http://www.kolonihager.no/kolonihageforbundet/pdf/soeknadsskjema.pdf>.

Medlem.historikk: Opplysninger om medlemmet som for eksempel om de ikke har møtt på visninger eller lignende. Informasjonen her skal blant annet brukes til å sile ut medlemmer som tilsynelatende ikke er interessert lenger.

Parsell.hage: navn på hage som parsell ligger i

Parsell.medlemsnummer: Medlemsnummer til de som leier parsellen. Databasen er medlemsorientert, og en parsell er derfor kun av interesse for databasen om den er tildelt et medlem.

Parsell.omsatt_dato: Når parsellen sist ble solgt omsatt_pris: Hvor mye parsellen sist ble solgt for

Søkerliste.sist_betalt: Sist betalt medlemskontingent/søkeravgift.

3.2 SQL-spørringer

Her er en rekke SQL-spørringer og resultatene av disse når de brukes mot databasen.

3.2.1 Spørringer for å opprette databasen

Listing 3.1: tabeller.sql

```
CREATE TABLE Person (
  autokey          INTEGER          NOT NULL,
  medlemsnummer   CHAR(8)          NOT NULL,
  fornavn         VARCHAR(100)     ,
  etternavn       VARCHAR(100)     NOT NULL,
  fødselsdato     DATE             NOT NULL,
  er_kontaktperson CHAR(1)        NOT NULL,
  er_barn         CHAR(1)          NOT NULL,
  CONSTRAINT entydig_person
    PRIMARY KEY (autokey)
);

CREATE TABLE Medlem (
  medlemsnummer   CHAR(8)          NOT NULL,
  adresse         VARCHAR(500)     NOT NULL,
  info            VARCHAR(4000)    ,
  historikk       VARCHAR(4000)    ,
  CONSTRAINT entydig_medlem
    PRIMARY KEY (medlemsnummer)
);

CREATE TABLE Hage (
  hagenavn        VARCHAR(100)     NOT NULL,
  CONSTRAINT entydig_hagenavn
    PRIMARY KEY (hagenavn)
);

CREATE TABLE Søkerliste(
  hage            VARCHAR(100)     NOT NULL,
  medlemsnummer   CHAR(8)          NOT NULL,
  søkedato        DATE             ,
  sist_betalt     DATE             ,
  CONSTRAINT entydig_søkerliste
    PRIMARY KEY (hage, medlemsnummer)
);

CREATE TABLE Parsell (
  parsellnummer   CHAR(4)          NOT NULL,
  hage            VARCHAR(100)     NOT NULL,
  medlemsnummer   CHAR(8)          NOT NULL,
  omsatt_dato     DATE             ,
  omsatt_pris     INTEGER          ,
  CONSTRAINT entydig_parsell
    PRIMARY KEY (parsellnummer)
);

ALTER TABLE Person
  ADD CONSTRAINT entydig_medlemsperson
    FOREIGN KEY (medlemsnummer) REFERENCES Medlem ;

ALTER TABLE Person
  ADD CONSTRAINT unik_person
    UNIQUE(fornavn, etternavn, fødselsdato, medlemsnummer);

ALTER TABLE Søkerliste
  ADD CONSTRAINT entydig_medlems søker
    FOREIGN KEY (medlemsnummer) REFERENCES Medlem ;

ALTER TABLE Parsell
  ADD CONSTRAINT entydig_medlemsparsell
    FOREIGN KEY (medlemsnummer) REFERENCES Medlem ;

ALTER TABLE Søkerliste
  ADD CONSTRAINT entydig_hagesøker
    FOREIGN KEY (hage) REFERENCES Hage ;

ALTER TABLE Parsell
  ADD CONSTRAINT entydig_hageparsell
    FOREIGN KEY (hage) REFERENCES Hage ;
```

3.2.2 SELECT * på alle tabellene i databasen

Listing 3.2: select_from.sql

```
SQL> select * from Person;
autokey medlem fornavn      etternavn      fødselsdato    kontakt barn
-----
1      1234  Solveig      Slakte        11-JAN-67     1      0
2      1234  Sleikulf     Slakte        11-JUN-67     0      0
3      1234  Hans         Slakte        12-APR-83     0      1
4      4321  Brummund    Dal           11-JAN-50     1      0
5      4321  Roms        Dal           11-JAN-50     0      0
6      4321  Gaus        Dal           11-JAN-50     0      0
11     6543  Trond       Trondsen     01-DEC-49     1      0
12     8765  Pål         Askeladd     19-MAY-69     1      0
7      2345  Gordon     Frimann      07-APR-47     1      1
8      2345  Lisa       Frimann      07-APR-49     0      1
9      9876  Kari       Fuglesang    17-MAY-75     1      0
10     9876  Finn       Fuglesang    13-AUG-72     1      0

SQL> select * from medlem;
medlemsnummer  adresse                info      historikk
-----
1234           Hyttheveien 4, 0666 Oslo
```

```

4321      Prf. dals gate 1, 0626 Oslo
2345      Langgata 90, 0500 Oslo
9876      Oslogata 1, 0123 Oslo
6543      Blindernveien 2, 0364 Oslo
8765      Sporveien 33, 0100 Oslo

```

```
SQL> select * from hage;
hagenavn
```

```

Solvang 79
Etterstad
Hjemmet
Rodeløkken

```

```
SQL> select * from parsell;
pnr      hage      medlemsnr      omsatt_dato      omsatt_pris
-----
1        Solvang 79    4321            01-JAN-01        100512
3        Hjemmet      8765            08-JUN-99        154500
2        Etterstad    9876            02-FEB-04        143500

```

```
SQL> SELECT * FROM søkerliste;
hage      medlemsnr      søkedato      sist_betalt
-----
Solvang 79    1234            01-JAN-04
Etterstad    1234            01-JAN-04
Rodeløkken   6543            01-DEC-00
Hjemmet      2345            12-DEC-03
Rodeløkken   9876            15-FEB-05

```

3.2.3 Oppslag som viser primitiv funksjonalitet

3.2.3.1 Eksempel 1

Oppslag som viser kontaktperson og ansiennitet for medlemmer som står på venteliste i Solvang 79

```
SQL> SELECT fornavn, etternavn, søkedato FROM Person, Søkerliste WHERE Person.medlemsnummer = Søkerliste.medlemsnummer AND er_kontaktperson = '1' AND Hage = 'Solvang 79' ORDER BY søkedato, etternavn, fornavn;
```

| fornavn | etternavn | søkedato |
|---------|-----------|-----------|
| Solveig | Slakte | 01-JAN-04 |

3.2.3.2 Eksempel 2

Oppslag som viser navn for kontaktperson for alle parseller i alle hager

```
SQL> SELECT fornavn, etternavn, parsellnummer, hage FROM Person, Parsell WHERE Person.medlemsnummer = Parsell.medlemsnummer AND er_kontaktperson='1' ORDER BY hage, parsellnummer;
```

| fornavn | etternavn | parsellnummer | hage |
|----------|-----------|---------------|------------|
| Kari | Fuglesang | 2 | Etterstad |
| Finn | Fuglesang | 2 | Etterstad |
| Pål | Askeladd | 3 | Hjemmet |
| Brummund | Dal | 1 | Solvang 79 |

Kapittel 4

Brukerdokumentasjon

4.1 Innsyn for vanlige medlemmer

Figur 4.1: Adressen til systemet



A screenshot of a web browser's address bar. The address bar contains the URL <http://heim.ifi.uio.no/sigvei/kolonistbasen/>. To the left of the address bar is a small globe icon, and to the right is a dropdown arrow.

Adressen til systemet for vanlige medlemmer er <http://heim.ifi.uio.no/sigvei/kolonistbasen/>.

Figur 4.2: Innlogging for vanlige medlemmer



A screenshot of a web page titled "Kolonihagedatabasen". The page has a decorative header with a row of colorful trees and a white picket fence. Below the header, the text reads: "Her kan du få innsyn i data som er registrert på ditt medlemsnummer. Tast inn ditt medlemsnummer og etternavn til en av de registrerte personene i husstanden for å fortsette." There are two input fields: "Medlemsnummer" and "Etternavn". Below the "Etternavn" field is a button labeled "Se data". At the bottom left, there is a link: [Innlogging for administratorer](#).

Gjennom det offentlige grensesnittet kan vanlige medlemmer logge seg inn for å se på de registrerte data om dem. For å gjøre dette må de taste inn en gyldig kombinasjon av medlemsnummer og etternavn (se figur 4.2).

Brukeren får, ved riktig innlogging, se sin egen medlemsinformasjon, hvilke personer som er registrert på medlemsnummeret, eventuelle tildelte parseller og ventelister medlemmet måtte befinne seg på.

4.2 Pålogging for administratorer

Figur 4.3: Innlogging i systemet



Tast inn brukernavn og passord for å fortsette.

Brukernavn:

Passord:

Logg inn

Adressen til pålogging for administratorer er <http://heim.ifi.uio.no/sigvei/kolonistbasen/admin.php>.

