

# Lokasjons -og Kontekstbaserte Tjenester fra et Informasjonsforvaltningsperspektiv

LOST: Et Lokasjonsbasert Informasjonssystem for studenter ved  
NTNU Gløshaugen

**Bjørn Egil Ødegård Haugvik**

Master i informatikk

Oppgaven levert: Juni 2006

Hovedveileder: Ingeborg Torvik Sølvberg, IDI



## Forord

*”Å vite når man vet noe og å vite når man ikke vet noe - det er kunnskap” sa Konfucius. Så hvordan stiller det seg da når jeg nå har viten om at jeg ikke vet noen ting?*

*Men fra spøk til alvor, arbeidet med denne avhandlingen har vært utrolig lærerik på mer enn én måte. Det har innbefattet personlig læring og utvikling såvel som gitt meg større faglig innsikt og kompetanse.*

*Avhandlingen er avslutningen på et mastergradsstudium i informasjonsforvaltning ved Institutt for Datateknikk og Informasjonsvitenskap ved Norges Teknisk-Naturvitenskaplige Universitet (NTNU).*

*Min medstudent Geir Marius Gjøl og jeg troppet opp på vordende veileder Ingeborg Sølvbergs kontor med kun en vag idé om hva vi ønsket å skrive om. Lokasjonsbaserte tjenester syntes å være et nytt og interessant emne for oss teknologiinteresserte studenter. Vi hadde store planer om å levere en felles oppgave, men den gang ei. Ingeborg krevde intet mindre enn at vi skulle levere hver vår individuelle oppgave med fokus på forskjellige aspekter ved temaet, og slik ble det.*

*Arbeidet med oppgaven har vært en lang og utfordrende prosess, spesielt med tanke på at jeg på forhånd ikke hadde hatt noen som helst befatning med emnet jeg valgte. Heldigvis var det mye dyktig veiledning å få fra Ingeborg, som - til tross for at hun befant seg i California det siste semesteret - stødig veiledet meg igjennom arbeidet og passet på at jeg hele tiden holdt riktig fokus. Jeg vil gjerne rette en stor og dypfølt takk til Ingeborg for den gode veiledningen.*

*Videre vil jeg takke min medstudent, Geir, som har vært en interessert og lyttende, men også kritisk, nysgjerrig og nyttig diskusjons- og samarbeidspartner under oppgaveskrivingen.*

*Til slutt vil jeg også takke mine foreldre som har støttet og hjulpet meg gjennom den lange tiden som student.*

NTNU, 8. Juni, 2006

Bjørn Egil Haugvik

## Sammendrag

I arbeidet med denne oppgaven er det avdekket et stort behov for dynamisk informasjon og for tjenester som benytter dette i lokasjonsbaserte systemer. Dette behovet kommer fra brukerne selv, og stiller i sin tur større krav til håndtering og strukturering av denne informasjonen på en hensiktsmessig måte. Tid er et aspekt som også har vist seg svært viktig i forbindelse med lokasjonsbaserte tjenester, og er i så måte nært knyttet til dynamisk informasjon da dette er temporær informasjon.

Gazetteers ble funnet å være et svært velegnet rammeverk å bygge lokasjonsbaserte tjenester rundt da gazetteers i sin enkleste form støtter opp om flere av funksjonene som et slikt system krever, og kan enkelt utvides og bygges på for å legge til flere og mer avanserte funksjoner og tjenester.

Det er også funnet et behov for en felles standard for representasjon av lokasjonsbasert informasjon for å tilby lokasjonsbaserte tjenester, hvor det i tillegg bør finnes støtte for flere posisjoneringsteknologier og metoder. Den foreslåtte lokasjonsmodellen/dataformatet er et skritt på veien til å finne en slik standard, hvor en gazetteer danner grunnlaget.

I denne avhandlingen blir det sett på en rekke eksisterende og mulige løsninger på lokasjonsbaserte systemer og tjenester, og hvordan disse har blitt gjennomført og benyttet i praksis. Det blir identifisert viktige aspekter ved lokasjonsbaserte systemer og tjenester og den informasjonen disse skal håndtere. Aktuell teori på området rundt lokasjons -og kontekstbasert informasjon og tjenester blir gjennomgått, og metoder og teknologier for å posisjonere brukere av lokasjonsbaserte systemer blir presentert. Videre blir det presentert en lokasjonsmodell/dataformat basert på en gazetteer for å tilby lokasjons og kontekstbaserte informasjonstjenester. Denne modellen har blitt navngitt CULBIS - Context-User-Location Based Information Structure. Modellen baserer seg på den litterære studien av eksisterende lokasjons og kontekstbevisste systemer, øvrig arbeid gjort innen området, på selve gazetteer-strukturen og på en analyse av aktuelle informasjonselementer. Et mulig lokasjonsbasert system for studenter ved NTNU Gløshaugen blir foreslått, og en prototype blir implementert med nevnte lokasjonsmodell/dataformat som grunnlag. Det er til slutt utført en brukertest for å validere valgte funksjoner og tjenester og for å finne ut om modellen gir ønsket støtte for disse tjenestene.

Lokasjonsmodellen/dataformatet som er foreslått søker å integrere støtte for flere ulike typer lokasjonsbaserte tjenester fra forskjellige domener i ett felles rammeverk for representasjon av lokasjoner og informasjon. Det søker også å være åpent i den forstand at det kan konfigureres til å støtte individuelle behov fra en implementasjon til en annen. Dette innebærer også muligheten til å kunne benytte mer enn en enkelt posisjoneringsteknologi i ett og samme system.

## Summary

Through the work with this thesis there has been uncovered a great need for dynamic information and for services which utilize this in location based systems. This need originates from the users themselves, and in turn makes greater demands for handling and structuring of this information in an appropriate manner. Time is an aspect which also has proven itself of greater importance in connection with location based services, and is as such closely connected to dynamic information which is temporal.

Gazetteers were found to be a very suitable framework for building location based services upon, as gazetteers in its simplest form supports several of the functions such a system demands, and can easily be expanded and built upon to add further and more complex functions and services.

There has also been found a need for a common standard for representing location based information to offer location based services, in which there in addition should be support for several positioning technologies and methods. The proposed location model/data format is a step in the right direction for such a standard, featuring a gazetteer as its foundation.

In this thesis a number of existing and possible solutions for location based systems and services are reviewed, and how they have been executed and utilized in a real setting. Important aspects of location based systems and services are identified together with the information they handle. Relevant theory in the area of location and context based information and services are reviewed, and methods and technologies for positioning users of location based systems are presented. Furthermore, a location model/data format based on a gazetteer is presented to offer location and context based information services. This model, which is named CULBIS - Context-User-Location Based Information Structure, draw its inspiration upon the literary study of existing location and context aware systems, other work done in the domain, the gazetteer structure itself and an analysis of relevant information elements. A possible location based system for students at the NTNU Gløshaugen campus is proposed, and a prototype applying the aforementioned location model/data format is implemented. Finally, a user validation test is performed to validate selected functions and services, and to find out if the model produce the sought-after support for these services.

The proposed location model aims to integrate support for several different kinds of location based services from several different domains into one common framework for representation of locations and information. It also aims to uphold an open structure in the sense that it can be configured to suit individual needs from one implementation to another. This also includes the possibility to utilize more than one single positioning technology in one single system.



# Innhold

<b>Forord</b>	<b>i</b>
<b>Sammendrag</b>	<b>ii</b>
<b>Innholdsliste</b>	<b>v</b>
Figurliste . . . . .	viii
<b>1 Innledning</b>	<b>1</b>
<b>2 Problembeskrivelse</b>	<b>2</b>
2.1 Oppgaveformulering . . . . .	2
2.2 Mål for oppgaven . . . . .	2
2.3 Fremgangsmåte . . . . .	2
2.4 Avgrensing av oppgaven . . . . .	3
<b>3 Teoretisk bakgrunn</b>	<b>4</b>
3.1 Georeferert informasjon . . . . .	4
3.2 Spatial informasjonsbrowsing . . . . .	4
3.3 Gazetteers . . . . .	5
3.4 ADL Gazetteer . . . . .	7
3.4.1 ADL GCS . . . . .	7
3.4.2 ADL FTT . . . . .	8
<b>4 Lokasjonsbaserte tjenester</b>	<b>9</b>
4.1 Definisjon . . . . .	9
4.2 Teknologi . . . . .	10
4.2.1 GSM-nett basert lokalisering . . . . .	11
4.2.2 Beaconbasert lokalisering . . . . .	12
4.2.3 Satellittbasert lokalisering . . . . .	12
4.2.4 Radio/WiFi-basert lokalisering . . . . .	13
4.3 Kontekst . . . . .	14
4.4 Tjenestetyper . . . . .	15
4.5 Tjenesteparadigmer . . . . .	16
<b>5 State-of-the-art</b>	<b>17</b>

---

5.1	Eksisterende systemer . . . . .	17
5.1.1	GUIDE . . . . .	17
5.1.2	TouristGuide . . . . .	22
5.1.3	myCampus . . . . .	27
5.1.4	Cyberguide . . . . .	32
5.1.5	comMotion . . . . .	37
5.1.6	Andre liknende systemer . . . . .	42
5.2	Arkitekturer . . . . .	43
5.2.1	CyberGuide . . . . .	43
5.2.2	PoLoS . . . . .	44
<b>6</b>	<b>Lokasjonsbasert IS for studenter ved NTNU</b>	<b>48</b>
6.1	Tjenester og funksjoner . . . . .	48
6.2	Teknisk løsning . . . . .	50
6.3	Systemarkitektur . . . . .	51
6.4	Teknologi . . . . .	53
6.4.1	Posisjonering . . . . .	53
6.4.2	Klientutstyr . . . . .	53
6.5	Info i systemet . . . . .	55
<b>7</b>	<b>Prototype</b>	<b>56</b>
7.1	Analyse av informasjon . . . . .	56
7.1.1	Lokasjonsinformasjon . . . . .	57
7.1.2	Brukerinformasjon . . . . .	62
7.2	Lokasjonsmodell . . . . .	65
7.2.1	CULBIS . . . . .	65
7.3	Tjenester og Funksjoner . . . . .	70
7.4	Teknisk løsning . . . . .	88
7.5	Systemarkitektur . . . . .	89
7.6	Posisjonering . . . . .	91
7.6.1	WiFi Fingerprinting . . . . .	91
7.6.2	GPS . . . . .	93
7.7	Prototype klientapplikasjon - LOST . . . . .	94
7.7.1	Bruerklient . . . . .	95
7.7.2	Administrator . . . . .	95



---

7.7.3	Tjenester	96
7.8	Info i systemet	98
7.8.1	Testlokasjoner	98
7.8.2	Testinfo	99
7.9	Oppsummering	101
<b>8</b>	<b>Testing og evaluering</b>	<b>103</b>
8.1	Testgruppen	103
8.2	Testoppsett	103
8.3	Testdata	104
8.4	Spørreskjema	104
8.5	Intervju	104
8.6	Resultater	105
8.6.1	Oppsummering	108
<b>9</b>	<b>Konklusjon</b>	<b>110</b>
<b>10</b>	<b>Videre arbeid</b>	<b>112</b>
	<b>Referanser</b>	<b>114</b>
	<b>Appendix</b>	<b>117</b>
<b>A</b>	<b>- Brukermanual</b>	<b>117</b>
A.1	Installering	117
A.2	Bruk	119
A.2.1	Administrator	119
A.2.2	Brukerklient	123
<b>B</b>	<b>- Use Cases</b>	<b>129</b>
<b>C</b>	<b>- ADL GCS</b>	<b>136</b>
<b>D</b>	<b>- Skjema for lokasjonsmodellen</b>	<b>140</b>
<b>E</b>	<b>Fingerprint lokasjoner</b>	<b>142</b>
<b>F</b>	<b>- Spørreskjema</b>	<b>145</b>

## Figurer

1	Spatial informasjonsbrowsing . . . . .	5
2	GUIDE browser . . . . .	18
3	GUIDE klient - tabletPC . . . . .	19
4	GUIDE arkitektur . . . . .	20
5	GUIDE informasjonsmodell . . . . .	22
6	TouristGuide GUI . . . . .	23
7	TouristGuide klient - PDA . . . . .	24
8	TouristGuide HTML sider . . . . .	25
9	TouristGuide med flere lokasjoner . . . . .	26
10	TouristGuide lokasjonspolygon . . . . .	27
11	myCampus Restaurantveileder . . . . .	29
12	PanGO lokasjonsinfrastruktur . . . . .	30
13	myCampus arkitektur . . . . .	31
14	myCampus eWallet spørring . . . . .	31
15	CyberGuide klient - PDA . . . . .	33
16	CyberGuide grensesnitt for kart og informasjon . . . . .	34
17	CyberGuide grensesnitt for bar guiden . . . . .	35
18	CyberGuide posisjonering . . . . .	35
19	comMotion GUI for viktig lokasjon . . . . .	38
20	comMotion To-do liste . . . . .	39
21	comMotion GUI for melding . . . . .	39
22	comMotion GUI kart . . . . .	40
23	comMotion systemarkitektur . . . . .	41
24	PoLoS arkitektur . . . . .	45
25	PoLoS kernel arkitektur . . . . .	46
26	Ideell teknisk arkitektur . . . . .	50
27	Ideell systemstruktur . . . . .	51
28	Ideelt klientutstyr: PDA . . . . .	54
29	Ideelt klientutstyr: Laptop . . . . .	54
30	CULBIS konseptmodell . . . . .	68
31	CULBIS ER-diagram . . . . .	69
32	CULBIS Push tjeneste 1 . . . . .	73