Hver gang du skal benytte systemet, må du logge inn med brukernavn og passord. Brukernavnet er foreløpig satt til 'admin' og passordet til 'sikker-Streng'. Innloggingsskjermen er vist i figur 4.3. Når du logger inn, vil du bli sendt videre til "Vis medlemmer".

For å logge ut kan du klikke på "logg ut" øverst i høyre hjørne.

4.3 Hovedmenyen

Hovedmenyen er vist i figur 4.4. Denne menyen vises så lenge du er logget inn, og du kan alltid klikke på en av knappene for å begynne på en ny oppgave.

Hovedmenyen er delt inn i tre kategorier: Medlemmer, parseller og oversikter. Denne brukerveiledningen har samme rekkefølge. Menyens valgmuligheter reflekterer de oppgavene som det er mulig å bruke systemet til, og de enkelte funksjonene er lagt opp slik at de følger normal arbeidsflyt.

4.3.1 Medlemmer

4.3.1.1 Nytt medlem

Figur 4.5 viser bildet du får opp når du velger alternativet "Nytt medlem".

For å legge til/opprette et nytt medlem må en her skrive inn medlemsnummeret medlemmet skal ha, samt kontaktadresse for medlemmet. Det er ikke mulig å generere medlemsnummer automatisk, ettersom Kolonihageforbundet ikke er interesserte i en slik funksjon.

Er det annen informasjon som kan være relevant om medlemmet, som for eksempel yrke, maksimumspris på parsell, hvordan parsellen skal benyttes

Figur 4.4: Hovedmenyen



(sommerbolig, helger, ferier el.), om medlemmet kan bidra til fellesskapet eller annet, skrives dette inn i “info”-feltet.

Feltet historikk benyttes for informasjon om medlemmet i søkerperioden. F.eks. om medlemmet har møtt til visninger, takket nei til en parsell eller lignende. Sjekk så at informasjonen stemmer og klikk så på “Lagre”.

Figur 4.6 viser det neste bildet som så kommer opp. Personer som tilhører medlemmet skal legges til i dette bildet (medlemsnummeret som ble skrevet inn i forrige bilde er synlig øverst i skjemaet).

Skriv inn fornavn, etternavn og fødselsdato til personen/den første av personene som ønskes lagt til medlemmet. Hak så av for om personen skal registreres som kontaktperson eller som barn av husstanden¹. Dette gjøres ved å trykke i firkanten etter ønsket resultat. Sjekk så at informasjonen stemmer, og trykk lagre. Ønsker du å gå tilbake til forrige vindu benytter du deg av “Videre”-knappen. Trykk på “Start på nytt” for å blanke ut feltene i skjemaet for ny person.

I det neste bildet får du mulighet til å legge til en ny person. Fremgangsmåten er den samme som for den første personen. For hver person du legger til kommer muligheten for å legge til flere personer. Om medlemmet ikke har flere personer benyttes “Videre”-knappen.

Figur 4.7 viser det bildet som så kommer opp. Her er en oversikt over data og personer som er lagt til. Nederst er det en mulighet for å legge medlemmet til venteliste.

For å legge medlemmet til ventelisten hakes det av for hagen(e) medlemmet

¹Det er verd å nevne at dette ikke er en aldersbestemt test; bare om personen er barn av de i husstanden som er medlem. Under “ventelister” kan du lete opp alle medlemmer som har barn under 16 år.

Figur 4.5: Skjermbildet for nytt medlem

Legg til medlem

Medlemmer

Nytt

Endre

Vis

Parseller

Ny

Endre

Vis

Oversikter

Statistikk

Ventelister

Her kan du legge til et nytt medlem.

Medl.nr:

Adresse:

Info:

Historikk:

Lagre

ønsker å stå på venteliste til (dette gjøres ved å klikke i firkanten til venstre for hagenavnet). Dato for når medlemmet søkte skrives inn i feltet(ene) "Søkedato" (dd.mm.åååå). Om det ikke skrives noe inn i dette feltet settes datoen til dagens dato. "Sist betalt"-feltet benyttes dersom det er registrert en innbetaling av ventelisteavgift. Datoen for siste registrerte betaling skrives da inn (dd.mm.åååå). Om det enda ikke er registrert noen betaling settes disse feltene tomme (Endre medlem benyttes for å legge inn denne opplysningen på et senere tidspunkt). Klikk så på "Lagre".

Da er registreringen av nytt medlem ferdig. Menyen til venstre kan benyttes for andre alternativer.

4.3.1.2 Endre medlem

Figur 4.8 viser bildet du får opp om du velger "Endre medlem". I dette bildet skal du velge et medlemsnummer. Dette gjøres ved å klikke på "velg". Da får du opp alle medlemsnummerene som finnes i databasen. Velg medlemsnummeret til medlemmet du ønsker å gjøre endringer for. Trykk så på "Søk".

Informasjonsoversikten for medlemmet som kommer opp er den samme som i figur 4.7. Ønskes endringer i "Adresse", "Info" eller "Historikk" (medlemsnummeret kan ikke endres), tar en bort eventuell utdatert data, og skriver inn det som er rett. Trykk så på "Lagre".

Ønskes endringer i personopplysningene trykker en på "Se detaljer". En får da opp et bilde som gir mulighet for å gjøre endringer i personer som er registrert på medlemmet, og eventuelt legge til nye personer. Trykk "Lagre" for hver person. Benytt "Tilbake" for å komme tilbake til oversiktsbildet.

Ønskes endringer i ventelisteinformasjonen, hakes eventuelle nye hager av, søkedato skrives inn/endres og dato for siste betaling skrives inn/endres. For å ta bort en hage, trykk i firkanten til venstre for hagenavnet, slik at feltet blir tomt. Når endringene som ønskes er utført trykk "Lagre".

Figur 4.6: Skjermbildet for registrering av personer

Endre personer

Her kan du endre eller legge til opplysninger personene i husstanden.

MEDLEMSNUMMER: 9854

Her kan du legge til en ny person

Fornavn:

Etternavn

Fødselsdato (dd.mm.åååå):

Kontaktperson:

Barn av husstanden: Lagre

Start på nytt | Videre

4.3.1.3 Vis medlemmer

Her kan du hente frem detaljert informasjon om ett medlem, eller oversikter over hvilke medlemmer som oppfyller visse søkekriterier, som for eksempel barnefamilier eller alle medlemmer i en bestemt hage.

Menyen befinner seg øverst i skjermbildet (fig. 4.9). Under denne vises resultatet av siste søk. Før første søk vil dette være en liste over alle registrerte medlemmer.

Den enkleste formen for søk er å velge et medlemsnummer fra nedtrekksmenyen “medlemsnummer”. Denne inneholder alle registrerte medlemmer. Ved å velge et medlemsnummer og trykke på “søk”, vises all informasjon som er registrert om dette medlemmet.

Et mer avansert søk kan gjøres ved å oppgi et søkefilter. Det er da kun medlemmer som tilfredsstillende samtlige søkekriterier som vil vises. Ulike kombinasjoner av disse kan med andre ord gi et meget avgrenset søk. Mulige kriterier er:

Hage: Hvis denne er satt er det kun medlemmer som enten har parsell eller står på venteliste i den valgte hagen som vil vises.

Kun barnefamilier: Hvis denne er satt er det kun medlemmer som har barn under 16 år som vil vises.

Kun medlemmer med parsell: Hvis denne er satt er det kun medlemmer som allerede er tildelt parsell som vil vises.

Kun medlemmer på venteliste: Hvis denne er satt er det kun medlemmer som står på venteliste som vil vises.

Fritekst: Hvis denne er satt er det kun medlemmer der den oppgitte strengen forekommer i enten fornavn eller etternavn som vil vises.

I oversikten som presenteres som resultat av et filtrert søk, gis det begrenset informasjon om hvert enkelt medlem. Ønskes mer informasjon om et av disse medlemmene kan bruker klikke på medlemsnummeret til medlemmet i listen. Det vil da vises en mer detaljert informasjon tilsvarende den som vises ved søk på medlemsnummer.

4.4 Parseller

4.4.1 Ny parsell

For å registrere en ny parsell, velger du “Ny Parsell” fra hovedmenyen. Skjerm-bildet i figur 4.10 kommer opp. Skriv inn et parsellnummer, velg en hage og velg hvilket medlem som eier parsellen i listen over medlemsnummer. I tillegg kan du skrive inn sist omsatt dato og pris ved siste salg, men dette er frivillig.

4.4.2 Endre parsell

For å endre opplysninger om en parsell, begynner du med å velge hage i det første skjermbildet. Deretter velger du hvilken parsell du ønsker å endre. Du får da opp skjermbildet som vises i figur 4.11. Her kan du endre opplysninger om sist omsatt dato og pris. For å lagre disse opplysningene velger du “Lagre endringer”. Merk at eventuelle endringer forsvinner dersom du velger “Tildel nytt medlemsnr.”

For å bytte parsell-leier, altså hvilket medlem som disponerer parsellen, velger du “Tildel nytt medlemsnr.”. Da kommer skjermbildet i figur 4.12 opp. Her kan du velge et nytt medlemsnr i nedtrekkslisten.

Hvis du ombestemmer deg, og altså ikke vil endre disponent av parsellen, velger du bare en ny handling i hovedmenyen.

4.4.3 Vis parseller

Denne siden gir mulighet for å liste opp alle parseller som er registrert i en hage. Bruker starter med å velge hage fra nedtrekksmenyen, hvorpå listen vises. Ved å klikke på parsellnummeret til en av parsellene i listen kan bruker få se en mer detaljert oversikt over den informasjon som er registrert vedrørende den valgte parsellen.

4.5 Oversikter

4.5.1 Statistikk

Denne siden gir mulighet for å hente frem statistiske data for en hage. Bruker starter med å velge hage fra nedtrekksmenyen. Den oversikten som da vises er en avpersonifisert liste over minimumsverdier, maksimumsverdier og gjennomsnitt for utvalgte data.

4.5.2 Vis ventelister

Dette skjermbildet gir mulighet for å liste opp alle medlemmer som er registrert på ventelisten for en hage. Bruker starter med å velge hage fra nedtrekksmenyen, hvorpå listen vises. Ved å klikke på medlemsnummeret til et av medlemmene i listen kan bruker få se en mer detaljert oversikt over den informasjon som er registrert vedrørende det valgte medlemmet.

Figur 4.7: Medlemsredigering

Her kan du redigere informasjon om det valgte medlemmet.

Medl.nr:

Adresse:

Info:

Historikk:

Ventelister

| Hage | Søkedato | Sist betalt |
|-------------------------------------|----------------------|----------------------|
| <input type="checkbox"/> Etterstad | <input type="text"/> | <input type="text"/> |
| <input type="checkbox"/> Hjemmet | <input type="text"/> | <input type="text"/> |
| <input type="checkbox"/> Rodeløkken | <input type="text"/> | <input type="text"/> |
| <input type="checkbox"/> Solvang 79 | <input type="text"/> | <input type="text"/> |

Personer

FORNAVN ETTERNAVN
per kristiansen

Parsell

Parsellnummer 54

Hage Rodeløkken

Figur 4.8: Endre medlem

Endre medlem

Her kan du endre opplysninger på et eller flere medlemmer til kolonihagene. Velg først et medlemsnummer før du går videre.

Velg et medlemsnummer:

Figur 4.9: Vis medlem-skjermbildet

Vis medlemmer

Her kan du hente frem en liste fra databasen med oversikt over medlemmer.

Velg medlem:
Medlemsnummer

Eller angi søkefilter:

Hage

Kun barnefamilier

Kun medlemmer med parsell

Kun medlemmer på venteliste

Fritekst

Viser medlemmer med barn under 16 år som står på venteliste i Etterstad

Medlemsnummer: [0111](#)

| FORNAVN | ETTERNAVN | FØDSELSDATO |
|------------|------------|-------------|
| Frederikke | Andreassen | 09.05.1998 |

Figur 4.10: Skjermbildet for ny parsell

Registrer ny parsell

Her kan du registrere en ny parsell

Parsellnummer: *

Hage: *

Medlemsnummer: *

Omsatt dato (dd.mm.åååå):

Omsatt pris:

Figur 4.11: Endring av parsell, før endring av medlemsnummer

Her kan du endre parsell. Velg først hage, deretter parsellnummer som skal endres.

| | |
|---------------------------|------------------------------------------------------------|
| Hagenavn: | Solvang 79 |
| Parsellnummer: | 1 |
| Medlemsnummer: | 1717 <input type="button" value="Tildel nytt medlemsnr."/> |
| Kontaktperson: | Bent Preben Lauritsen |
| Omsatt dato (dd.mm.åååå): | <input type="text" value="01.01.2005"/> |
| Omsatt pris: | <input type="text" value="100500"/> |

Figur 4.12: Skjerm bilde for endring av parsell-leier

| | |
|---------------------------|------------------------------------------------------------|
| Hagenavn: | Solvang 79 |
| Parsellnummer: | 1 |
| Medlemsnummer: | 1717 <input type="button" value="Tildel nytt medlemsnr."/> |
| Kontaktperson: | Bent Preben Lauritsen |
| Omsatt dato (dd.mm.åååå): | <input type="text" value="01.01.2005"/> |
| Omsatt pris: | <input type="text" value="100500"/> |

Velg nytt medlemsnummer:

Figur 4.13: Vis parseller-skjermbildet

Vis parseller

Her kan du hente frem en liste fra databasenen med oversikt over alle parseller i en hage.

Velg hage:

Parseller i Hjemmet

| | | |
|--------------------|-------|-----------|
| 3 | Pål | Askeladd |
| 10 | Sigve | Indregard |
| 22 | Eli | Rygg |

Figur 4.14: Statistikk-skjermbildet

Vis statistikk

Her kan du hente frem statistikk for en hage.

Velg hage:

Statistikk for Solvang 79

| | |
|------------------------------------------------|----------------------|
| Antall parseller i hagen | 32 |
| Antall beboere i hagen | 51 |
| Antall barn i hagen | 11 |
| Gjennomsnittlig eietid | 6 år |
| Gjennomsnittlig pris | 109652 kroner |
| Sist omsatt (pris) | 2005 (100500 kroner) |
| Høyeste pris | 200000 kroner |
| Laveste pris | 60 kroner |
| Antall medlemmer på ventelisten | 8 |
| Gjennomsnitlig ansenitet på ventelisten | 2003 |

Figur 4.15: Vis ventelister-skjermbildet

Ventelister

Her kan du hente frem en liste fra databasenen med oversikt over ventelisten for en hage.

Velg hage:

Venteliste for Etterstad

| MEDLEMSNUMMER | FORNAVN | ETTERNAVN | SØKEDATO | SIST_BETALT |
|----------------------|---------|----------------|------------|-------------|
| 8824 | Sofie | Abrahamsen-Lie | 30.03.2005 | |
| 0111 | Maria | Andreassen | 31.03.2005 | 02.02.2005 |

Del II

Utviklingsprosessen

Kapittel 5

Proessen

Prosjektets rammer har i all hovedsak vært fastlagt gjennom retningslinjene som gjelder for prosjektoppgaven i inf1050. Dette gjelder både tilgang til tid, penger og teknologisk plattform. Prosjektets delinnleveringer har utgjort naturlige milepæler for prosjektet. I tillegg har progresjonen i forelesningsrekken lagt sterke føringer for valg av utviklingsstrategi og utviklingsmetoder. Prosjektets fremste risiko har vært hvorvidt det ville bli for omfattende i forhold til den tiden vi har hatt til rådighet.

I tillegg utgjorde det faktum at systemet faktisk skulle tas i bruk i åden virkelige verdenå en viktig faktor for prosessen. Dette medførte i realiteten at vi hadde to oppdragsgivere og slik måtte forholde oss til to sett med krav; de formelle krav til kursets leveringer, samt krav fra systemets fremtidige brukere. Som en følge av dette kunne vi ikke selv ta avgjørelsene vedrørende hva systemet skulle omfatte, eller hvilken funksjonalitet det skulle tilby. I tillegg kom risikoen for at kurset krav og brukernes krav kunne komme i konflikt med hverandre. Hva hvis for eksempel brukerne ønsket at all registrert informasjon skulle være allment tilgjengelig, mens kurset krever at vi skal holde oss innenfor de rammer som eksisterende lovgivning setter?

I utgangspunktet trodde vi at vi skulle utvikle et rent ventelisteregister, muligens med et grensesnitt mot kolonihagens medlemsregister. I tillegg antok vi at systemet skulle være desentralisert, og slik kun omfatte en kolonihage. Heldigvis hadde vi tidlig i prosessen (2. februar) et møte i en utvidet prosjektgruppe, som også omfattet representanter fra hagen og fra forbundet. Vi fikk da avdekket at forbundet ønsket å avvikle dagens medlemsregister, og erstatte dette med vårt system. I tillegg var det ønskelig med et sentralisert system som omfattet et ukjent antall hager, men med lokal tilgang for tillitsvalgte i den enkelte kolonihage. I løpet av dette møtet fikk de fremtidige brukerne også anledning til å gi oss en inngående beskrivelse av systemets interesseområde, og de sentrale begreper som skulle inngå i systemet ble kartlagt. Dette ble til uvurderlig hjelp når UML modellene skulle utvikles, og disse gjengir svært nøyaktig det som ble avdekket under vårt innledende møte.

Det kan være verdt å nevne et usikkerhetsmoment relatert til dette møtet, nemlig vår kompetanse som systemutviklere i å kommunisere med en oppdragsgiver og fremtidige brukere. Dette var en problematikk som ikke var behandlet i kursets progresjon, og vi hadde derfor ingen andre retningslinjer å gå ut i fra en "common sense" og en mer allmenn kommunikasjonsforståelse. Dette viste seg i vårt tilfelle å være tilstrekkelig, men medførte en viss usikkerhet i situasjonen.