---

33	CULBIS diagram Push tjeneste 1 . . . . .	74
34	CULBIS diagram Push tjeneste 2 . . . . .	75
35	CULBIS Push tjeneste 2 . . . . .	76
36	CULBIS Pull tjeneste 1 . . . . .	78
37	CULBIS diagram Pull tjeneste 1 . . . . .	79
38	CULBIS Pull tjeneste 2a . . . . .	80
39	CULBIS Pull tjeneste 2b . . . . .	82
40	CULBIS diagram Pull tjeneste 2b . . . . .	83
41	CULBIS Interact tjeneste 1 . . . . .	84
42	CULBIS diagram Interact tjeneste 1 . . . . .	85
43	CULBIS Interact tjeneste 2 . . . . .	86
44	CULBIS diagram Interact tjeneste 2 . . . . .	87
45	Prototype teknisk arkitektur . . . . .	88
46	Prototype systemstruktur . . . . .	89
47	Fumble Fingerprint score eksempel . . . . .	92
48	Prototype login and info skjerm . . . . .	94
49	Prototype lag fingerprint . . . . .	95
50	Lokasjon fingerprint plassering eksempel . . . . .	99
51	LOST info eksempel . . . . .	100
52	Webtjeneste eksempel . . . . .	100
53	Spørreskjema side 1 og 2 . . . . .	104
54	NetStumbler installasjon 1 . . . . .	118
55	NetStumbler installasjon 2 . . . . .	118
56	LostAdmin add location . . . . .	119
57	LostAdmin add info . . . . .	120
58	LostAdmin add fingerprint . . . . .	121
59	LostClient login and info . . . . .	123
60	LostClient kart . . . . .	124
61	LostClient interact and annotate . . . . .	125
62	LostClient 2do list and search . . . . .	126
63	LostClient event -and 2do notifier and info . . . . .	127
64	LostClient buddy and settings . . . . .	128
65	Skjema data prototyp . . . . .	140
66	Lokasjon fingerprint IT-bygget 1.etasje . . . . .	142

---

67	Lokasjon fingerprint IT-bygget 2.etasje . . . . .	142
68	Lokasjon fingerprint Sentralbygget Syd . . . . .	143
69	Lokasjon fingerprint Sentralbygget midtparti . . . . .	143
70	Lokasjon fingerprint Sentralbygget Nord . . . . .	144
71	Lokasjon fingerprint Realfagbygget . . . . .	144
72	Spørreskjema side 1 . . . . .	145
73	Spørreskjema side 2 . . . . .	146
74	Spørreskjema side 3 . . . . .	147
75	Spørreskjema side 4 . . . . .	148

# 1 Innledning

”Informasjonsalderen” er et begrep som har dukket opp for å beskrive tiden vi nå lever i. I kontrast til steinalderen hvor det viktige var å utnytte det nye redskapet stein til å forme hverdagen og fremtiden, er det i dag informasjon som er vår viktigste ressurs. Det snakkes mye om ”den nye informasjonshverdagen” og hvordan livene våre blir påvirket av ny teknologi og mengden informasjon tilgjengelig. For vi lever i dag i et informasjonssamfunn hvor alt baserer seg på å være raskt, enkelt og effektivt og man skal alltid være oppdatert og tilgjengelig. Dette er også blitt et krav fra forbrukere som ønsker å ha tilgang til informasjon og tjenester og være tilgjengelig over alt til enhver tid, om det enn er hjemme, på jobb, ferie eller fritid, eller på veien mellom noen av disse stedene eller aktivitetene.

Stadig ny teknologi dukker opp for både å skape ny informasjon og for å la oss utnytte de ressursene vi allerede har. Det har nå kommet til et punkt hvor dette integreres i hverdagen til den grad at vi ikke lenger tenker over at vi må skaffe til veie den informasjonen vi trenger, det forventes at den er der for oss. Og teknologien kan også hjelpe oss i å tolke når, hvor og hvordan vi ønsker tilgang til hvilke ressurser. En av disse mulighetene ligger i det faktum at teknologien nå kan posisjonere oss og dermed finne ut hvor vi befinner oss. Dette gir opphav til å ny rekke tjenester kalt lokasjonsbaserte tjenester.

Dagens litteratur og teknologimiljøet generelt har stor tro og fokus på lokasjonsbaserte tjenester, men det fokuseres for øyeblikket mest på den tekniske delen og selve lokasjonsbestemmelsen. I dette arbeidet har man kommet ganske langt. Men lokasjonen i seg selv har ingen større verdi, man må kunne knytte denne opp mot relevant informasjon for at det skal bli interessant og gi større nytteverdi. Det er også flere aspekter enn lokasjonen som må vurderes, siden denne typen tjenester er å regne som kontekstbaserte tjenester, og lokasjon bare er en del av en brukers kontekst. Hva som er relevant og nyttig informasjon vil også variere alt etter applikasjonsområdet til tjenesten og brukerne som skal benytte den. Dagens systemer og prototyper bærer preg av å ha en provisorisk informasjonsstruktur alt etter hva tjenesten skal tilby. I fremtiden vil det være ønskelig å kunne utveksle denne informasjonen slik at man kan ta den i bruk til flere forskjellige tjenester i flere forskjellige systemer og applikasjonsområder.

## 2 Problembeskrivelse

Hovedtema for denne oppgaven er lokasjons -og kontekstbaserte tjenester med størst fokus på lokasjon - hvordan vi kan strukturere data og informasjon for å støtte denne typen tjenester, hvilke typer data og informasjon som kreves og når/hvor/for hvem denne informasjonen skal være tilgjengelig, hvilke funksjoner og tjenester dette skal fasilitere og hvilke elementer et slik system vil kreve for å kunne realiseres.

### 2.1 Oppgaveformulering

I denne oppgaven tas det utgangspunkt i et mulig system for lokasjonsbaserte tjenester for nye studenter ved Gløshaugen som skal støtte studentene i deres opphold på campus gjennom kontekstrikkelig levering av informasjon. Det blir foreslått et dataformat for å støtte opp om nødvendig informasjon for å muliggjøre slike tjenester, og det vil i tillegg bli tatt i betraktning at systemet skal kunne både utvides og eventuelt overføres til andre applikasjonsområder. Naturlige spørsmål som da dukker opp er:

- Hva slags informasjon trenger vi for å lage et system for lokasjonsbaserte tjenester?
- Hvilke elementer må være til stede for å kunne opprette lokasjonsbaserte systemer?
- Hvilken type funksjoner bør man støtte og hva slags tjenester skal man tilby?
- Kan vi utnytte og/eller utvide egenskaper ved strukturer som gazetteers og tesauri til å støtte lokasjonsbaserte tjenester?

### 2.2 Mål for oppgaven

Målet er å finne ut hva slags informasjon et system for lokasjonsbaserte tjenester bør kunne håndtere for å kunne være et effektivt hjelpemiddel i hverdagen, samt hvordan denne informasjonen bør håndteres, struktureres, hvilke funksjoner og tjenester dette vil støtte og hvilke andre elementer som bør og må være til stede i et slikt system. [Effektivt hjelpemiddel er her definert som et hjelpemiddel som setter studenten i stand til å utføre oppgaver og gjøre mål raskere og med mindre forsinkelse]

### 2.3 Fremgangsmåte

For å nå målene er det gjennomført en litterær studie med fokus på å identifisere viktige aspekter ved planlegging, utvikling og implementering av lokasjonbaserte systemer og fasilitering av lokasjonsbaserte tjenester. Dette danner grunnlaget for

et dataformat baserte på en gazetteer, myntet på å støtte lokasjons -og kontekstbaserte tjenester, samt design av et mulig system. Det blir også sett på mulige alternative teknologier for å kunne posisjonere en bruker. Videre er det implementert en prototype av et lokasjonsbasert informasjonssystem for nye studenter ved NTNU Gløshaugen for å teste funksjoner og tjenester og finne ut hvorvidt dette er til hjelp for nye studenter ved et stort campus, og om det kan være til nytte i det daglige studentlivet.

## 2.4 Avgrensing av oppgaven

Lokasjonsbaserte tjenester er et problemområde som krysser mellom flere fagområder innen både data, tele, elektro og geografi. Blant disse kan vi nevne posisjoneringsteknologi, nettverk, ”ubiquitous computing” (allestedsnærværende databruk), kontekstbevisste systemer, Geographic Information Systems (GIS) m.fl. Hovedfokus for denne oppgaven ligger innenfor håndteringen av informasjon i et slikt system, og kobler dette opp i mot en del av kravene og forutsetningene som ligger til grunn for at et slikt system skal fungere, og for å støtte den faktiske bruken av systemet.

Datatilsynet stiller visse krav i forhold til Personvernloven vedrørende lagring og bruk av personlig informasjon i datasystemer. Dette er det sett bort i fra og ikke tatt hensyn til i denne oppgaven.

## 3 Teoretisk bakgrunn

I dette kapitlet vil det gis en kort innføring i diverse teori som danner grunnlaget for denne avhandlingen. En del konsepter som arbeidet bygger på og som er nyttige å forstå vil bli beskrevet. Lokasjonsbaserte tjenester er skilt ut som eget kapittel og vil derfor bli presentert i det påfølgende kapitlet.

### 3.1 Georeferert informasjon

Geografisk referert, eller georeferert informasjon [24] [16] er et samlebegrep på alle typer informasjon som på en eller annen måte henviser til eller er knyttet til en geografisk lokasjon på jordens overflate. Dette kan være ved å nevne eller henviser til geografiske stedsnavn eller områder i tekst, at et litterært verk er skrevet ved, handler om eller er utgitt ved et bestemt sted, bilder tatt av et bestemt sted eller eventuelt fra et bestemt sted, eller data som temperaturer, vindstyrker og andre værforhold samlet inn fra et bestemt sted. Geografisk informasjon av forskjellig art vil derfor også kunne kalles georeferert informasjon, men de to må ikke forveksles da de er to forskjellige begreper.

Vi kan skille mellom direkte og indirekte georeferert informasjon. Direkte georeferering er når det refereres til til en lokasjon gjennom geografiske koordinater som lengdegrad og breddegrad for å lokalisere et sted på jorda. Indirekte georeferering vil si når det refereres til en lokasjon gjennom det geografiske navnet for stedet som for eksempel Norge, Trondheim eller Gløshaugen.

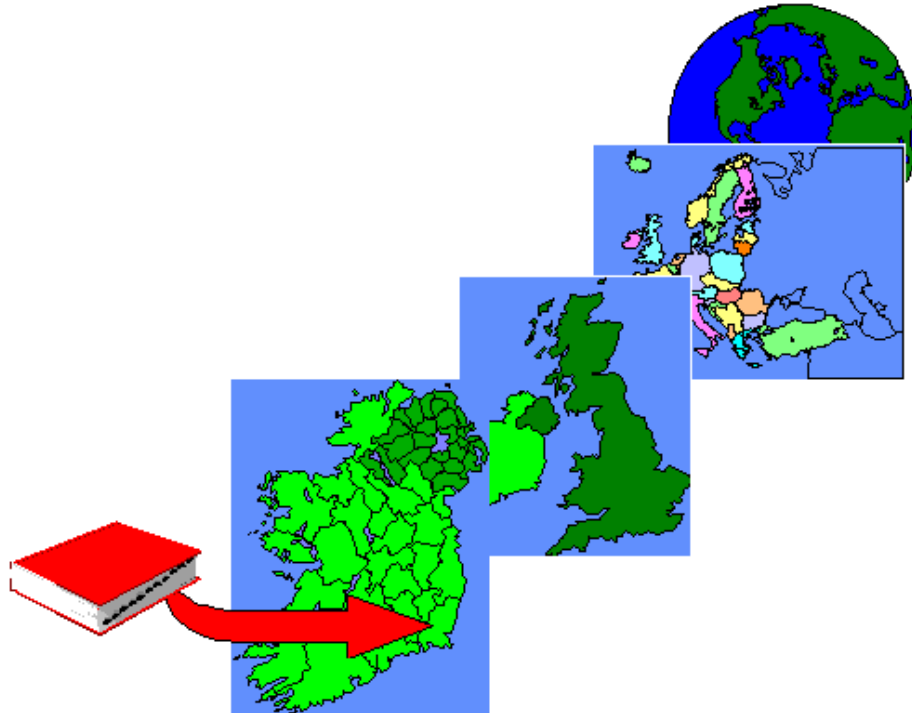
I rapporten "geoXwalk - A Gazetteer Server and Service for UK Academia" [27], gjengitt etter Marit Olsen [24], uttaler forfatteren James Reid at britiske "National Geographic Data Framework" (NGDF) estimerer at omtrent 80 prosent av all registrert informasjon i UK på en eller annen måte er georeferert. Uansett andel er dette et faktum som kan utnyttes når det gjelder digitale bibliotek og informasjonssystemer. Dette leder oss videre til neste delkapittel.

### 3.2 Spatial informasjonsbrowsing

Browsing er et uttrykk som ikke har noe godt norsk ord, men betyr å skimme eller se igjennom noe eller lete seg fram til, og blir ofte koblet sammen med det å "surfe" gjennom nettsider på Internett. Spatial informasjonsbrowsing betyr derfor altså å "surfe" gjennom informasjon som er sortert, klassifisert eller på annen måte organisert etter hvor informasjonen refererer til. Dette er et konsept som er blitt mer og mer aktuelt ettersom digitale bibliotek blir større og større, og gir en litt annen innfallsvinkel for å imøtekomme informasjonsøkerens behov, spesielt i tilfeller hvor dette behovet ikke fremstår klart for søkeren, og han ikke vet helt hvordan han skal formulere dette i en spørring til et aktuelt system. Et eksempel på slik browsing er illustrert i figur 1 [16]. Her ser vi en sekvens med kart som kan bli presentert for en person som leter gjennom et digitalt bibliotek, hvor han går fra et globalt perspektiv ned til en oversikt over Europa, videre til UK, så til



Irland og velger til slutt et ikon på kartet over Irland som representerer en bok med informasjon om det valgte området.



Figur 1: Eksempel på spatial informasjonsbrowsing

### 3.3 Gazetteers

Selv om de fleste av oss kanskje ikke har hørt ordet gazetteer før har mange av oss brukt det, kanskje hjemme eller kanskje i geografitimene på grunnskolen. Indeksen bakerst i vanlige atlas inneholder navn på steder og informasjon om hvor i atlaset og hvor på kartene man finner de, ofte i form av koordinater. Mange atlas inneholder også tilleggsinformasjon om stedene, så som areal, populasjon, historie og annen informasjon som er med på å beskrive det aktuelle stedet. Disse postene i indeksen kalles gazetteers.