Etter møtet i den utvidede prosjektgruppen ble delinnlevering 1 oppdatert med de endringer som var kommet frem under møtet. Deretter ble da-

tabasens UML diagrammer generert. Dette var en relativt enkel operasjon etter den begrepsavklaring som ble gjort under møtet. Deretter ble databasen realisert med utgangspunkt i modellen. Dette var også en relativt enkel operasjon, da de fleste SQL statments var gitt av modellen. Rekkefølgen “begrepsavdekning”⇒“modellering”⇒“realisering” fungerte med andre ord utmerket for vårt formål. Den ene aktiviteten ga naturlige retningslinjer for den neste, og avklarte de spørsmål som måtte avklares for å kunne gå videre.

Selv om dette kan se ut som en fossefallsmetode, hadde prosessen også iterative innslag. Dette kommer blant annet til uttrykk gjennom de endringer som ble gjort i prosjektbeskrivelsen etter møtet den 2. februar. Videre medførte hver milepæl behov for å vende tilbake og evaluere eventuelle behov for revidering av tidligere delleveranser. I tillegg vil vår endelige leveranse til kurset kunne sees som første iterasjon i en totalleveranse ovenfor kolonihagene; det vil måtte foretas endringer og utvidelser før systemet virkelig kan tas i bruk av disse. Dette er for øvrig konsistent med en antagelse om at hver løkke i en iterativ metode gjerne har form av en fossefallsmetode.

Neste delinnlevering (nummer 4) innebar en form for helomvending. Vi skulle nå gå fra en “fra kjernen og ut” til en “fra skallet og inn”-tilnærming til den videre utviklingsprosessen. Dette hadde særlig en uheldig konsekvens for det endelige resultatet; vi begynte å realisere et skall som kommuniserte direkte med databaselaget, og ble derfor for sent oppmerksomme på gevinstene ved å implementere et applikasjonslag som kunne håndtere systemets forretningslogikk. Vår manglende erfaring med PHP og systemutvikling generelt bidrog nok til dette, men det er rimelig å anta at den endrede tilnærming til prosessen også var av betydning. Faktum er at den skrittvis logikken i prosessen forsvant, og slik usynliggjorde det neste logiske skritt (et applikasjonslag). En sterkere vektlegging av dette tema i kursets pensumlitteratur og forelesningsrekke kunne muligens ha avhjulpet dette problemet. Konsekvensen for vårt system ble at applikasjonslaget kun er delvis realisert.

For øvrig var genereringen av bruksmønstre, sekvensdiagrammer og klassemiagrammer for delinnlevering 4 også ev relativt enkel operasjon med utgangspunkt i den informasjon vi hadde innhentet fra den utvidede prosjektgruppen. Dette er en indikasjon på at begge metoder (kjernen-ut og skallet-inn) representerte adekvate tilnærminger til prosessen, og bekrefter dermed antagelsen om at vårt problem var en konsekvens av spranget fra den ene metoden til den andre.

Kapittel 6

Jus og etikk

Vårt system inneholder en rekke ikke-anonymisert personopplysninger, og er derfor underlagt personvernlovgivningen. Det viktigste spørsmål i denne sammenhengen var informasjonens grad av sensitivitet, samt vurderingen av hvem som skal ha tilgang til hvilken informasjon. Det ble tidlig avklart at vi ikke ville ha behov for å forholde oss til personnummer eller annen konsekvenspliktig informasjon, samt at informasjonen ville samles inn og ha et innhold som var i overensstemmelse med personvernlovens krav om frivillighet og relevans. Det var imidlertid ikke like klart hvordan vi skulle imøtekomme lovens krav til konfidensialitet og informasjonssikkerhet. Dette kom blant annet av at kolonihageforbundet ønsket å benytte systemet som en informasjonskanal ovenfor medlemmene. Fritt innsyn ville medføre et klart brudd på lovens bestemmelser. Vi måtte derfor realisere en ordning for gradert innsyn. I tillegg kom motforestillinger mot å forvalte unike brukernavn og passord for alle medlemmer.

Foreløpig er dette løst med to tilgangsnivåer. Nivå 1 gir fri tilgang til innsyn og oppdatering og er beskyttet av brukernavn og kryptert passord. Nivå 2 gir tilgang til innsyn i all informasjon om ett bestemt medlem. Dette nivået er beskyttet av samhörighet mellom det navn og medlemsnummer som oppgis ved oppslag. Vi vurderte medlemsnummeret som en tilstrekkelig ukjent faktor til å oppfylle et rimelig konfidensialitetskrav. I neste fase av prosjektet vil det også åpnes for innsyn i avpersonifisert statistikk fra den enkelte hage. Denne informasjonen vil være tilgjengelig for allmennheten.

Det som gjenstår for å oppfylle lovens krav er funksjonalitet for å slette data fra databasen. Foreløpig kan dette kun gjøres gjennom å operere direkte på databasen. Årsaken til at vi ikke har lagt inn denne funksjonaliteten i systemet er at det muligens kan medføre behov for et nytt tilgangsnivå for å sikre databasen mot utilsiktet sletting av data. Dette er imidlertid en politisk avgjørelse som må foretas av kolonihageforbundet. Vi har derfor valgt å avvende den videre dialog med dem før dette realiseres.

Dette bringer oss inn på andre politiske avgjørelser som påvirker, og påvirkes av, systemet. Ikke minst behovet for en avklaring av hvem som skal utføre de ulike operasjoner på systemet. I dag er det ansatt en person i 40% stilling for å håndtere innkommende søknader samt administrere den eksisterende (sentraliserte) databasen. I teorien kan denne personen overflødiggjøres ved innføringen av et desentralisert system, gjennom å delegere arbeidsoppgavene ut i den enkelte hage. Dette vil imidlertid medføre nye arbeidsoppgaver for de lokalt tillitsvalgte, samtidig som enkelte av deres tidligere arbeidsoppgaver forenkles.

Vi har forsøkt å legge minst mulig føringer for disse valgene i utforming-

en av systemet. Slik kan de politiske avgjørelser tas der de hører hjemme (i valgte styrer og organ) heller enn av oss. Et resultat av dette er at vi konsekvent har benyttet den generelle betegnelsen åsekretærå i våre bruksmønstre. Denne funksjonen kan utfylles både sentralt og lokalt, og kan også delegeres videre. Betegnelsen “salgskomite”, som finnes i vår innledende prosjektbeskrivelse, har derfor utgått.

En beslektet problemstilling med (fag)etiske implikasjoner finner vi igjen i utformingen av databasen. Et IT-system basert på en gjennomført stringent datamodell, der alle felt er strengt definert, legger sterke føringer for brukers anvendelse av systemet, og gir derfor lite rom for faglig og annet skjønn. Vi har valgt en modell basert på en erkjennelse av at vi som systemutviklere aldri vil kunne favne alle de variasjoner og endringer som kan finne sted i interesseområdet. Dette gjenspeiles i modellen gjennom en utstrakt bruk av åpne tekstfelt som i utgangspunktet kan benyttes til hva som helst. Feltene har fått navn som indikerer hva de er tenkt benyttet til, men faktisk bruk er til syvende og sist opp til bruker. Slik håper vi å ha åpnet for bruk av skjønn der dette er mulig og ønskelig.

Kapittel 7

Gruppedynamikk

På grunn av det relativt store omfang vårt prosjekt etter hvert fikk, ble vi nødt til å benytte en stor grad av arbeidsdeling mellom gruppedeltakerne, noe som ikke var helt i tråd med kursets krav til samarbeidsform. Dette løste vi ved å ha gjennomganger i plenum av alt som ble produsert, slik at vi kunne forsikre oss om at alle medlemmer av gruppen var i stand til å forstå og forsvare alle leveranser. Der vi antok at nødvendig øvelse i en aktivitet kunne gå tapt ved en stringent arbeidsdeling, utarbeidet alle deltakerne hvert sitt løsningsforslag. Deretter ble endelig løsning satt sammen i plenum, basert på de ulike forslagene. Den store variasjonen i php kodestil er en direkte konsekvens av denne arbeidsmåten.

Et annet resultat av arbeidsdelingen var at vi ble meget avhengig av den enkeltes innsats og individuelle bidrag. Hvis ett bidrag uteble kunne det være nok til å forsinke hele prosessen. Det skal ikke legges skjul på at vi opplevde problemer med dette, og var nødt til å iverksette disiplinerende tiltak for å komme i mål. Den erfaring vi trekker med oss til neste prosjekt er at med en sterkere avklaring av ansvarsforhold, roller og sanksjoner på et tidlig tidspunkt i prosessen kunne vi ha unngått noen av disse problemene. Feilen vi gjorde var vel å anta at prosjektets omfang var for lite til å rettferdiggjøre en slik innsats.

Et tredje problem som oppstod som en konsekvens av arbeidsformen var versjonshåndtering, eller rettere sagt mangelen på sådan. Også her antok vi initielt av vårt prosjekt var for lite til å rettferdiggjøre den overhead et versjonshåndteringssystem medfører. I ettertid ser vi imidlertid at selv et prosjekt på denne størrelsen kunne hatt stor nytte av et slikt system. Dette er også en lærdom vi tar med oss videre.

Kapittel 8

Endringer som følge av evalueringen

På grunnlag av evalueringsrapporten gjorde vi følgende endringer:

- Uoverensstemmelse mellom UML-diagrammene og implementasjonen. I UML-diagrammene er det et 0-1 forhold mellom medlem og parsell. Et medlem kan ha ingen eller en parsell. Systemet åpner imidlertid for at et medlem kan ha flere parseller. Uoverensstemmelsen skyldes en justering underveis. Under implementeringen kom vi frem til at det kan oppstå situasjoner der et medlem må kunne stå oppført med to parseller, f.eks. ved bytte av parsell. Mangelfull rettelse av UML-diagrammene var en forglemmelse, derfor har vi rettet opp dette.
- Dobbeltlagring(er_barn). Det hevdes at er_barn er en overflødig lagring av data, da denne informasjonen kan hentes opp ut fra fødselsdato. Dette er vi uenige i, da et barn av husstanden kan være en voksen person. Vi har forsøkt å gjøre dette klarere i rapporten.
- Salgskomiteen. Salgskomiteen har tilgang til samme funksjoner som sekretæren, derfor har vi valgt en aktør, sekretæren. Vi har forsøkt å gjøre dette klarere i rapporten.
- Funksjon for å slette medlemmer. Vi har valgt foreløpig ikke å implementere en slik funksjon, siden vi avventer tilbakemelding fra kolonihagene om tilgangsnivå til dette. Ved en utvidelse av systemet vil dette bli implementert. Vi har forsøkt å gjøre dette klarere i rapporten.
- Søk som eget bruksmønster. Vi mener at denne funksjonen allerede eksisterer under "vis". Vi har forsøkt å gjøre dette klarere i rapporten.
- Konstruktøren New() på entitetsobjektet "Medlemsregister". Metoden New() med de nødvendige variabler kan virke unødvendig. Derimot hviler dette diagrammet på at det skal opprettes en ny parsell og vi ønsker å vise dette på en slik måte at det ikke foreligger tvil om hva konstruktøren har til hensikt å gjøre: å opprette et nytt objekt (parsell) som skal inneholde relevant data. Vi har forsøkt å gjøre dette klarere i rapporten.
- Brukerdokumentasjonen som en beskrivelse ut fra oppgaver og ikke menyvalg. Vi mener at det går ut på det samme, da menyvalgene er oppgaveorienterte, og følger normal arbeidsflyt for sekretæren. Vi ser imidlertid fordelene ved å dele det opp ytterligere, da dette hadde vært et fint

oppslagsverk. Vi ønsker å la det være som det er, da vi tror dette vil være det beste hjelpemiddelet for brukeren. Vi har forsøkt å gjøre menyens oppgaveorientering klarere i rapporten.

- Det bør opplyses om hvor en finner parsell og medlemsnummere. Dette er administrative oppgaver som ligger utenfor systemet. Det er imidlertid en potensiell utvidelse av systemet, som kan bli aktuell på et senere tidspunkt. Vi har forsøkt å gjøre dette klarere i rapporten.
- Hvor mange heltall medlemsnummeret skal bestå av. Vi har valgt å sette dette relativt åpent da medlemsnummerformat håndteres utenfor systemet. Vi har forsøkt å gjøre dette klarere i rapporten.
- Evalueringsgruppen påpekte flere steder at enkelte ord og navn på etiketter kunne være misvisende for brukeren av systemet. Her har vi endret på teksten eller satt inn nye felter med informasjon for å gjøre forståelsen mer intuitiv. For eksempel vises nå navn på nåværende eier av parsell under "endre parsell".
- Evalueringsgruppen påpekte også at feilaktige fødselsdatoer ga "tekniske" feilmeldinger, og at dersom man tastet inn datoer å la 17.05.14 ble det registrert som 17.05.0014. Dette er nå rettet opp – gale datoer gir en snill feilmelding¹.
- Vanskeligheter med å sette medlemmer på venteliste under "nytt medlem". Funksjonaliteten for å gjøre dette ligger allerede der den skal, men systemet gjorde det ikke intuitivt nok å forstå hvordan man skulle gå frem om man ikke leste brukerdokumentasjonen godt nok. Vi har gjort endringer i systemet for å bedre dette.
- Ved registrering av ny parsell måtte en skrive inn alt på nytt om en har gjort feil. Dette er nå endret.
- Import/eksport av data til informasjonssystemet er en funksjonalitet vi ikke har implementert. Her har vi formulert oss uklart i rapporten ang hva vi mener med import/eksport. Det dreier seg nemlig om å lage et system som forenkler rutiner for å legge inn data (import), og for å hente ut data til sekretæren (eksport). Det dreier seg altså ikke om en teknisk import/eksport funksjonalitet av data. Vi har gjort endringer i rapporten for å oppklare denne misforståelsen.
- Vis venteliste - funksjon for allmennheten. Årsaken til at vi ikke ønsker å vise venteliste for allmennheten er fordi vi må ta hensyn til personvernloven. Hvis sensitiv informasjon skal være synlig på dette punktet er det en nødvendighet å ha tillatelse til dette fra kolonihagene og de enkelte medlemmene dette gjelder. Vi har forsøkt å gjøre dette klarere i rapporten.
- Evalueringsgruppen påpeker at det burde være en logg-ut funksjon på siden. Systemet inneholder allerede en utloggingsfunksjon øverst til høyre på alle sider, noe som også står forklart i brukerdokumentasjonen. Vi er usikre på om gruppen har oversett dette, eller om den ikke har vært synlig når gruppen testet den ut. Den ser i alle fall ut til å fungere godt nå, så her har vi ikke gjort endringer annet enn å gjøre funksjonen enda tydeligere visuelt.

¹Med unntak for semantisk gale datoer som 31. februar.

- Formuleringer i futurum. Vi har gått igjennom rapporten og gjort endringer der dette var aktuelt.
- Feilstaving av ordet ansiennitet. Vi har gått igjennom rapporten og gjort endringer der det var aktuelt.
- Manglende mulighet for å se statistikk fra ventelister for medlemmer. Vi har gjort statistikk-sidene tilgjengelig for ordinære medlemmer, i tråd med beskrivelsen i forprosjektet.
- Manglende beskrivelse av hva som er gjort for å sikre utvidbarhet i systemet. Vi har beskrevet dette bedre i rapporten.
- Evalueringsgruppen hevder at det er behov for bedre informasjon på siden om hvordan man skal gå frem for å utføre ulike bruksmønstre. Samtidig påpeker gruppen også at den første siden som admin-brukeren møter etter innlogging er noe overveldende. Vi har forsøkt å løse begge disse problemene ved å erstatte denne forsiden med en side som gir en beskrivende oversikt over hvilke valgmuligheter brukeren har, og hvordan brukeren kan gå frem for å løse ulike oppgaver.
- Endre medlem-siden hadde to forskjellige skjema: ett for registrering av medlemsdata, og ett for registrering av ventelistedata. Dette kunne være forvirrende for brukeren, og er nå erstattet av ett felles skjema.
- Vi har plassert en "hjem"-knapp i banneren øverst til venstre, for å gi brukeren en større opplevelse av kontroll i tråd med evalueringsgruppens anbefalinger.
- Det ytres også et ønske om å bytte plassering på søk og medlemslistingen på forsiden. Forsiden er nå endret. Vi mener søkemenyen er viktigere enn medlemsvisningen, og ønsker derfor å beholde plasseringen slik den er.
- Utseende i Internet Explorer. Alle sidene validerer ikke skikkelig med xhtml "strict". Dette skyldes primært en av biblioteksfunksjonene ("lag-SelectMeny") som ikke oppfyller xhtml kravene. I og med at vi ikke ønsker å endre på koden i denne funksjonen (som ikke er skrevet av oss), vi har istedet valgt å endre DTD til "transitional" inntil videre.
- Innloggingsfeltet kommer nå opp på nytt hvis brukeren skriver feil brukernavn/passord, i motsetning til tidligere (da måtte du bruke tilbakeknappen).

Følgende punkter har ikke ført til endringer:

- Det etterlyses en begrunnelse for hvorfor kolonihagene ikke ønsker at systemet skal generere medlemsnummere. Vi har ikke mottatt noen begrunnelse for dette, men godtar det som et valg fra oppdragsgivers side. Vi mener det ikke er behov for begrunnelse utover dette.
- Manglende funksjonalitet for å koble seg til eksterne databaser. Dette er ment som en mulig utvidelse. Vi har forsøkt å gjøre dette klarere i rapporten.
- Legge medlemmer til ventelister - funksjon direkte fra "vis medlemmer". Vi mener dette er en overflødig funksjon, da dette kan gjøre under nytt medlem og endre medlem. Normalt settes medlemmet på venteliste ved første registrering, da dette er informasjon fra søknadsskjema.