En gazetteer er med andre ord en liste eller samling av geografiske navn sammen med informasjon om geografisk lokasjon og eventuelt annen informasjon knyttet til lokasjonen. Slike geografiske steder inkluderer mer eller mindre alle steder vi kan sette navn på, slik som land, byer, politiske og administrative områder, menneskeskapte områder som bygninger og monumenter, naturfenomener og mange flere. Slike steder kan og vil til tider overlappe hverandre. Videre utvikling av slike grunnleggende gazetteers har resultert i digitale gazetteers som har mulighet for enda større bruksområder enn oppslag i atlas. En digital gazetteer får gjerne mer en form som en informasjons- eller kunnskapsbase som minimum definerer geografiske navn ved spatial representasjon (m.a.o. koordinater). Det vi kan kalle kjerneelementene i en digital gazetteer er stedsnavnet, den spatiale representasjo-

nen som et avtrykk på jordens overflate og en beskrivelse av hva slags type sted det er [24].

<b>Geografisk navn</b>	- Offisielt navn på stedet.
<b>Lokasjon</b>	- Dette tar form som koordinater som representerer et punkt, en linje eller et areal.
<b>Navnetype</b>	- En klassifisering av typen sted valgt fra et kontrollert skjema.

I tillegg har digitale gazetteers muligheten til å uttrykke spatiale relasjoner eksplisitt, som for eksempel "lokasjon A" *erDelAv* "lokasjon B" eller "lokasjon B" *harDeler* "lokasjon A".

En digital gazetteer kan brukes integrert i andre systemer som hjelp til automatisk oppslag av steder og lokasjoner eller som en selvstendig informasjonsressurs. Hva slags informasjon som representeres i en gazetteer i tillegg til kjerneelementene og detaljeringsgraden av denne kan variere fra gazetteer til gazetteer og avhenger av hva man skal bruke gazetteeren til. Man må også ta stilling til hvilke typer steder som skal kunne representeres, som land, byer, bygninger, elver osv, ved å definere navnetypeskjemaet.

For å bli litt mer konkret kan vi spørre hva en digital gazetteer egentlig kan gjøre for oss. For å svare på dette kan vi sette opp en liste over funksjoner som gazetteeren vil støtte: [24]

- Svare på spørsmål som *hvor er ... ?*
- Bestemme typen sted
- Gjøre oversetting mellom geografiske navn og posisjoner på jordens overflate
- Finne samlingsobjekter ved å matche geografisk posisjon med avtrykk for samlingsobjekter
- Beregne avstand mellom steder ved hjelp av koordinater
- Et oppgitt avtrykk kan finne et sett med flere steder i området
- Bestemte typer steder kan identifiseres innenfor gitte områder

I tillegg til disse basisfunksjonene som kan tilbys gjennom basiselementene kan en gazetteer:

- Tilby flere variable navn for et sted
- Oppgi forskjellige spatiale representasjoner så som punkt, linje, omsluttende boks eller definert areal
- Peke til andre typer informasjon
- Peke til andre relaterte gazetteer oppslag med lenker som *erDelAv* eller *erHovedstadI*
- Kombinere informasjon fra varierte kilder med henvisning til kilden for hver bidragsyter

### 3.4 ADL Gazetteer

Da ”The Alexandria Digital Library Project” startet i 1994 ble et initiativ til å skape sin egen gazetteer raskt påbegynt. Etter en del datainnsamling, bruk og eksperimentering ble det etterhvert avdekket at mangelen på en felles struktur, standard eller format skapte problemer. Dette førte i sin tur til utviklingen av ADL Gazetteer Content Standard (ADL GCS) som skulle bidra til å løse problemene som ble avdekket. Standarden har vært under utvikling siden og er blitt ganske omfattende. Den kan være en god pekepinn for utvikling av nye gazetteers. ADL Gazetteer er realisert i form av en relasjonsdatabase-implementasjon basert på GCS spesifikasjonene.

#### 3.4.1 ADL GCS

ADL Gazetteer Content Standard (GCS) er skapt for å være et utfyllende rammeverk for ta vare på beskrivelser av navngitte geografiske steder, inkludert historiske data for stedene, spatial lokasjon i forskjellige representasjoner, klassifisering av sted i forhold til et typeskjema og kildehenvisning til deler av beskrivelser samlet fra forskjellige kilder for forskjellige steder. Dette skal være med å bidra til en bedre forståelse for slike data og fostre skaping, bruk og deling av disse dataene mellom lokale, nasjonale og internasjonale organisasjoner. Den skal i tillegg støtte internasjonale og flerspråklige bruksområder. Den er også bygd for å kunne kobles til andre informasjonskilder angående et sted. Et annet underliggende formål er å rette fokus på de beskrivende komponentene for navngitte geografiske steder.

Seksjonene i standarden håndterer

- Navn og detaljer om deres opprinnelse, språk og bruk
- Klassifisering (i forhold til et angitt skjema)
- Koder og identifikatorer assosiert med stedet (f.eks. FIPS koder)
- Spatial lokasjon (koordinater for omsluttende boks og detaljerte geometrier)
- Gateadresser
- Forhold til andre navngitte steder
- Data (f.eks. befolkning, høyde, areal etc.)
- Beskrivelse (fortellende)
- Linker til eksterne ressurser angående lokasjonen
- Annet: supplerende notat; utvidet metadata

#### ADL Gazetteer Content Standard

Det er 14 hovedelementer i GCS, hvorav fem er obligatoriske (sett bort i fra det obligatoriske kildeelementet) og står med uthevet skrift i listen. Elementene merket med (R) er repeterbare. Ikke-obligatoriske elementer kan ha obligatoriske poster hvis de først blir benyttet. En komplett oversikt over elementene finnes i Appendix.

1. **Geographic Feature ID**
2. **Geographic Name**
3. Variant Geographic Name (R)
4. **Type of Geographic Feature** (R)
5. Other Classification Terms (R)
6. Geographic Feature Code (R)
7. **Spatial Location** (R)
8. Street Address
9. Related Feature (R)
10. Description
11. Geographic Feature Data (R)
12. Link to Related Source of Information (R)
13. Supplemental Note
14. **Metadata Information**

### 3.4.2 ADL FTT

ADL Feature Type Thesaurus, eller feature type thesaurus generelt, er en klassifiseringsliste over typer steder og representerer en karakterisering av et sted organisert i et hierarkisk skjema. En slik liste kan variere i detaljgrad og dybde i hierarki. ADL FTT er svært omfattende og dekker de mest tenkelige og utenkelige typer steder, alt fra "befolkede områder" og "byer" til "kanaler", "skytebaner", "landbruksområder" og "vulkaner". Antallet topptermer er 6 og antallet foretrukne termer er 210. Det er noe over 1000 ikke-foretrukne termer. FTT er laget for å kontrollere vokabularet og gjennom dette veilede en bruker til å finne den typen sted han leter etter uten at det blir for mange varianter å bruke for hvert enkelt oppslag.

## 4 Lokasjonsbaserte tjenester

Lokasjonsbaserte tjenester er et forholdsvis nytt begrep som har mottatt masse lovord, men som ennå ikke har kommet seg opp på et nivå hvor det har kommet inn i teknologi -eller informasjonshverdagen. Teknologien for å bestemme en persons lokasjon har nå eksistert i lengre tid, det kommer stadig nye og bedre metoder å gjøre dette på, og prisene nærmer seg et nivå hvor det vil være aktuelt å ta dette i bruk i hverdagen til vanlige mennesker. Men hva er egentlig en lokasjonsbasert tjeneste? For få klarhet i hva begrepet omfatter starter vi med noen definisjoner.

### 4.1 Definisjon

*Hva er lokasjonsbaserte tjenester? I programmet FBI på NRK1 ble det onsdag 21. mars -01 vist et system som via satellitt gjorde det mulig for et forsikringsselskap å finne ut hvor en bil befant seg til enhver tid. Også annet teknologi, for eksempel mobiltefonteknologi, kan på lignende måte brukes til å fastslå hvor bæreren av mobilen befinner seg. Dette vil trolig danne grunnlag for at aktører i telemarkedet utvikler tjenester som er basert på kunnskapen om hvor en mobilbruker oppholder seg. Vi kan kalle dette "lokasjonsbaserte tjenester".*

- Dag Wiese Schartum, Avdeling for forvaltningsinformatikk ved UiO

*Location services are defined as applications that deliver location-based information where and when it is needed. Users can access these services via the desktop, Web browser, mobile phone, personal digital assistant (PDA), pager, or other device. Diverse applications include fleet tracking, emergency dispatch, sensor monitoring, and customer location for target marketing and advertising, roadside assistance, stolen vehicle recovery, navigation, directory services, and more. [29]*

- Sumit Sen og Smita Sengupta, Applied Technology Group, Tata Infotech Limited

*Location-based Services (LBSs) are mobile services for providing information that has been created, compiled, selected or filtered under consideration of the users' current locations or those of other persons or mobile devices. Typical examples are restaurant finders, buddy trackers, navigation services or applications in the areas of mobile marketing and mobile gaming. The attractiveness of LBSs is due to the fact that users are not required to enter location information manually but are automatically pinpointed and tracked.*

- Axel Küpper, forfatter av boka "Location-based Services: Fundamentals and Operation" [15]

Videre er følgende uttalelse fra Nickolai Roland til god hjelp for å definere hva essensen i lokasjonsbaserte tjenester er:

*Lokasjonsbaserte tjenester vil komme i mange varianter. Lokaliseringen kan baseres på forskjellige teknologier hvilket betyr at posisjonsbestemmelsen kan være av forskjellig nøyaktighet. Tjenestene som blir utviklet må derfor ta hensyn til dette da de kanskje ikke er interessante dersom lokasjonen blir for unøyaktig. Lokasjon i seg selv er ikke så mye verdt. Det er først når man kan assosiere lokasjonen med annen informasjon at det gir mening. Lokasjonsbaserte tjenester kan være så mangt, men en hovedutfordringen vil uten tvil være å kople lokasjonsinformasjonen opp mot annen informasjon og presentere resultatet på en god måte.*

- Nickolai Roland, Fakultet for Elektroteknikk og Telekommunikasjon ved NTNU

Som de fleste vet finnes det opptil flere teknologier for å bestemme en brukers posisjon, den meste kjente av dem er GPS. I tillegg til å kunne oppgi lokasjonen til en bruker av slik teknologi kan dette benyttes i andre informasjonssammenhenger, som f.eks. å tilby informasjonsressurser basert på denne posisjoneringen. Slik informasjon kan være alt fra reklame til nytteinformasjon og underholdning. Dette gir muligheter til en helt ny dimensjon innen kontekst -og brukerspesifikk distribusjon av informasjon, og kan kombineres med andre typer filtreringsteknologi som for eksempel brukerprofiler. Det neste som derfor er viktig er å lære hvordan vi kan knytte denne lokasjonen til relevant og nyttig informasjon. Hvis dette kan kombineres for å gi en bruker nyttig og viktig informasjon der han er kan dette lette hverdagen for mange mennesker i svært mange sammenhenger. Man ville kunne slippe å måtte oppsøke informasjonen der man er eller eventuelt lete andre steder, informasjonen ville i stedet komme til deg der du er, når du er der, når du trenger den. Dette kan bidra samfunnsmessig til økt produksjon, bedre sikkerhet, større tilgjengelighet og bedre forståelse for ting rundt oss i hverdagen.

Basert på dette kan vi formulere en egen enkel og generell definisjon som vil brukes som utgangspunkt:

*Lokasjonsbaserte tjenester er ressurser som tilbys en bruker i form av informasjon som er skapt, samlet, valgt ut eller filtrert med grunnlag i lokasjonen til den aktuelle brukeren eller brukerne, og som er behandlet med nettopp dette i tankene.*

## 4.2 Teknologi

For å kunne finne lokasjonen hvor bruker befinner seg trengs teknologi som gjør dette, og helst gjør det så raskt, billig og presist som mulig. Det finnes flere teknologier og medhørende teknikker som har forsøkt å gjøre dette, og utsiktene

blir stadig bedre. Det vil her presenteres en kort oversikt over et lite utvalg av de fremste teknologiene og metodene for posisjonering [25] [3].

#### 4.2.1 GSM-nett basert lokalisering

Denne typen lokalisering baserer seg på signalene i et mobiltelefonnett. Ganske enkelt sagt tar denne teknologien utgangspunkt i den faste statiske posisjonen til mobilmastene, og bruker dette til å anslå eller beregne posisjonen til tilkoblet mobil enhet, i de fleste tilfeller en mobiltelefon.

##### CGI - Cell Global Identity

Denne metoden benytter et svært enkelt prinsipp, og baserer seg på at dersom en mobil enhet har kontakt med en aktuell mobilmast må den befinne seg innenfor mastens senderekkevidde. Presisjon vil avhenge av rekkevidden på masten, men uansett gir denne teknikken i beste fall forholdsvis dårlig presisjon, et sted i mellom 300 meter i tettbygde strøk til 3,5 km i utkantstrøk. En fordel er at denne metoden kan benyttes uten noe videre beregning eller ekstra implementasjon.

##### CGI + TA - Cell Global Identity + Timing Advanced

Dette er en metode som forbedrer presisjonen til vanlig CGI posisjonering. Det den gjør er å legge til en TA (Timing Advanced) variabel, som ganske enkelt måler forsinkelsen i signalet i tid mellom den mobile enheten og mobilmasten. Den kan gjennom dette anslå omtrent hvor langt unna masten klienten befinner seg og dermed øke presisjonen. Selv om dette er en forbedring over den forrige metoden har den fortsatt ganske dårlig presisjon, fra rundt 100 meter til opptil 3,5 km.