- Det ytres et ønske om å bytte plassering på søk og medlemslistingen på "vis medlemmer". Vi mener søkemenyen er viktigere enn medlemsvisningen, og ønsker derfor å beholde plasseringen slik den er.
- Hvorfor er rullgardinen med medlemsnummer ikke begrenset til de som står på venteliste? Alle medlemmer må være tilgjengelig på denne listen, da det kan oppstå tilfeller der noen som allerede har parsell (og derfor ikke står på venteliste) skal bytte parsell.
- Evalueringsgruppen mener det er problematisk at en parsell MÅ være koblet til et medlem, da dette kan by på problemer når nye parseller som ikke har blitt tildelt medlem kommer til. Grunnen til at vi har valgt å gjøre det slik er at etter vår mening har en parsell ingen grunn til å legges inn i databasen ved mindre den er tildelt et medlem. Databasen er medlemsorientert, og ikke parsellorientert. Vi har forsøkt å gjøre dette klarere i rapporten.
- Ellers lurte evalueringsgruppa på hvor resten av informasjonen fra søknadsskjema blir av. Resten av informasjonen er tenkt under "info" feltet. Vi ønsket at det skulle være relativt åpent, slik at all relevant informasjon kan lagres her. En utvidelse av dette i samarbeid med kolonihagene er en mulighet, men det er "nice to have" funksjonalitet utenfor rammene av dette prosjektet. Vi har forsøkt å gjøre dette klarere i rapporten.
- Det påpekes at "info" feltet under medlemsdataene gjør det umulig å vite hvem av personene tilhørende medlemmet informasjonen gjelder. Dette ble diskutert i gruppa hvordan dette skulle løses i starten av prosjektet. Begrunnelsen for hvordan vi valgte å løse det ligger i at denne informasjonen er av interesse ved behandling av søknad. Det er irrelevant for kolonihagen hvem av personene som er rørlegger (og kan bidra med slike arbeider i hagen) eller hvem som kan sitte med styreverv, bare en av personene er /kan dette. Vi har forsøkt å gjøre dette klarere i rapporten.
- Menyen ser lite tiltalende ut i Internet Explorer. Dette skyldes IEs manglende støtte for CSS2.1.

Del III

Evaluering av PIMEDAB

Innledning

Under arbeidet med evalueringen av informasjonssystemet PIMEDAB har vi tatt utgangspunkt i den foreslåtte strukturen i EVALUERING.HTML fra kursets hjemmesider. Det vil si at vi ikke har fulgt systematikken som er foreslått i kurslitteraturen til punkt og prikke. Med utgangspunkt i spørsmålene fra EVALUERING.HTML gjennomførte en felles diskusjon i gruppen rundt disse og fordelte deretter skrivearbeidet mellom oss. Vi møttes deretter for å presentere resultatene for evalueringen og trekke konklusjoner i plenum.

Rapporten er strukturert i to deler. Kapittel 9 omfatter en evaluering av prosjektrapporten som ligger til grunn for systemet, og Kapittel 10 tar for seg selve systemet. I kapittel 10 utførte vi en rekke oppgaver for å teste utførelse, funksjonalitet og i hvor høy grad vi hadde kontroll når vi utførte oppgavene.

Kapittel 9

Evaluering av prosjektrapporten

9.1 Konsistens mellom gruppert og ugruppert UML -diagram.

9.1.1 Den ugrupperte modellen

En ugruppert modell skal være en visualisering av koplingen mellom de begreper som interesseområdet omfatter. Den ugrupperte modellen til *PIMEDAB* dekker de aktuelle begrepene på en god måte, og viser en naturlig og logisk sammenheng mellom dem. Assosiasjonene mellom klassene inneholder gode rollebeskrivelser og modellen viser også identifiserende assosiasjoner. I tilfeller med mange-til-mange multiplisiteter mellom klasser har gruppen inkludert assosiasjonsklasser.

I beskrivelsen av systemets funksjonalitet viser gruppen til at *PIMEDAB* skal kunne håndtere ulike grupper av brukere med ulike rettigheter, men at dette blir tilleggsfunksjonalitet som må utvikles utenfor rammen av denne prosjektoppgaven. Vi synes det er positivt at gruppen likevel har valgt å inkludere en slik utvidelse i datamodellen for å sikre at ikke datamodellen skaper sperrer når dette skal implementeres senere, men den grupperte modellen lykkes ikke i å skille dette like godt ut som den ugrupperte.

9.1.2 Konsistens med gruppert modell

I den grupperte datamodellen har begrepene blitt “trukket sammen” på en god måte. Det er i stor grad konsistens mellom begreper, assosiasjoner og klasser i den ugrupperte og grupperte modellen. Det er imidlertid et par unntak.

For det første bør det kommenteres at assosiasjonsklassene som ble satt inn i den ugrupperte modellen fremdeles består som assosiasjonsklasser i den grupperte modellen. Disse assosiasjonsklassene burde vært erstattet med vanlige klasser, og blitt plassert inn i modellen på en bedre måte. Videre har det skjedd et underlig bytte mellom plasseringen av klassene *Kategorier* og *Hovedkategori* fra den ugrupperte til den grupperte modellen. Dette er beskrevet nærmere i neste avsnitt i sammenheng med konsistens med SQL-strukturen.

Tilslutt må det også nevnes noe om én av assosiasjonene i datamodellen: Slik vi forstår datamodellen er hovedkategoriene en delmengde av kategoriene. Selv om dette for oss virker som en noe unaturlig struktur forstår vi hvordan gruppen har tenkt. Det ville imidlertid være mer naturlig å forestille seg

at flere kategorier kan inngå i en hovedkategori. Uansett finner vi at multiplisitetene mellom disse klassene ikke representerer noen av variantene. Modellen forteller oss at en kategori hører til en til mange hovedkategorier, men i følge gruppens logikk burde en kategori kun tilhøre null eller en hovedkategori.

9.2 Konsistens mellom UML-diagram og SQL-statements

Det er i hovedsak god konsistens mellom UML-diagrammene som beskriver databasestrukturen og de SQL-statements som er brukt for å opprette databasen. Noen unntak er imidlertid verdt å bemerke. La oss ta tabell for tabell.

9.2.1 Piktogrammer

Dette er databasens sentrale tabell.Attributtene *piktogramID*, *bildenavn* og *tegnspråk tegn* er implementert i henhold til modellen. Det eneste som måtte være å utsette på dette er en inkonsistens i bruken av stor og liten forbokstav i navnet. Dette er for øvrig noe som går igjen i de fleste av tabellene, men må vel allikevel kunne anses som en relativt uvesentlig detalj. Mer problematisk er det at tabellen i modellen har tre attributter (*Last*, *personAvbildet* og *PersonID*) som ikke er implementert i det hele tatt. Etter nærmere granskning forstod vi at disse var tiltenkt en senere utvidelse av systemet, men dette kunne godt ha kommet tydeligere frem av modellen.

9.2.2 Bildefiler

Denne tabellens primærnøkkel (*FilID*) er forsvunnet i implementeringen. I stedet er en kombinasjon av to andre attributter (*filnavn* og *størrelse*) benyttet som primærnøkkel. Det har sikkert vært gode grunner for å gjøre det slik, men endringen burde vært gjenspeilet i modellen.

9.2.3 Kategorier

Denne tabellen har skiftet navn fra *Kategori* til *Kategorier*, og gjenspeiler slik en mindre inkonsistens i navngivingen av tabellene i modellen. For øvrig er tabellen konsistent med modellen.

9.2.4 Hovedkategorier

Denne tabellen fant vi det noe problematisk å forstå formålet med. Den består utelukkende av ett attributt, som til overmål er en fremmednøkkel. Selv om implementasjonen er i henhold til modellen finner vi det verdt å bemerke at denne burde vært overflødig. Tabellen er et sub-sett av *Kategorier*-tabellen, og kunne slik sett vært implementert for eksempel ved hjelp av et flagg i sistnevnte. Se for øvrig kommentarene til neste tabell nedenfor.

9.2.5 Kobling

Dette er en assosiasjonstabell, og burde slik sett vært tegnet inn mellom de to tabellene med mange-til-mange-relasjoner, og ikke kun markert med en stiplet linje slik det er gjort i modellen. For øvrig viser implementasjonen at den er

feilplassert i modellen. I realiteten identifiserer den relasjoner mellom *Piktogrammer* og *Kategorier*, og ikke mellom *Piktogrammer* og *Hovedkategorier*, slik det fremgår av modellen. Dette problemet antar vi er en konsekvens av *Hovedkategorier*-tabellens noe uklare rolle, og vi tror problemet kunne vært løst ved å fjerne denne.

9.2.6 Om assosiasjonsmultiplisiteter

Ingen av assosiasjonene er angitt med multiplisitet 0 i modellen. Imidlertid tillater implementasjonen at det godt kan eksistere piktogrammer uten bildefiler, kategorier uten hovedkategorier og hovedkategorier uten piktogrammer. Dette burde kanskje vært angitt i modellens assosiasjonsmultiplisiteter.