##### UL-TOA - UpLink Time Of Arrival

Denne metoden, som også går under navnet Time Difference of Arrival (TDOA), er en trianguleringsteknologi som krever tre eller flere mobilmaster. Den måler hvor lang tid signalet bruker på å komme fram til hver av mastene, og beregner ut i fra dette hvor klienten befinner seg. For å gjøre dette kreves ekstra utstyr for måling kalt Location Measurement Unit (LMU) og beregning av posisjon basert på målingene kalt Mobile Positioning Centre (MPC) for hver mast. Denne teknologien vil med andre ord bare være effektiv på steder hvor man har høy grad av overlapping mellom dekningssoner slik at man er i stand til å måle tre eller flere master samtidig. I tillegg vil den kreve kostbart tilleggsutstyr. Presisjonen vil også variere etter hvor mange master man har kontakt med, best målte resultat med 7 master på 38 meter til rundt 500 meter med 3 master.

Det finnes også flere hybride metoder som Enhanced Observed Time Difference (E-OTD) og dens 3G ekvivalens Observed Time Difference of Arrival (OTDOA), som både er billigere og delvis mer presise enn de overnevnte, men er mye mer kompliserte, krever mye prosesseringskraft og vil være avhengig av aktiv innsats fra både klientenheten og nettverket selv. Siden slike posisjoneringsteknologier

både er kostbare og krever at man har kontroll over hele mobilnettverk vil ikke disse bli sett nærmere på i denne oppgaven.

#### 4.2.2 Beaconbasert lokalisering

Beaconbasert lokalisering går ut på at såkalte beacons sendes ut fra bestemte steder man ønsker å kunne posisjonere. En klientenhet vil fange opp et slikt beaconsignal når den er innen rekkevidde og dermed vite hvor den befinner seg.

##### **IR - Infrarød**

Infrarøde beacons vil sende ut et infrarødt signal som identifiserer lokasjonen hvor beaconet er plassert. Det kreves at den infrarøde senderen sender ut en predefinert bitsekvens som tilsvarende en unik identifikator. Denne bitsekvensen må være lang nok til å støtte det antallet lokasjoner man ønsker å kunne identifisere. Man kan med dette få en presisjon på noen få meter, men man er avhengig av fri sikt direkte mellom senderen og mottakeren. Denne metoden skalerer også svært dårlig siden man må ha ett beacon per lokasjon.

##### **Bluetooth**

Siden Bluetooth er en teknologi som baserer seg på automatisk oppdaging av enheter egner den seg til denne typen applikasjon. Det er også en teknologi som er godt standardisert og utbredt blant flere typer enheter som mobiltelefoner, PDA, bærbare datamaskiner, printere osv. Fremgangsmåten for denne metoden er at en Bluetooth enhet vil representere et område eller en lokasjon, slik at når en klientenhet kommer innen rekkevidde vil den fange opp signalet og identifisere hvor den befinner seg. Dette vil også virke andre veien slik at en enhet som representerer en lokasjon kan fange opp at en klientenhet nettopp kom innen rekkevidde. Bluetooth krever ikke fri sikt, men vil være noe dyrere enn infrarød. Det vil kreves en Bluetooth enhet per lokasjon. Presisjonen vil være rundt 10 meter siden det er den generelle rekkevidden på signalet. Med teknikker for overlapping av signaler kan denne presisjonen økes ytterligere.

#### 4.2.3 Satellittbasert lokalisering

Dette er trianguleringsorientert lokalisering ved hjelp av satellitter. Siden den eneste sivile formen for slik posisjonering foreløpig er GPS, selv om den europeiske Galileo er underveis, vil dette avsnittet hovedsaklig se på dette.

##### **GPS - Global Positioning System**

GPS baserer seg på signalet fra minst 4 satellitter som mottas av en GPS klient som beregner sin lokasjon ut ifra dette. Det kreves fri sikt til disse 4 satellittene, noe som ikke alltid er mulig i urbane strøk og innendørs. Dette er et minus siden det er ofte er i slike omgivelser man ønsker å benytte mobile og lokasjonsbaserte tjenester. Et annet minus er at det ofte tar ganske lang tid å få inn en posisjon



siden klienten må finne satellittene. Hvis man mister noen av de under bruk må de søkes opp igjen før man kan posisjonere. Presisjonen er veldig god, offisielt 4-10 meter, men de fleste opplever en presisjon på rundt 2-5 meter.

### **A-GPS - Assisted Global Positioning System**

Assisted GPS prøver å forbedre tiden det tar å posisjonere klienten. Dette gjøres ved hjelp av mobilnettet som kan gi klienten en efemeride, en astronomisk tabell, som forteller hvor de nærmeste satellittene befinner seg, og dermed sparer klientene å søke de opp. LMU'er plassert ved mobilmastene har GPS enheter som kontinuerlig følger satellittene og sender fra seg efemeriden når klienter har behov for det.

### **D-GPS - Differentiated Global Positioning System**

Differentiated GPS er en metode for å forbedre presisjonen på GPS posisjoneringen gjennom å koblet sammen flere stasjonære GPS mottakere i et bakkebasert nettverk som beregner differansen seg imellom, og dermed tilbyr enda mer nøyaktig posisjonering.

Et stor pluss for GPS er presisjon og dekningsområde som dekker hele jorda. Minuset som er dårlig eller ingen dekning i urbane omgivelser og innendørs er vanskelig å overkomme for de fleste lokasjonsbaserte tjenester.

## **4.2.4 Radio/WiFi-basert lokalisering**

Det finnes et stort antall teknikker for posisjonering basert på radiosignaler, hvorav mobilnettposisjonering benytter en del av disse. De teknikkene som blir nevnt i dette avsnittet omhandler radiosendere av mindre kalibre og er hva vi for det meste kjenner som WiFi.

### **Trianguleringsteknikker**

Triangulering er et velkjent prinsipp som lenge har vært brukt innen radioteknologi, og inngår også i GSM-basert posisjonering. Kort sagt baserer dette seg på signalet fra to eller flere stasjonære signalkilder man vet lokasjonen til og målinger av disse. JO flere stasjonære kilder dess bedre og mer presis posisjonering. Disse dataene kan behandles med trigonometriske og geometriske beregninger for å bestemme posisjonen til den siste signalkilden, brukers klientutstyr. For WiFi-utstyr vil disse stasjonære signalkildene være aksesspunktene i det trådløse nettet og klientutstyret vil være bærbare pc'er, PDA'er og smarttelefoner med trådløs tilkobling.

### **WiFi Location Fingerprinting**

Også kjent som Location Pattern Matching. Denne metoden går ut på å trene opp et system til å gjenkjenne likheter i radiosignalet og gjennom statistiske

utregninger finne beste treff for en lokasjon. Signalprøver hentes inn, behandles og lagres som et ”fingeravtrykk” slik at de kan sammenlignes mot signalet fra en nomadisk enhet og gjennom dette bestemme hvor brukeren befinner seg.

### **Cordis RadioEye™**

RadioEye [8], eller ”radioøyet”, er et integrert posisjoneringssystem utviklet av Radionor Communications, et norsk selskap med base i Trondheim. Dette består av en radiobasert enhet med en mengde små sendere og mottakere integrert som er i stand til å posisjonere radiobaserte klientenheter med en presisjon på ned til +/- 50 cm ifølge deres egne tester. Radionor jobber også med å integrere små mikro-mottakere i SIM-kort for bruk sammen med mobiltelefoner. Vi vil sannsynligvis få se mere til dem de neste årene.

## **4.3 Kontekst**

Lokasjonsbaserte tjenester og lokasjonsbevisste systemer er termer som kommer inn under et bredere begrep som kalles kontekstbevissthet og kontekstbevisste systemer. Når mennesker kommuniserer med hverandre har vi en egen evne til å oppfatte og forstå ideer og overføre de til hverandre og reagere hensiktsmessig på forskjellige situasjoner i forskjellige omgivelser gjennom vår forståelse for hvordan verden fungerer. De tingene vi tar i betraktning i slike situasjoner er det vi kaller kontekst. Datamaskiner derimot har ikke denne evnen til å oppfatte situasjoner og omgivelser på samme måte. Kontekstbevisste systemer søker derfor å samle inn data og informasjon som kan sette dem i stand til å tolke konteksten på en hensiktsmessig måte og bruke den til å berike tjenestene og minimere inputen fra brukerne.

Definisjon kontekst:

*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. [2]*

- Anind K. Dey og Gregory D. Abowd, Graphics, Visualization and Usability Center and College of Computing, Georgia Institute of Technology, Atlanta

Definisjon kontekstbevissthet:

*Context-awareness is a term that describes the ability of the computer to sense and act upon information about its environment ... [19]*

- Natalia Marmasse, MIT Media Laboratory

For å simplifisere dette litt kan vi si at kontekst er brukerens omgivelser, personlige og generelle situasjon og applikasjonens hensikt, og at kontekstbevissthet er systemets evne til å oppfatte/måle den hensiktsmessige konteksten, tolke/behandle den og reagere hensiktsmessig i forhold til dette.

Ideen om kontekstbevisste systemer kommer igjen fra konseptet ”ubiquitous computing”, eller allestedsnærværende databruk. Dette konseptet, eller fagfeltet som det har utviklet seg til, går ut på at vi har tekniske ”dingser” og datamaskiner overalt rundt oss i hverdagen. Disse er gjerne små og mer eller mindre usynlige for brukerne, kan kommunisere med hverandre og blir benyttet mer eller mindre ubevisst. De setter oss i stand til å være mer effektiv i hverdagen og beriker omgivelsene og situasjonene vi er i. Definisjonen fra thefreedictionary.com [31] beskriver det som den tredje bølgen av databruk, hvor den første var flere personer per datamaskin, den andre én datamaskin per bruker og den tredje flere datamaskiner per bruker. Ifølge Mark Weiser [32], kjent som faderen til ubiquitous computing, går datamaskinene over fra å bli benyttet bevisst til å gli inn i bakgrunnen og bli en del av omgivelsene.

Dermed har vi plassert lokasjonsbaserte systemer og tjenester i et teknologisk hierarki, hvor disse begrepene både kan og vil sammenfalle i en del sammenhenger, og vil påvirke hverandre:

Ubiquitous Computing ->Context-aware Systems ->Location-aware Systems and LBS

## 4.4 Tjenestetyper

Basert på kunnskap om lokasjon har vi et sett tjenestetyper vi kan kategorisere tjenester innenfor.

### Lokalisering

Vi kan fortelle en bruker hvor han eller andre befinner seg og illustrere dette i form av f.eks. kart. Enkel bruk av GPS enheter er et eksempel på dette. Vi har også sett GSM-tjenester som tillater andre å spore hvor en bekjent befinner seg.

### Veibeskrivelse

Tjenester kan benyttes til å finne veien fra stedet du befinner deg til stedet hvor du ønsker å dra. Slike systemer finnes bl.a. for biler, og kan vise deg veien på et kart eller fortelle deg hvor du skal svinge mens du kjører rundt i byen.

### Flåtestyring

Selskaper med mobile ”feltarbeidere” som for eksempel fraktebyråer med fraktebiler, buss-selskaper med busser, taxiselskaper med taxier o.l. kan bruke flåtestyring hvor en overordnet enhet får automatisk rapportert hvor de forskjellige delene av deres flåte befinner seg, og kan organisere og distribuere den basert på denne informasjonen.

### Enhetssporing

Denne typen tjeneste brukes mye i frakteindustrien til å holde rede på kolli som sendes rundt i verden. Disse spores ofte ved hjelp av RFID teknologien, men man kan også kalle nettsporingen i den norske Posten, som benytter strekkoder, for en slik tjeneste.

### Point-of-Need

Levering av informasjon for området man befinner seg i kan kalles en Point-of-Need

tjeneste. Man kan for eksempel bruke mobiltelefonen og be om værmeldingen og automatisk få værmeldingen for området man befinner seg i. Et annet eksempel er reklame for en butikk eller et kjøpesenter man passerer. Dette kombineres gjerne med personlig tilpasning gjennom en brukerprofil. Automatiske turistguider vil også havne under denne kategorien.

## 4.5 Tjenesteparadigmer

Når man skal utvikle og bruke lokasjonsbaserte tjenester må disse tjenestene komme i en eller annen form, tjenestene må ha en eller annen måte å forhandle og interagere med brukeren på. Her presenteres kort de paradigmenes som har blitt identifisert.

### **Publish/subscribe**

Denne paradigmen har blitt benyttet siden boktrykkerkunsten ble oppfunnet og sannsynligvis også lenger før det. Det går ut på at man abonnerer på en bestemt ressurs, for eksempel en avis, og når denne blir utgitt får man den levert. I den elektroniske verden er dette kjent som blant annet nyhetsbrev som man kan abonnere på med e-post.

### **Pull**

Dette er den klassiske informasjonshentings-paradigmen. Brukeren spør etter en tjeneste, oppgir eventuelle kriterier og får levert tjenesten eller informasjon han etterspør. For eksempel kan en bruker sende en SMS hvor han spør hvor han befinner seg. Systemet svarer da med å gi ham navnet på området han befinner seg i.

### **Push**

Pull var første generasjon når det gjelder informasjonslevering, push er neste steget videre. Push av en tjeneste går ut på at tjenesten leveres automatisk til brukeren når visse kriterier for levering av tjenesten er møtt. Dette er i tillegg til en videreføring av pull en utvikling innenfor publish/subscribe paradigmen hvor leveringen ikke lenger avhenger av når noe blir publisert, men avhenger av flere faktorer. I et lokasjonsbasert system vil dette bl.a. være når brukerens lokasjon stemmer over ens med tjenestens tjenesteområde. Så hvis en bruker benytter en tjeneste som forteller ham når han er i nærheten av folk i kontaktlisten sin vil dette bli gjort automatisk når dette kriteriet er møtt.

### **Interact**

En bruker vil i mange tilfeller kunne interagere direkte med en tjeneste og også påvirke innholdet i disse tjenestene. Ett eksempel kan være i intelligente omgivelser hvor man får tilgang til å styre temperaturen i et rom med en håndholdt enhet når man befinner seg i det aktuelle rommet. Et annet eksempel kan være å bruke online booking når man er i nærheten av et hotell, eller elektronisk kjøp av kinobilletter ved kinoen. Det vil også innebære at man kan være forfatter av informasjon i et system, noe mange har erfaring fra med blant annet elektroniske forum.

## 5 State-of-the-art

I dette kapitlet blir det presentert arbeid som har blitt gjort innen området med lokasjonsbaserte tjenester og systemer. Dette vil gi et bilde hvor langt man er kommet innen utvikling av lokasjonsbaserte tjenester og gi en pekepinn på hvordan en implementering av et liknende system kan gjennomføres.

### 5.1 Eksisterende systemer

Denne delen ser på allerede eksisterende systemer som tilbyr lokasjonsbaserte informasjonstjenester. Det er her valgt å se nærmere på følgende systemer: GUIDE, TouristGuide, MyCampus, CyberGuide og ComMotion. I tillegg er systemene Stick-E-Notes og RADAR nevnt. Systemene er presentert i forhold til brukerne av systemet, bruksområde, funksjonalitet i systemet, presentasjon av system og tjenester, anvendt teknologi og informasjon i systemet.

#### 5.1.1 GUIDE

GUIDE, eller The GUIDE Project, er et kontekstbevisst hypermedia informasjonssystem for håndholdte enheter utviklet av Distributed Multimedia Research Group ved Department of Computing, Lancaster University. Systemet fungerer som en kontekst- og lokasjonsbevisst turistguide for byen Lancaster, England, og er et samarbeid mellom universitetet og Tourist Information Center (TIC) i Lancaster.

**Brukere:**

Systemet er ment til bruk for turister i byen Lancaster. Turister kan få utdelt en håndholdt enhet klargjort for bruk ved Tourist Information Center i byen.

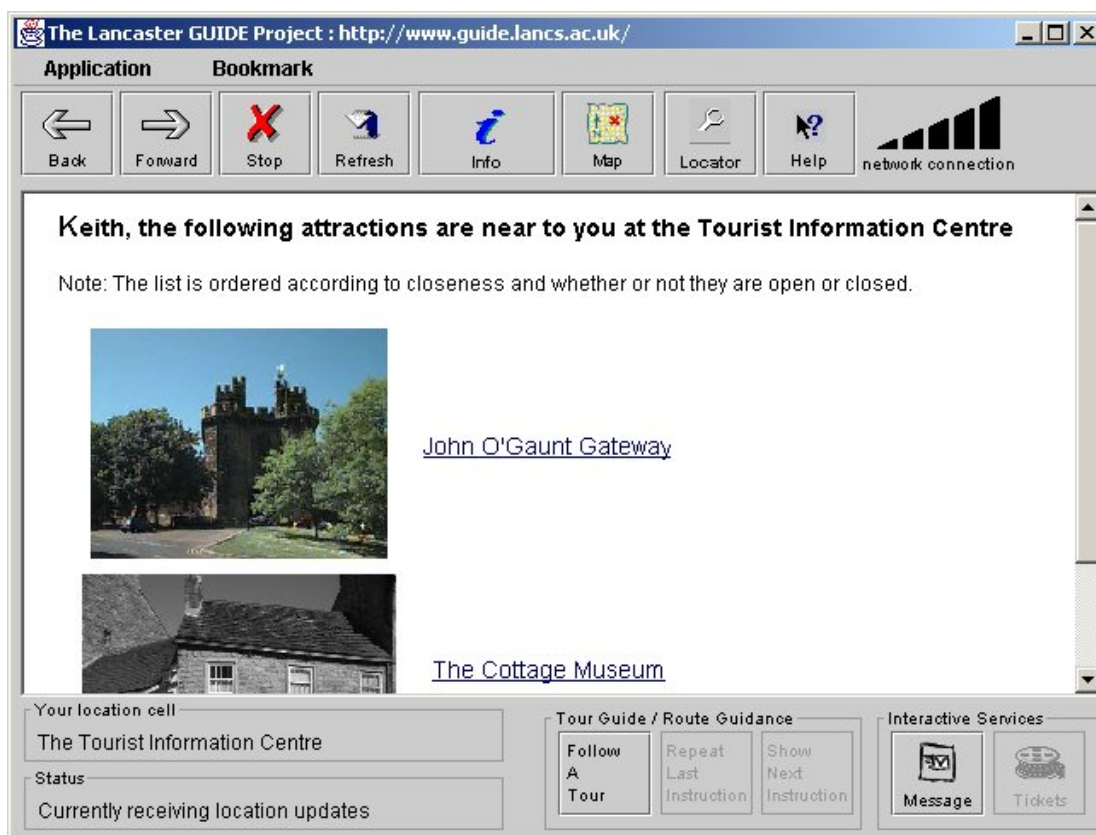
**Bruksområde:**

Turistene kan få informasjon om steder og severdigheter i Lancaster basert på hvor de befinner seg i byen, og eventuelle interesser de måtte ha som turister. Systemet kan også brukes til billettbestilling og reservasjoner.

**Funksjonalitet:**

Hovedfunksjonaliteten til GUIDE er å presentere kontekstrikkelig turistinformasjon om severdigheter og attraksjoner i Lancaster basert på lokasjon og brukerprofiler. Den gjør dette ved å vise informasjon i en modifisert web browser på en mobil håndholdt enhet. Filtreringsfunksjoner basert på informasjonsmodellen vil rangere de nærmeste attraksjonene basert på diverse kriterier som nærhet til attraksjon og åpningstider.

Når en turist med en GUIDE enhet kommer innen rekkevidde av en aktuell severdighet vil han få informasjonen presentert på skjermen. Hvis flere attraksjoner er like i nærheten vil turisten få listet opp attraksjonene slik at han selv kan velge hvilken han ønsker å se nærmere på. Hvis turisten ønsker å få informasjon om steder og attraksjoner han for øyeblikket befinner seg langt unna, støttes dette



Figur 2: GUIDE browseren med velkomstskjerm

gjennom et fritekstsøk, som vil returnere informasjon på samme måte som for den automatiske visningen. Turister kan generere omvisninger med GUIDE ved å velge hvilke severdigheter de ønsker å besøke. Systemet beregner da en rute mellom severdighetene basert på bl.a. hvilken rute som vil være mest naturskjønn å følge og hva som er mest tidsbesparende, og viser så denne ruten for turisten. Etter hvert som man følger denne ruten vil systemet lokalisere hvor man er på ruten og vise aktuell informasjon på veien. Hvis turisten ønsker å overstyre den genererte ruten kan han gjøre dette ved å selv velge rekkefølgen på severdighetene, og systemet vil generere nye ruter basert på dette. En veldig interessant funksjon som er bygget inn i dette systemet er muligheten til interaktivitet. Turister kan for eksempel reservere hotellovernattinger eller kjøpe billetter til turer og utstillinger gjennom GUIDE systemet. Siden dette systemets bruksområde strekker seg over en hel by, og lokaliseringen baserer seg på lokasjonsceller, er det designet for å kunne brukes i offline modus, dog med begrenset funksjonalitet. Dette gjøres ved at store deler av informasjonen, og da helst den statiske informasjonen, er lagret lokalt i den bærbare enheten. Når enheten er innenfor dekning vil den sjekke etter oppdateringer av informasjonen for det området den befinner seg i og laste ned denne. På denne måten kan deler av systemet fortsatt brukes selv om man ikke har dekning. Turisten kan også velge å angi hvor han befinner seg ved å velge mellom en rekke bilder som representerer stedene i rundt området hvor han sist ble lokalisert.

### Presentasjon:

Som nevnt bruker GUIDE en modifisert web browser for å vise informasjonen. Dette gjøres på en tablet PC av typen Fujitsu TeamPad 7600, med målene 213x153x15 mm. Grunnen til at valget falt på denne var at den var forholdsvis liten og lett, med tilstrekkelig datakraft på 166 MHz, og en skjerm med en oppløsning på 800x600 piksler som passet til å vise en web browser.



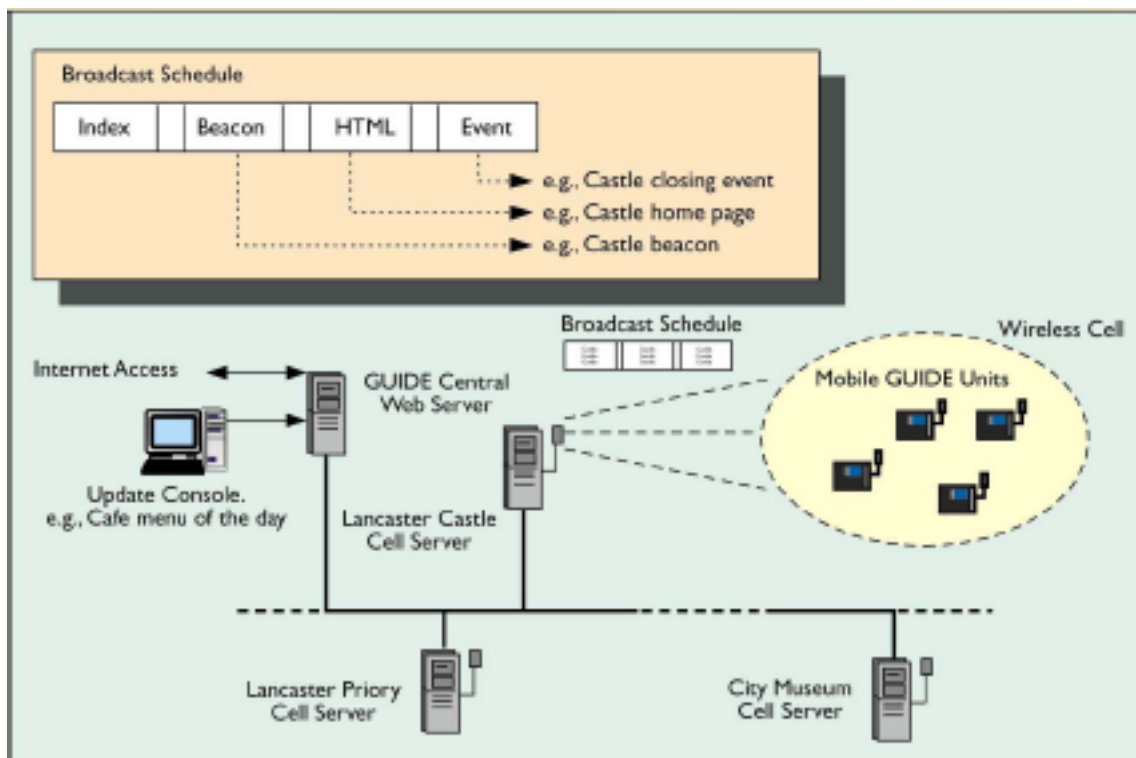
Figur 3: Tablet PC som GUIDE benytter

Bruken av den modifiserte browser metaforen ble valgt fordi den passer godt for å presentere den typen informasjon som er modellert i systemet, og fordi svært

mange er fortrolig med denne måten å interagere med informasjonssamlinger på, gjennom Internett og WWW. Det ble også forsøkt å gi systemet en vennlig personlighet slik at man skulle føle seg mer fortrolig med det og kunne interagere med det som om det var en virkelig guide. GUIDE ønsker derfor turistene hjertelig velkommen, og når de forlater et sted blir de ønsket velkommen tilbake ved en senere anledning. Vanligvis for brukergrensesnitt design anbefales ikke slik antropomorfisk oppførsel av en datamaskin. Men i dette tilfellet har besøkende turister uttrykt at de følte seg forsikret om at systemet gjorde jobben sin. Dette var da helst for ikke-tekniske besøkende og nybegynnere til systemet. For mer vante brukere vil dette valget være mindre passende. Siden dette systemet også fungerer til dels selv om det ikke har forbindelse til server er det implementert en signalindikator lik den man finner på en mobiltelefon for å signalisere til brukeren om han har dekning eller ikke. På denne måten vet alltid brukeren hvilken funksjonalitet han kan forvente av systemet til enhver tid. Han må for eksempel ha dekning for at det skal være mulig å bruke de interaktive tjenestene.

### Teknologi:

GUIDE systemet er satt sammen av en rekke webservere som hver tjener sitt område av byen. De er utstyrt med trådløse aksesspunkter som kjører IEEE 802.11 standarden på 2.4GHz radiobåndet. Disse støtter en overføringshastighet på 2 Mbps per enhet.



Figur 4: Den trådløse kommunikasjonsarkitekturen til GUIDE

Hver server holder informasjon om sin del av byen og dens attraksjoner, og får oppdateringer og dynamisk innhold via fastlinjer til den sentrale GUIDE web-



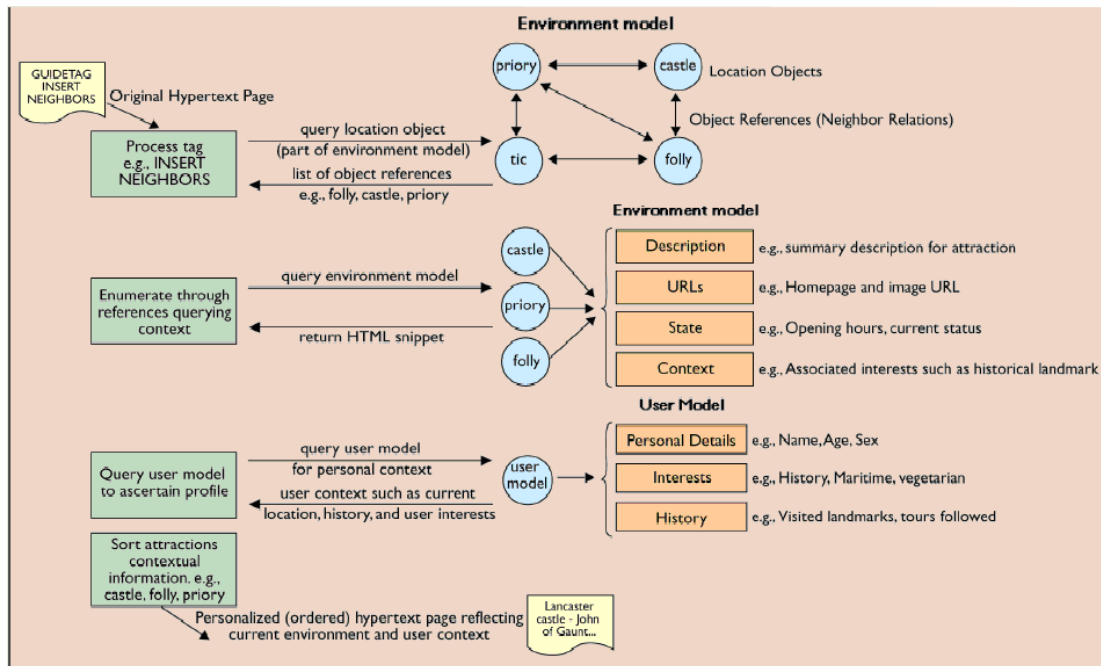
serveren. For å oppdatere innholdet brukes en standard PC for laste opp ny og oppdatert informasjon. Posisjoneringen skjer ved at de mobile enhetene mottar lokasjonsinformasjon som sendes fra strategisk posisjonerte basestasjoner. Denne fremgangsmåten ble valgt fremfor DGPS først og fremst fordi det trengte ingen ekstra maskinvare, og fordi det i områder med mye bygninger og andre hinder gjorde det vanskelig å lokalisere enhetene fordi de ikke hadde fri sikt til et tilstrekkelig antall satellitter for posisjonering.