9.3 Sammenheng mellom bruksmønstre og de overordnede mål

PIMEDAB har bruksmønsterdiagrammer for at en aktør (ansatt) skal kunne utføre følgende handlinger:

1. skrive ut piktogrammer
2. legge inn piktogrammer i databasen
3. oppdatere opplysninger om piktogrammer
4. slette piktogrammer
5. søke etter piktogrammer

I vår gjennomgang av bruksmønsterspesifikasjonene fant vi ikke en tekstlig beskrivelse for handling nummer to (legge inn piktogrammer i databasen). I stedet hadde gruppen valgt å kalle bruksmønsteret for "Last opp bildefil". Vi oppdaget også at flere deler er slått sammen i bruksmønsterspesifikasjonen. Dette gjelder handling nummer fire (slette piktogrammer) og fem (søke etter piktogrammer). Disse to er slått sammen til "Slette bildefil".

Vi savner en tydeligere og mer spesifikk skriftlig dokumentasjon på hvilket bruksmønster som utfører hvilken handling. Vi føler også at det skulle vært utarbeidet et bruksmønster for å vise piktogrammer. Foreløpig kan det nærmeste vi kommer for å vite hvordan vi skal kunne se et piktogram beskrevet i "Slette bildefil"-bruksmønsteret.

På side 5 i prosjektbeskrivelsen kan vi i avsnittet "Funksjonalitet" se at gruppen har satt opp en rekke kriterier for hva det påtenkte system skal kunne gjøre. Noe vi opplevde som uklart var det siste punktet i tabellen. Her har gruppen ført opp at dette skal være (i tillegg til de andre øvrige mål) "*et system for redigering av søkekriterier*". Vi hadde en diskusjon om hva slags type funksjonalitet dette kunne være, men vi kom ikke med en endelig konklusjon. Vi mener at dette punktet kan være mer konsekvent i forhold til de andre punktene i beskrivelsen.

9.4 Sekvensdiagram og objektorientert klasse-diagram

Vi forutsetter at det som vises er sekvensdiagrammet for bruksmønsteret "Skriv ut", selv om dette ikke er slått fast noe sted. Det er beklagelig at det

ikke finnes noen redegjørelse for forskjellene mellom sekvensdiagrammet og det objektorienterte klassediagrammet, og vi må derfor gjøre noen antagelser om dette.

I sekvensdiagrammet finnes det tre objekter (kantobjektet, kontrollobjektet *Søk* og forretningsobjektet *PIMEDAB*). Det siste er en sammentrekning av hele datastrukturen, og det forenkler mye av sekvensdiagrammet til et høyt abstraksjonsnivå ved at *PIMEDAB* er meget intelligent og forstår kommandoer av en såpass høy orden som *søkIRegister(søkekriterier)*. Dette abstraksjonsnivået er gjennomført og konsistent i diagrammet, men overlater relativt mange implementasjonsdetaljer til den "svarte boksen" *PIMEDAB*.

I det objektorienterte klassediagrammet finner vi ikke igjen kantobjektet. Metodene som kontrollobjektet kaller på kantobjektet er flyttet over til kontrollobjektet. Dermed har kontrollobjektet *Søk* fire metoder: *søk*, *velgBilde*, *bildeViser* og *printBilde*. I sekvensdiagrammet kalles to av disse, *bildeViser* og *printBilde*, på kantobjektet - ikke kontrollobjektet¹.

Selv om det er et spørsmål om smak og behag hvorvidt kant- og kontrollobjektene bør inkluderes i det objektorienterte klassediagrammet, er inkonsekvensene som vises i det objektorienterte diagrammet et sterkt argument for at de bør inkluderes. Vi har vanskelig å se for oss at kontrollobjektet "Søk" bør ha medlemsfunksjonene *printBilde* og *bildeViser*, noe også sekvensdiagrammet gjenspeiler.

Vi tror at klassediagrammet ville ha blitt mer informativt ved å inkludere kantobjektet.

Før øvrig er diagrammene konsistente med hverandre.

9.5 Brukerveiledningen

Brukerveiledningen er kortfattet, og relativt oversiktelig med forklaring av hva en ser i åpningsbildet, hvilke funksjoner som finnes og hvordan disse fungerer. Den går igjennom systemet fra venstre til høyre med underliggende funksjoner. Kortfattetheten gjør at den er enkel å lese igjennom, og gir raskt en god oversikt over systemet.

Det at brukerveiledningen er systemorientert og ikke bruksorientert kan være et minus da brukeren kan være en relativt fersk pc-bruker, som det fremkommer tidligere i oppgaven:

"Vi kan ikke forutsette at primærmålgruppas IT-kompetanse er over middels. Systemet må altså være så intuitivt og brukervennlig som mulig."

(Under "målgruppe" på s. 5)

Med systemorientert mener vi at utgangspunktet ligger i systemets oppbygning, og ikke på oppgaveutførelse og bruk. Vi tror med tanke på "ferske" brukere at det ville være hensiktsmessig med en veiledning som hjelper brukeren gjennom de ulike funksjonene programmet innehar. For eksempel funksjonene: legge inn nytt piktogram, redigere/slette et piktogram og søke etter piktogram. Brukerveiledningen kunne også med fordel innholdt en forklaring av variasjoner. Hvordan går en frem hvis noe ikke går som det skal? Vi tror dette ville kunne lette oppgavene for en nybegynner som forsøker å sette seg inn i systemet.

¹UML-notasjonen i sekvensdiagrammet er for oss noe ukjent i forhold til disse kallene, og det kan hende at det er meningen at "pil til boks utenpå et annet objekt" er ment å representere kall på seg selv. I så fall blir inkonsekvensen at "velgBilde" burde vært et medlem av kantobjektet, ikke kontrollobjektet.

Under forklaringen til funksjonene på høyre side (s. 21) vil det trolig være fordelaktig at knappen hadde et navn, for å unngå forvirring.

“...klikk på knappen som heter ‘bla gjennom’, ‘browse’ eller lignende.”

Kapittel 10

Evaluering av systemet

Under evalueringsprosessen av systemet valgte vi å fordele arbeidsmengden på en slik måte at vi tildelte tre ulike spørsmål som vi la til grunn for evalueringen av systemet. Vi lagde følgende tre oppgaver:

1. Hente frem tegnspråk tegn som omhandler mat,
2. legge inn et nytt piktogram og
3. endre et piktogram.

Vi tok for oss hvert av spørsmålene og så nærmere på om det var vanskelig å utføre oppgaven, om systemet gjør det som står i bruksmønsteret (inkludert variasjonene) og om vi følte at vi hadde kontroll under prosessen. La oss ta det kronologisk.

10.1 Hente frem språk tegn som omhandler mat

Vi har generelt sett et inntrykk av at systemets søkemotor er uferdig. Oppbygging og funksjonalitet synes å være godt gjennomtenkt, men realiseringen ser ikke ut til å fungere etter hensikten.

10.1.1 Utførelse

Søkefunksjonen er tilgjengelig fra systemets hovedside, og det er enkelt å fylle inn de ulike søkekriteriene. Skulle det være noe å utsette på dette måtte det være at vi må til brukerdokumentasjonen for å finne ut hvordan kombinasjoner av de ulike valgmulighetene påvirker hverandre (er det et additivt eller subtraktivt søkefilter etc.).

10.1.2 Funksjonalitet

Her er det problemene oppstår. Når vi velger tegnspråk piktogrammer og kategorien “mat” får vi, så vidt vi kan se, en liste over alle piktogrammene i databasen. Velger vi derimot å legge inn “mat” som søkeord får vi opp alle piktogrammer, også ikke-tegnspråk, som vedrører mat. Avhukningsboksene på søkesiden ser med andre ord ikke ut til å fungere.

10.1.3 Kontroll

Hadde søket gitt det resultatet vi forventet ville vi nok opplevd en stor grad av kontroll, i hvert fall hvis vi på forhånd hadde satt oss inn i de kombinatoriske søkemulighetene ved hjelp av brukerdokumentasjonen. Opplevelsen av funksjonelle feil medfører imidlertid at følelsen av kontroll går mot null; vi skjønner rett og slett ikke hva som skjer.

10.1.4 Konklusjon

Når de funksjonelle feil rettes opp tror vi denne oppgaven vil kunne utføres på en enkel og sikker måte. Hvis brukergrensesnittet i tillegg gir en klarere indikasjon av hvordan ulike søkekriterer kombineres tror vi brukeren vil oppleve en stor grad av kontroll.

10.2 Legge inn et nytt piktogram

10.2.1 Utførelse

Det er i hovedsak enkelt og intuitivt å laste opp bilder i systemet. Funksjonaliteten er lett tilgjengelig fra forsiden av systemet, og det er svært positivt at systemet gir god informasjon og veiledning om hvilke begrensninger systemet har og hvordan man skal gå frem ved opplasting.