#### **Informasjon i systemet:**

Siden utviklerne av systemet bedømte at eksisterende informasjonsmodeller var utilstrekkelige for å representere slik kontekstsensitiv informasjon, bestemte de seg for å utvikle sin egen. De hadde fire kriterier som denne informasjonsmodellen måtte kunne dekke:

- Informasjonen måtte kunne reflektere nåværende kontekst. Med andre ord, kunne reagere på hendelser i og utenfor systemet for å holde brukeren oppdatert.
- Må kunne holde geografisk informasjon og geografisk referert informasjon.
- Må kunne holde hypertekst informasjon, enten gjennom adresser for globalt bruk, eller ved lokal lagring av informasjonen.
- Må kunne støtte aktive komponenter som er i stand til å lagre tilstand, og å utføre bestemte handlinger eller imøtekomme spørringer.

Modellen ble delt inn i to distinkte deler, Environment Model og User Model. Environment Model inneholder informasjon om attraksjoner og steder. Den holder informasjon som beskrivelse, tilstand, URL til ekstern informasjon, lokasjoner i nærheten og andre relaterte lokasjoner, og kontekstinformasjon (for eksempel om dette er et historisk sted, maritimt, spisested, underholdning o.l., som kan knyttes opp i mot brukerens interesser). User Model inneholder informasjon om brukeren, personlig informasjon (navn, kjønn, alder), interesser, historie (log over besøkte steder). Dette knyttet opp i mot informasjonen i Environment Model og lokasjonen skaper grunnlaget for kontekstbevisstheten til systemet.



Figur 5: Eksempel fra informasjonsmodellen til GUIDE

### 5.1.2 TouristGuide

TouristGuide er et lokasjonsbasert turistguide system bygget av hovedsakelig COTS (hyllevare) hardware og software. Denne applikasjonen er et resultat av hovedoppgaven til Stephen Hillenbrand og Todd Simcock i Information Technology Software Engineering ved University of South Australia i 2001. Implementasjonen av systemet ble støttet av Motorola og tatt i bruk ved Mawson Lakes campus ved University of South Australia og North Terrace precinct i Adelaide sentrum.

#### Brukere:

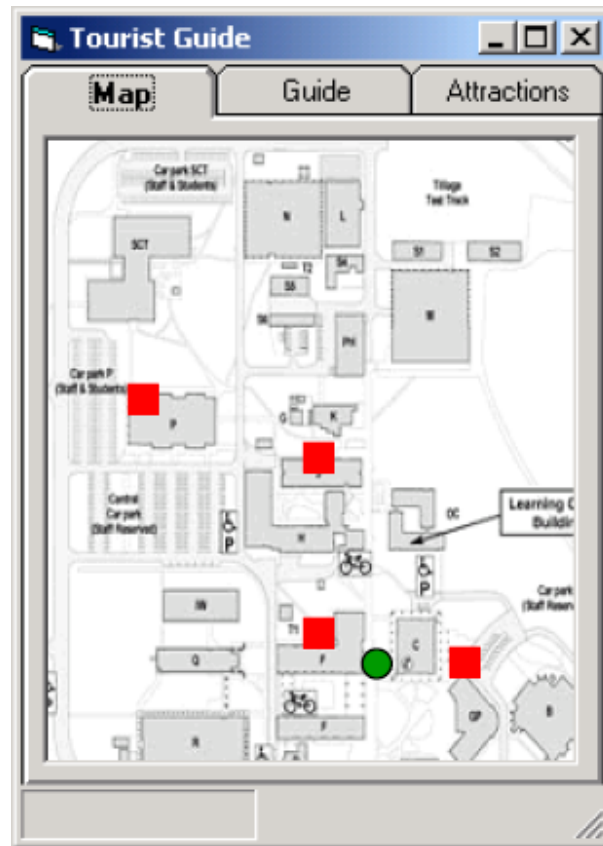
Systemet er beregnet for turister og besøkende i Adelaide sentrum og universitetet i Sør-Australia.

#### Bruksområde:

De besøkende kan finne ut hvor de befinner seg i forhold til interessante steder og attraksjoner, bli guidet rundt i området og få informasjon om severdigheter.

#### Funksjonalitet:

TouristGuide har delt funksjonaliteten til systemet inn i tre distinkte "moduser" som brukeren kan velge å benytte.



Figur 6: Modusene kan velges med arkfanene øverst

- **Map Mode:** Denne modusen består av et kart hvor brukeren kan se hvor han befinner seg i forhold til strukturer og attraksjoner i rundt seg. Systemet beregner posisjon og tegner en prikk på kartet hvor brukeren befinner seg. Man kan også velge å få vist bestemte fasiliteter på kartet, som for eksempel telefonkiosker, kafeer, barer, toaletter o.l. Denne modusen kan også kjøres uten lokalisering, og brukeren har da muligheten til å peke/klikke på "hotspots" på kartet for å få opp informasjon, slik at han slipper å måtte dra til den fysiske lokasjonen for å få tilgang til informasjonen.
- **Guide Mode:** Når brukerne benytter denne modusen vises en rute på kartet som følger et sett med interessante relaterte attraksjoner. Attraksjonene er merket på kartet, og når brukeren kommer i nærheten av en attraksjon går den automatisk over i attraction mode.
- **Attraction Mode:** I denne modusen presenteres brukeren med informasjon om attraksjonene i form lyd, bilder og tekstlige beskrivelser i små fokuserte kapitler. Mens brukeren beveger seg rundt vil denne informasjonen hele tiden oppdatere seg for å gjenspeile informasjon om stedet brukeren befinner seg.

Systemet har også lagt til rette for enkel skaping av ny ruter og attraksjoner, noe som var et av designmålene for systemet. "Målet vårt var å utvikle et rammeverk

som ville være svært enkelt å bruke og tilpasse til nye lokasjoner for folk som ikke har en skikkelig IT bakgrunn”, ifølge Hillenbrand [23].

### Presentasjon:

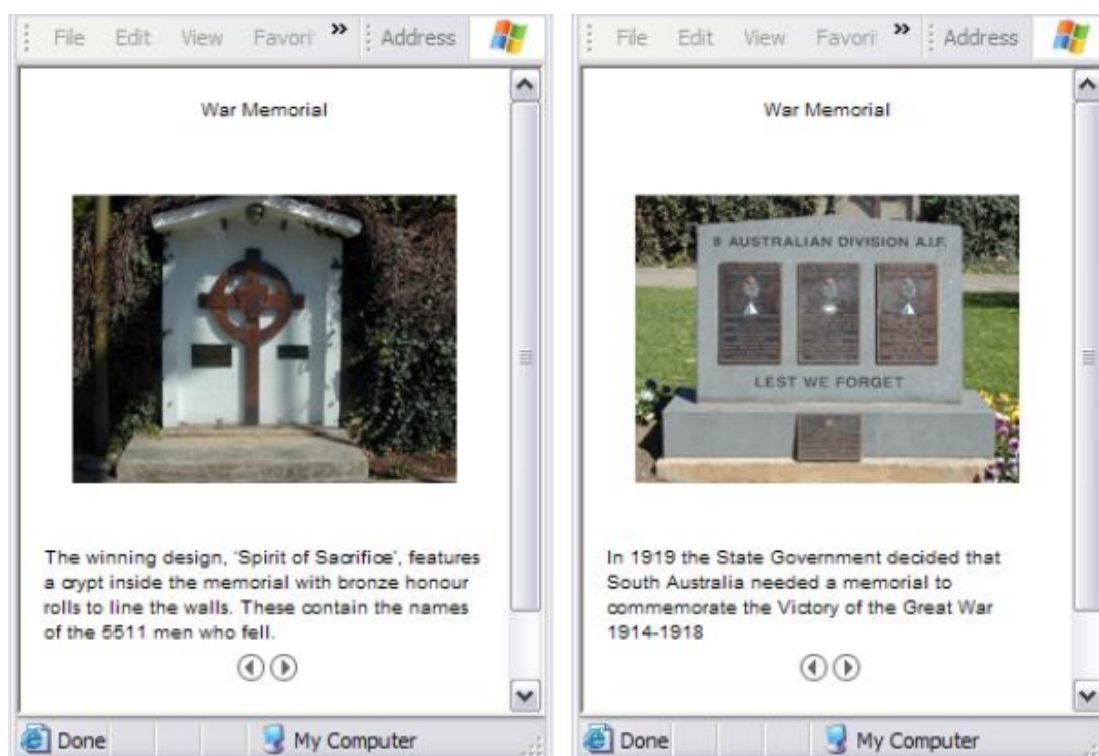
TouristGuide benytter en hendig liten PDA, Compaq Aero, til å presentere informasjon for brukeren. For å spore brukeren er PDA'en koblet til en GPS mottaker.



Figur 7: Klientenheten til TouristGuide, en PDA, koblet til en håndholdt GPS

Valget med å benytte en liten håndholdt enhet som en PDA har sine fordeler med tanke på mobilitet, men skaper også store utfordringer når det gjelder brukergrensesnitt og presentasjon av informasjon. Fordeler inkluderer bl.a. at det er enkle enheter som er lette å håndtere. De har få knapper, og det meste av interaksjonen med enheten går via touch screen (trykkfølsom skjerm), noe som overlater det meste av ansvaret for interaksjonen til designeren/utvikleren av applikasjonen. Det er også en fordel med et Windows basert brukergrensesnitt som de fleste i det minste er til dels kjent med. Størrelsen og oppløsningen på skjermen, som er på 240x280 piksler, utgjør en utfordring når man skal vise informasjon til brukeren og lage måter for brukeren å kommunisere med systemet på. Et av designkriteriene for systemet har derfor vært å kun vise et minimum av, og bare den mest essensielle, informasjonen på skjermen til enhver tid. Som nevnt er systemet delt inn i modusene map mode, guide mode og attraction mode som hver presenterer forskjellige lag med informasjon til brukeren. Map mode viser grafisk hvor brukeren er i forhold til omgivelsene. Ved å benytte en drop-down meny får brukeren

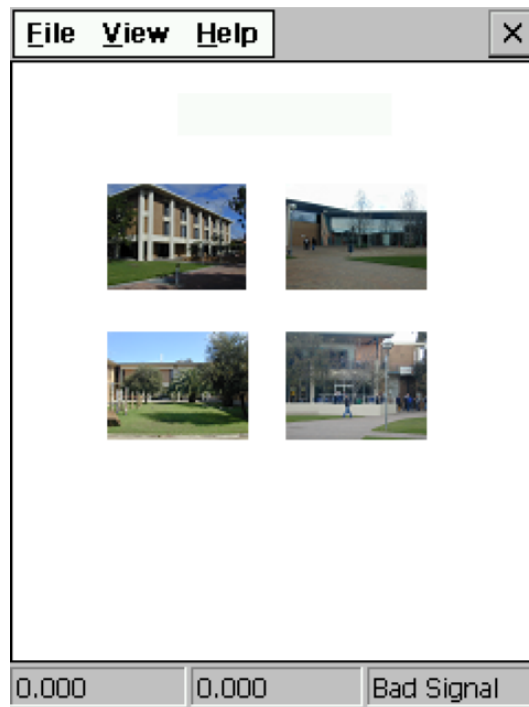
muligheten til å velge visning av forskjellig informasjon på kartet. Guide mode gjør noe av det samme, men inkorporerer visning av ruter og en automatisk overgang til informasjon om attraksjonene i nærheten. Attraction mode viser tekstlig og billedlig informasjon om attraksjonene av og til kombinert med lyd. Attraction mode viser informasjon gjennom HTML sider på skjermen. Egne maler for hvordan denne informasjonen skal presenteres er lagret i systemet. Hvis brukeren ønsker mer informasjon om attraksjonene kan han bla videre ved hjelp av piltastene nederst på skjermen. På denne måten kan brukeren selv avgjøre graden av informasjon han ønsker for hver attraksjon, og man unngår å overvelde brukeren med informasjon. Dette gir et ryddig og effektivt brukergrensesnitt.



Figur 8: HTML sider for attraksjoner i TouristGuide

### Teknologi:

TouristGuide systemet baserer lokaliseringen av brukeren på GPS teknologi. Dette fungerer tilstrekkelig siden systemet kun skal fungere i et utendørs miljø. GPS ble vurdert opp i mot DGPS da systemet ble utviklet. Tester og sammenligninger som ble utført viste at DGPS var ca. 50 % mer presist enn GPS, men siden GPS viste seg å være presist nok ble DGPS valgt bort grunnet kostnadene og vekten ved DGPS utstyret. Siden dette prosjektet fokuserte på å bruke hyllevere for å lage en enkel og anvendelig lokasjonsbasert tjeneste ble den bærbare GPS enheten Garmin 12XL benyttet sammen med Compaq Aero 2130 PDA. Dette utgjør en lett og enkel klientenhet. TouristGuide har ingen sentral server, hele systemet er inkorporert i den mobile enheten. All informasjonen ligger lagret i hver enkelt klientenhet, og blir aksessert når GPS koordinatene stemmer overens med lokasjonen. Dette vil medføre begrenset skalerbarhet og kan føre til ka-

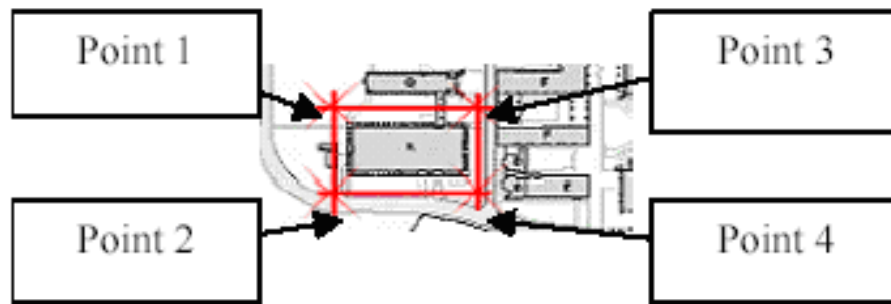


Figur 9: HTML sidevisning hvis flere attraksjoner er innen rekkevidde

pasitetsproblemer hvis systemet skulle vokse. Det legger også en begrensning på muligheten til dynamisk og up-to-date informasjon. Dette er derfor et av de videre designmålene for systemet. Økt kapasitet i den bærbare enheten med tanke på lagring og prosessering for å håndtere rikere multimedia er også et videre mål for systemet.

### Informasjon i systemet:

Informasjonsmodellen som blir benyttet i TouristGuide er enkel men effektiv. Et kart for et område blir lagret i form av en bitmap bildefil. Koordinater blir lagret for attraksjoner og "hot spots". Hot spots er informasjon som kan aksesseres ved å klikke på bygninger/objekter i map mode. Attraksjoner er severdigheter som inngår som en del av en omvisning i guide mode. Begge viser detaljert informasjon i attraction mode. Både attraksjoner og hot spots lagres med fire sett GPS (x, y) koordinater som danner et firkantet polygon. For attraksjoner blir dette gjort ved at man fysisk stiller seg på det som skal være et hjørne i polygonet og velger å lagre koordinaten for dette punktet. Dette gjør man for hvert av de fire punktene som skal omgi attraksjonen. Det blir så lagret en fil for attraksjonen med disse koordinatene. For hot spots merkes også disse punktene av rundt det aktuelle stedet, men det gjøres på skjerm ved å markere koordinatene på kartet. For hver attraksjon lagres det informasjon i form av en HTML side. Denne informasjonen inkluderer en overskrift/tittel for attraksjonen, et bilde av selve attraksjonen og en tekstlig beskrivelse på opp til femti ord. Denne sparsomme informasjonsmengden støtter opp om prinsippet om å til enhver tid kun vise den mest essensielle informasjonen. Brukeren har mulighet til å gå dypere inn i informasjonen ved å klikke seg videre med piltastene.



Figur 10: Avmerking av punkter for et lokasjonspolygon i TouristGuide

### 5.1.3 myCampus

myCampus er et semantisk web miljø for kontekstbevisste mobile tjenester som er under utvikling og validering ved Carnegie Mellon University og dets campus. Dette miljøet og arkitekturen det er bygget rundt inneholder en rekke agenter som tilbyr forskjellige tjenester basert på en brukers kontekst, også inkludert lokasjon.

#### **Brukere:**

Brukerne av dette systemet er studenter, ansatte og ellers alle som tilbringer mye av hverdagen på Carnegie Mellon universitetets campus. Siden et slik campus i seg selv er et slags mikrokosmos hvor medlemmene i samfunnet gjør alt fra å sosialisere, drive sport, shoppe, delta på arrangementer, studere, jobbe etc., gjør det at dette designet lett lar seg skalere til andre områder eller utvides utover campus området og til flere og andre typer brukere.

#### **Bruksområde:**

myCampus arkitekturen er laget for å støtte brukernes opphold og utførelse av oppgaver på campus. Dette innebærer ting som planlegging, kommunikasjon og lokalisering av personer, fasiliteter og utstyr med mer.

#### **Funksjonalitet:**

Funksjonaliteten til dette systemet/arkitekturen baserer seg på intelligente ”agenter” innenfor forskjellige bruksområder som igjen tilbyr en rekke relevante tjenester. Agenter kan opprettes og legges til systemet etter hvert som man finner behov og applikasjonsområder for dem. Informasjon om brukerne ligger lagret i det de har valgt å kalle en eWallet, og brukes for å dedusere konteksten til brukeren. Denne brukes også til kontroll av rettigheter i systemet, for eksempel om en bruker vil tillate at en annen bruker interagerer med hans eWallet på forskjellige måter, som det å sjekke hvor han befinner seg eller om han har planer for kvelden. Dette kan hver enkelt bruker bestemme i forhold til andre brukergrupper eller bestemte brukere. En agent som er implementert i systemet er en ”restaurant veileder”. Denne veilederen gir brukerne forslag til hvor de skal spise basert på kontekst som tid på dagen, foretrekt mat, tid tilgjengelig før nye forelesninger, møter e.l., lokasjon på campus-området og været. Hvis brukeren liker italiensk mat vil veilederen foreslå en italiensk restaurant, hvis det regner vil den foreslå et spisested hvor man slipper å gå utendørs fra lokasjonen man befinner seg, og hvis

man bare har 20 minutter tilgjengelig vil den foreslå et fast-food sted så nærme som mulig.

En annen agent under utvikling og testing er et kontekstsensitivt meldingsfilter. Brukere kan få meldinger om aktuelle arrangementer rundt om på campus som er matchet mot en statistisk brukerprofil. I tillegg kan filteret for eksempel velge å forsinke leveringen av beskjeden til en student er ferdig med forelesningen han sitter i, basert på kalenderen hans og hans preferanse om å ikke bli forstyrret av visse typer meldinger når han er opptatt. Meldingen blir da levert så snart forelesningen er over. En kontekstsensitiv påminner kan minne brukeren på gjøremål han har for dagen, som å handle melk når han er i nærheten av en butikk, eller hente en oppgave når han passerer riktig skolebygning på campus. En planlegger agent lar en bruker sjekke om en annen bruker er ledig for å for eksempel dra på kino klokken 20.00. Hvis brukeren har tillatelse til å sjekke dette hos den andre brukeren kan planleggeren automatisk legge dette inn i begges kalender som en avtale/møte.

Lokalisering av personer er en funksjon som enkelt kan gjøres i dette systemet. Gjennom aksesskontroll for denne tjenesten kan man også bestemme hvem som skal få lov til å se hvor man befinner seg, og hvor presis denne sporingen skal være, for eksempel på rom nivå, bygning eller by nivå.

En plakat agent tilbyr muligheten for virtuelle plakater på campus. Siden plakatoppheving både er tidkrevende og ikke spesielt pent å se over alt synes mange at dette er en nyttig agent. Meldinger med plakatbeskjeder kan sprette opp hos brukere når de beveger seg rundt på campus. Plakatene blir virtuelt hengt opp på bestemte steder, og kan planlegges ut i fra typiske ruter som studenter fra forskjellige linjer pleier å gå til daglig.

Det finnes også en rekke andre agenter under utvikling og testing, men de overnevnte er representative for de mest populære og funksjonsrike agentene for øyeblikket.

### **Presentasjon:**

Siden dette er en web basert tjeneste kan den benyttes via både laptop og PDA med trådløs nettverstkobling. Grensesnittet varierer fra agent til agent, men det er tatt forbehold om den begrensede plassen som er på en PDA skjerm.

### **Teknologi:**

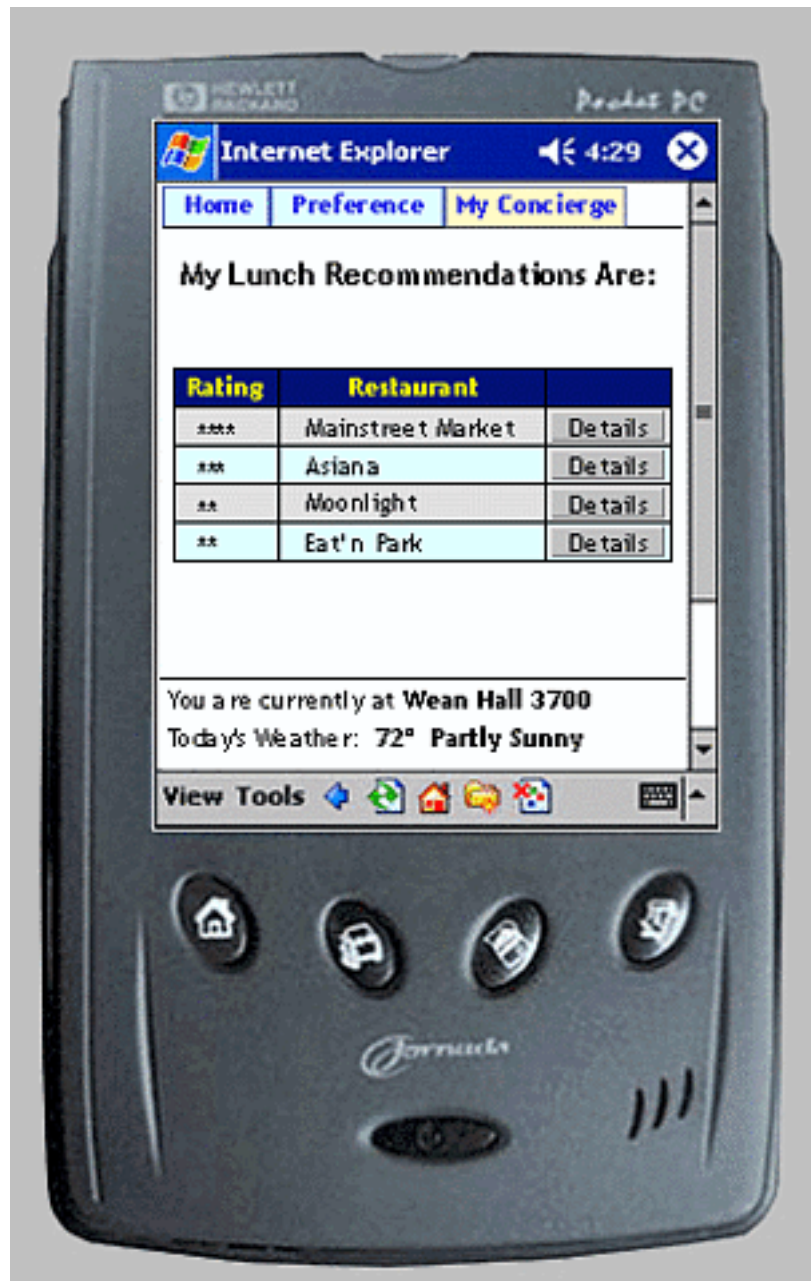
Som nevnt kan systemet benyttes fra mer eller mindre hvilken som helst webtilrettelagt enhet med trådløs nettverstkobling. Systemet kjører over IEEE 802.11 b/g nett, og lokaliseringen skjer ved hjelp av PanGo Networks lokaliseringplattform som kjører på det eksisterende WiFi-nettverket ved Carnegie Mellon University.

Siden myCampus er et kontekstbevisst semantisk web miljø bruker den ontologier gjennom OWL Semantic Web reasoning engine for å håndtere spørringer. Denne bruker da informasjonen som er semantisk beskrevet i eWallet til å svare på spørringen og deretter utfører agenten tjenesten.