10.2.2 Funksjonalitet

“Happy day scenario” virker slik det er beskrevet i bruksmønsteret. Det er enkelt og greit å velge bildet som skal lastes opp, knytte kategorier og nøkkelord til det, og få det lagret. Vi møtte imidlertid et problem: Selv om systemet forteller meg at jeg kan “knytte nye kategorier til piktogrammet”, finnes det ikke mulighet for å bruke andre kategorier enn dem som allerede er registrert i systemet. Hvordan skal brukeren opprette nye kategorier? I tillegg var det vanskelig å vite hvordan nøkkelord skal struktureres – skal det brukes komma, mellomrom eller linjeskift mellom dem? Vi savner også mulighet til å legge inn navn på piktogrammet, slik det beskrives i bruksmønsteret.

Variasjonene i bruksmønsteret fungerte ikke helt i tråd med slik de var beskrevet:

- Dersom man lastet opp et filformat som systemet ikke støttet ble det vist en feilmelding som i hovedsak bestod av SQL-feilsetninger. Det fremgikk at systemet ikke støttet filformatet, men det var ingen enkel måte å komme seg tilbake for å laste opp et annet bilde.
- Variasjonen om opplasting av for lite bilde slo ikke til i det hele tatt. Det var fullt mulig å laste opp bilder som ifølge systemets brukerveiledning var for små (under 200x200 piksler).
- Dersom man ikke oppga filnavn til bildet, ga systemet ingen forståelige feilmeldinger for personer som ikke forstår SQL.

10.2.3 Kontroll

Under “happy day scenarioet” ga systemet brukeren inntrykk av høy kontroll. God informasjon og tilgjengelighet på funksjonaliteten bidro til dette.

Variasjonene og feilmeldingene ga derimot grunn til bekymring. SQL er gresk for de fleste, og for mange kan det se ut som om man har gjort noe alvorlig galt når man ser meldinger som begynner med “WARNING!”. At man må bruke “back” knappen for å gå tilbake gir i tillegg inntrykk av at systemet ikke var forberedt på den “alvorlige” feilen man nettopp gjorde.

10.3 Endre et piktogram

10.3.1 Utførelse

På informasjonssystemets hovedside er det en søkefunksjon for redigering og sletting av bilder. Her er søkefunksjonen lik de andre og man taster inn et nøkkelord for å finne frem riktig piktogram. Dette er uproblematisk, jeg tastet inn “natur” og fikk opp alle mulige treff på denne forekomsten.

På hovedsiden for systemet er det en “Les mer [...]” link man kan lese før man foretar et søk for å endre eller slette et piktogram (dette er anbefalt, man blir ikke tvunget av systemet til å lese veiledningen, men man får nyttige tips).

For å endre et piktogram trykker man på valgt bilde etter at søket er utført for å komme til endre/slette siden. Selve endringsprosessen er relativt enkel. Man blir presentert for en rekke valg man kan utføre før man ønsker å lagre bildet. Dette er for eksempel å knytte nye nøkkelord til piktogrammet, tildele nye/flere kategorier til piktogrammet, skifte navn på piktogrammet og slette piktogrammet - med tilhørende opplysninger - fra databasen.

10.3.2 Funksjonalitet

Det virker som denne funksjonen ikke er fullt utviklet enda (endre piktogram). Vi forsøkte å endre informasjon tilknyttet piktogrammet, men det så ut til at informasjonen som ble tastet inn ikke ble lagret. Blant annet forsøkte vi å endre nøkkelordene tilknyttet bildet, navnet osv. uten hell. Man fikk ikke utført de endringene som var påtenkt piktogrammet. Vi forsøkte også flere variasjoner på endringsinformasjonen uten å lykkes (dette var for eksempel å kun skrive inn et nytt nøkkelord til piktogrammet, endre navn osv.).

10.3.3 Kontroll

Etter at man har lest veiledningen om endring/sletting av bilder kunne endringen virke uproblematisk. Søkemotoren virker godt for dette formål og viste frem et resultat som stod i stil til nøkkelordet vi tastet inn. Men det var etter dette vi opplevde problemer. Vi kunne dessverre ikke foreta endringer som ble lagret i informasjonssystemets database. Hvis denne funksjonen blir fullstendig utviklet tror vi at aktør ikke vil ha problemer med å få utført eventuelle endringer på et piktogram. Endringssiden er til dels intuitiv og selvforklarende og skal ikke by på de største utfordringer.

10.4 Konsistens i brukergrensesnitt

Brukergrensesnittet er bygd opp “flatt”, dvs. ved hjelp av en velkomstskjerm som gir tilgang til samtlige bruksmønstre umiddelbart. Grensesnittet er delt i tre kolonner, der den venstre gir generell informasjon om systemet, den midtre gir søkemuligheter og den høyre gir redigerings- og slettingsmuligheter.

På de ulike skjermbildene som følger er overskrifter, bruk av farger og navngivning av knapper gjennomført og konsistent.

Systemet gir en lenke tilbake til forsiden, men denne lenken har overskriften til gjeldende side som tekst. Overskriftslinjen med lenke til forsiden følger brukeren rundt i systemet, og denne lenken er til stor nytte som et sentralt orienteringspunkt i systemet. Nye brukere vil imidlertid – i likhet med oss – neppe forstå at å klikke på overskriften til gjeldende side fører til forsiden.

Systemet baserer seg på en meget bred side med mye informasjon på en gang, noe som har både fordeler og ulemper. En av ulempene er at det kan være vanskelig å få oversikt over alle mulighetene du har på en side.

Når dette systemet tas i bruk vil vi anta at den desidert vanligste brukeren vil være enkel fritekstsøking i registeret, og vi tror det ville føre til et mer konsistent og intuitivt grensesnitt dersom søkeboksen fikk sin faste, fremtredende plass gjennom hele grensesnittet¹. Det nåværende grensesnittet har en tendens til å flytte søkeboksen, f.eks. fra sentrum til høyre, noe som oppleves som litt forvirrende.

Bortsett fra dette er systemet konsistent og helhetlig, og det merkes at den brede siden er et bevisst valg som utnyttes i stor grad. Farger, overskrifter og knapper gir et inntrykk av helhet, struktur, trygghet og seriøsitet.

10.5 Konsistens mellom brukerveiledning og systemet

Konsistensen mellom brukerveiledningen og systemet er god. Imidlertid kan måten brukerveiledningen er skrevet på (systemorientert) gjøre at det kan oppstå variasjoner som ikke er redegjort for i brukerveiledningen. Når noe ikke går som det skal, vil en “fersk” bruker (som det forutsettes at dette systemet kan ha) lett kunne bli forvirret. Da dette systemet har få “tilbake”-knapper og ingen “tilbake til hovedmeny”- eller “start på nytt”-knapper ville en veiledning i slike situasjoner være svært nyttig. Mye av brukerveiledningens innhold ligger som informasjon underveis i systemet, noe som kunne være en god støtte når en står fast, men når noe ikke går som det skal mangler slike informasjonsknapper.

Ikke alle funksjoner fungerer som de står beskrevet i brukerveiledningen, dette redegjøres det for i andre deler av denne evalueringen.

¹F.eks. á la google - en eneste stor søkeboks med lenker til mer avanserte funksjoner. Et slikt grensesnitt er grunnleggende intuitivt for 99% av brukerne og bruksmønstrene. Kanskje vil det etterhvert være noenlunde likt for PIMEDAB.

Figurer

| | |
|-----------------------------------------------------------------|----|
| 2.1 Bruksmønsterdiagram (use-case) | 8 |
| 2.2 Sekvensdiagram for registrer ny parsell | 13 |
| 2.3 Klassediagram for registrer ny parsell | 13 |
| 3.1 Dataorientert klassediagram, ugruppert form | 15 |
| 3.2 Dataorientert klassediagram, gruppert form | 16 |
| 4.1 Adressen til systemet | 19 |
| 4.2 Innlogging for vanlige medlemmer | 19 |
| 4.3 Innlogging i systemet | 20 |
| 4.4 Hovedmenyen | 21 |
| 4.5 Skjermbildet for nytt medlem | 22 |
| 4.6 Skjermbildet for registrering av personer | 23 |
| 4.7 Medlemsredigering | 25 |
| 4.8 Endre medlem | 25 |
| 4.9 Vis medlem-skjermbildet | 26 |
| 4.10 Skjermbildet for ny parsell | 26 |
| 4.11 Endring av parsell, før endring av medlemsnummer | 27 |
| 4.12 Skjermbilde for endring av parsell-leier | 27 |
| 4.13 Vis parseller-skjermbildet | 28 |
| 4.14 Statistikk-skjermbildet | 28 |
| 4.15 Vis ventelister-skjermbildet | 29 |

Litteraturliste

- Skagestein, Gerhard 2002: "*Systemutvikling - fra kjernen og ut, fra skallet og inn*". Høyskoleforlaget, Oslo.
- Hojtsy, Gabor (red.) 2005: "*PHP Manual*". The PHP Documentation Group, URL: <http://www.php.net/manual/en>
- Niederst, Jennifer: "*Web design in a nutshell*". 2. utgave, 2001. O'Reilly.