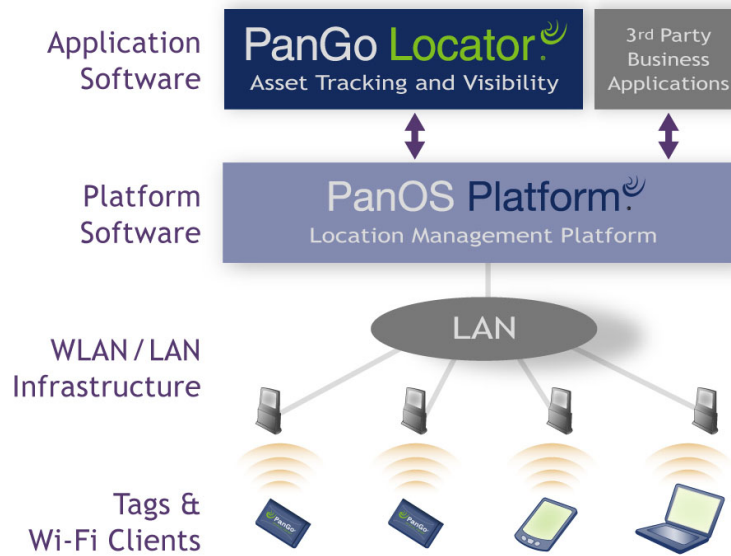
### **Informasjon i systemet:**

Informasjonen i myCampus rammeverket er som nevnt tidligere bygget rundt det



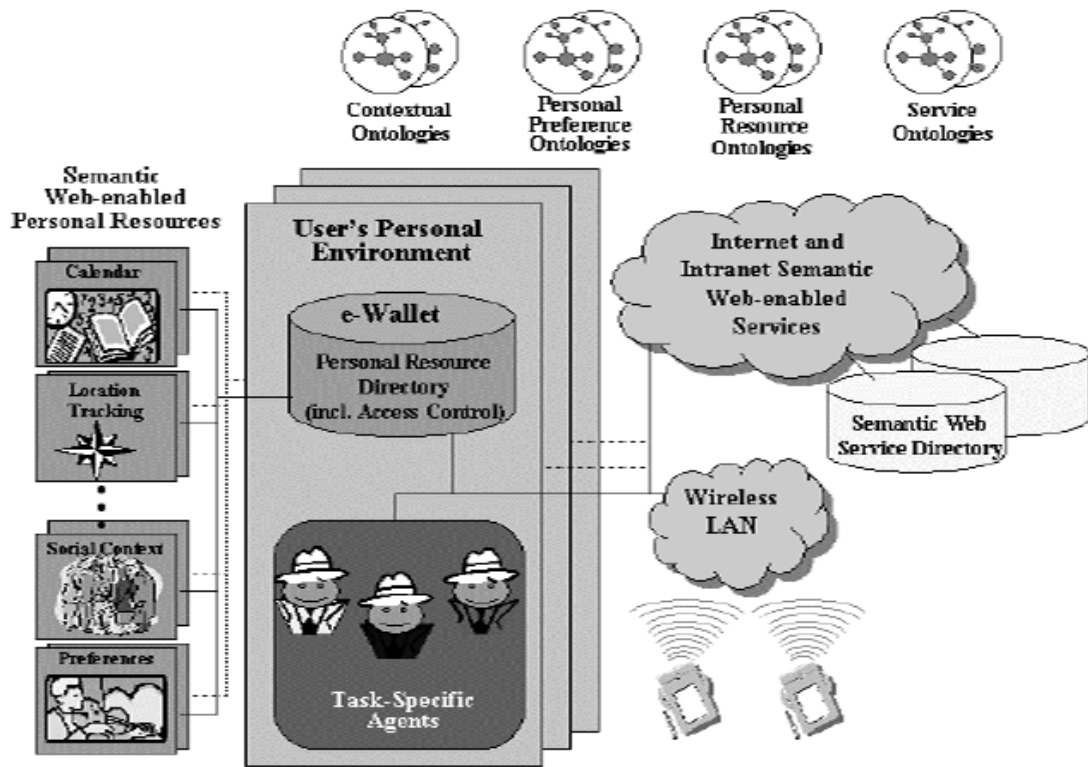


Figur 11: myCampus restaurantveilederen på en PDA

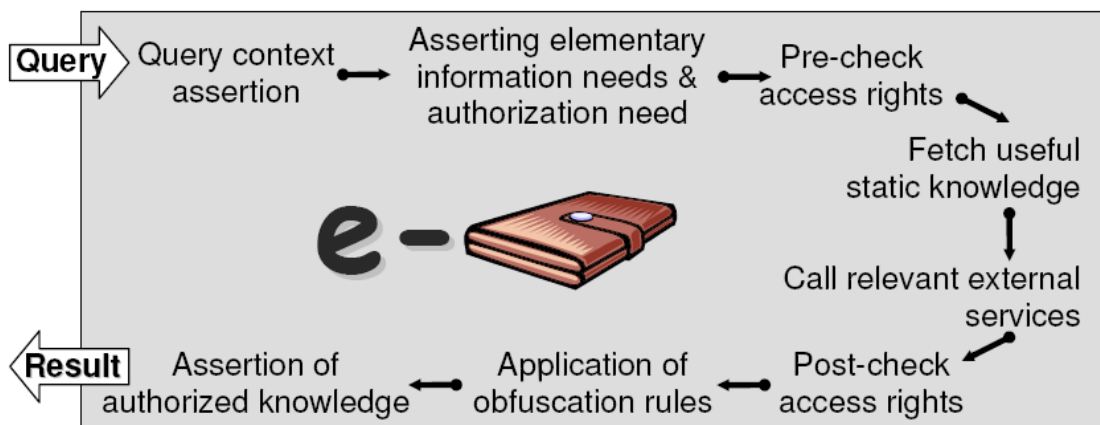


Figur 12: PanGO Networks' lokasjonsinfrastruktur

de har valgt å kalle en eWallet. Denne samler og holder kontekstinformasjon om brukeren, og benyttes av agentene til å utføre kompliserte spørringer. eWallet forsøker å samle all kontekstinformasjonen om en bruker som vanligvis finnes spredt rundt på forskjellige enheter og systemer ellers i hverdagen. Blant annet gjelder dette kalendere og kontaktlister. I tillegg brukes en avansert brukerprofil, som inneholder alt fra personalia til personlige preferanser, og lokalisering av brukeren til å avgjøre riktig kontekst for de forskjellige agentene og tjenestene de tilbyr.



Figur 13: myCampus arkitekturen med agenter og eWallet



Figur 14: Slik ser en spørring mot eWallet ser ut

#### 5.1.4 Cyberguide

CyberGuide[1] er et av de første prosjektene (1996-1997) som gikk inn for å lage en lokasjonsbasert tjeneste for mobile enheter. Prosjektet hadde som mål å utvikle en enkel og billig arkitektur for prototyping av lokasjonsbaserte tjenester, og ble utført ved Graphics, Visualization and Usability (GVU) Centre, Georgia Institute of Technology, også kjent som Georgia Tech, av studenter og ansatte.

##### **Brukere:**

Systemet var opprinnelig tiltenkt brukt av besøkende ved GVU, men ble etter hvert også utvidet til å gjelde besøkende i sentrum av Atlanta, Georgia, som var på utkikk etter forfriskninger.

##### **Bruksområde:**

Hovedsakelig skulle systemet benyttes som en erstatning for kart og en informasjonspakke til besøkende ved GVU under deres månedlige "åpent hus" omvisning, hvor de besøkende kunne vandre rundt og se på utstillinger, eksperimenter og demonstrasjoner ved senteret. Etter hvert ble det også utviklet en utendørs versjon av systemet som hjalp besøkende i sentrum av Atlanta, Georgia med å finne barer basert på forskjellige kriterier.

##### **Funksjonalitet:**

CyberGuides funksjonalitet er relativt enkel, men er den som de fleste systemer i dag fortsatt benytter som kjernefunksjoner. Den tilbyr et enkelt kart over området brukeren befinner seg i, og merker av punktet hvor brukeren befinner seg på kartet. Severdigheter merkes også av, i dette tilfellet med en stjerne, slik at den besøkende vet hvor han kan finne dem. Ved å klikke på severdigheten vil brukeren få opp et kort tekstlig sammendrag av hva som befinner seg der. Brukeren kan også gå inn i denne informasjonsmodusen og søke etter informasjon basert på navn eller bla igjennom ved hjelp av kategorier. Systemet tar vare på historiekontekst for brukeren ved å merke av besøkte severdigheter med en hake på kartet.

Systemet har implementert en meldingstjeneste som kort og godt lar brukere få adgang til å sende e-post. Denne ble benyttet til å lage bl.a. et spørreskjema angående tjenestene i systemet som ble sendt på e-post til utviklerne.

Da systemet ble utvidet til utendørs bruk ble det også lagt til en bar guide for sentrum av Atlanta, hvor brukerne kunne finne barer på kartet og få opp informasjon om disse. I tillegg var det lagt til mulighet for brukerne å notere sine egne synspunkter på barene, både objektive synspunkter så som mulighet for parkering og gjennomsnittlig pris, og subjektive som miljø/atmosfære og andre kommentarer. Denne funksjonen minner litt om kjernefunksjonen i systemet Stick-E-Notes [26].

Siden CyberGuide ikke bare er et turist guidesystem men et rammeverk for rask utvikling/prototyping av lokasjonsbaserte tjenestesystemer[17] med turistnæringen som utgangspunkt, er ikke all funksjonaliteten implementert, men foreslått og beskrevet. Bl.a. er en ønsket del av meldingstjenesten beskrevet som muligheten til å kringkaste viktige beskjeder til bestemte typer brukere, f.eks. meldingen "Bussen går fra stoppestedet om 15 minutter!" til en bestemt gruppe besøkende turister. En annen funksjonalitet beskrevet er en "historiker"-tjeneste som tar vare på in-

formasjon om hvor brukeren har vært (dette er delvis implementert i form av avmerking av besøkte severdigheter), for så å gi en oppsummering over f.eks. dagens observasjoner, evt. kunne lage automatiske minnebøker fra ferieturer o.l. basert på denne historiske informasjonen. I tillegg nevnes et ønske om å kunne lage utvidet funksjonalitet i forhold til annen kontekstinformasjon som tid på døgnet og spesielle hendelser (events) sammen med historisk informasjon om hvor brukeren har vært.

Under ”videre arbeid” i [1] for CyberGuide beskriver utviklerne ønsker om større mulighet for brukerne til å påvirke informasjonsbasen, økt kontekstbevissthet for systemet, bedre utnytting av den etterhvert så velkjente informasjonsvisningsmetaforen ”web browsing” og større mulighet for bruk av multimedia informasjon.

### Presentasjon:

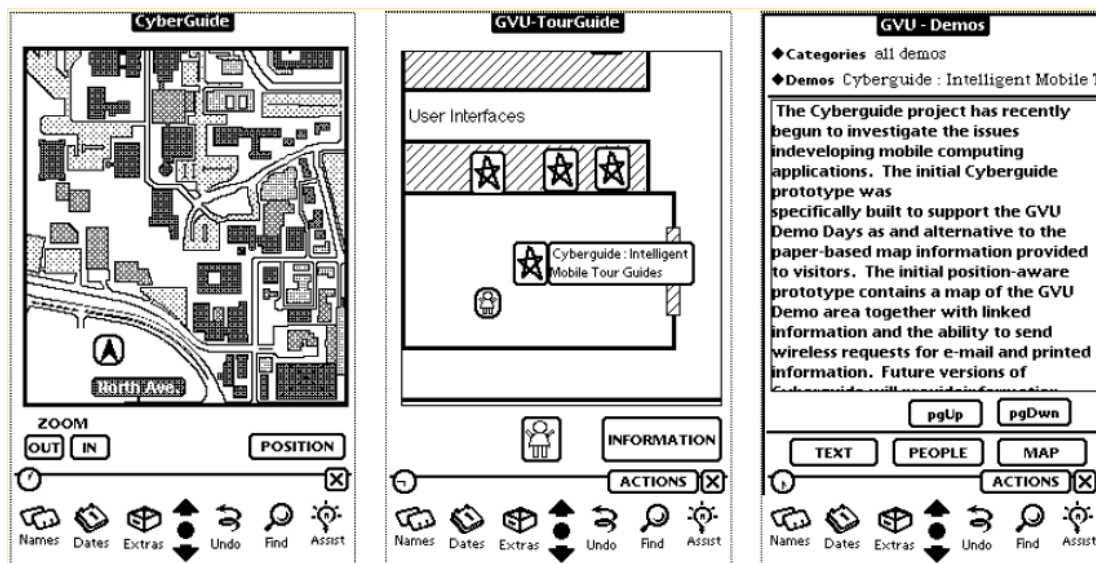
CyberGuide er designet og laget for mobile/bærbare enheter, og da helst PDA/pen-baserte enheter. Det ble tatt utgangspunkt i en Apple MessagePad 100 med Newton 1.3 og pen-baserte PDA enheter med Windows.



Figur 15: CyberGuide klientenhet Apple MessagePad med GPS-enhet

Brukergrensesnittet har i likhet med TouristGuide [30] en kartdel og en informasjonsdel, hvor man på kartet kan se sin egen posisjon i forhold til severdigheter og få informasjon om severdighetene i informasjonsdelen. For å finne fram i informasjonen har CyberGuide grensesnittet et kjekt lite kontrollpanel nederst på skjermen.

Bar guiden for sentrum av Atlanta er strukturert på omtrent samme måte. Den viser posisjonen av aktuelle barer i forhold til brukeren, samt aktuell informasjon om barene i et eget informasjonsvindu hvor man også finner muligheten til å påvirke denne informasjonen selv gjennom å bl.a. rangere baren, angi inntrykk av



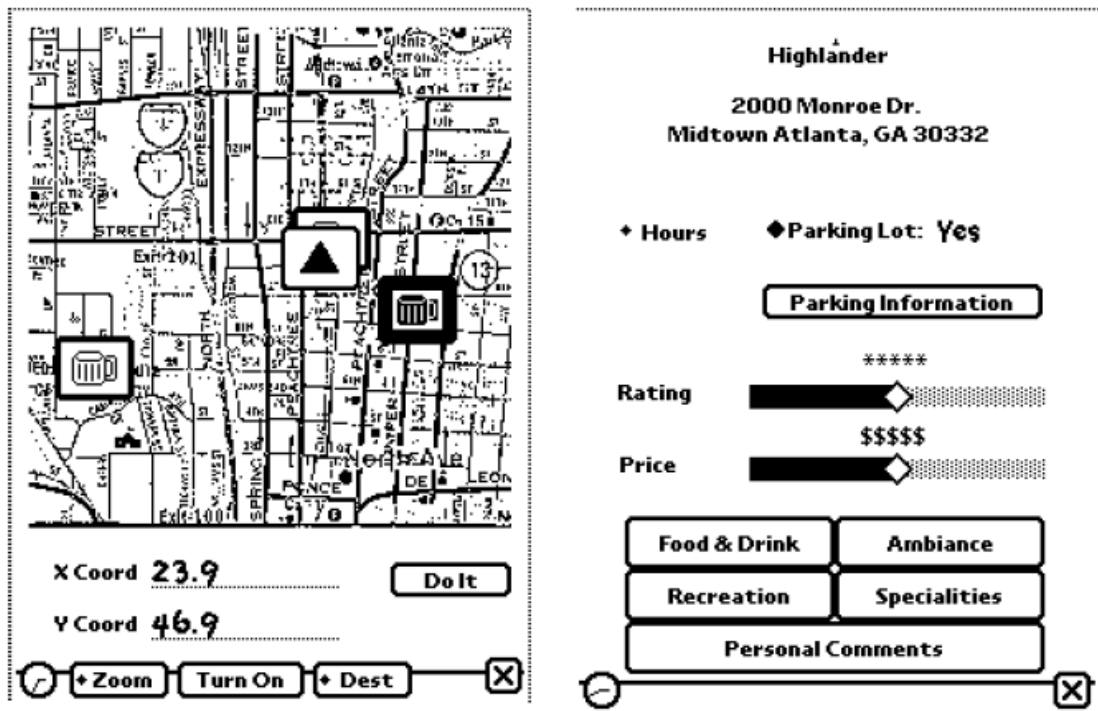
Figur 16: CyberGuide prototype grensesnitt på MessagePad for kart og informasjon. Utendørs versjon til venstre, innendørs i midten, informasjon til høyre.

prisgjennomsnitt, kommentere med mer.

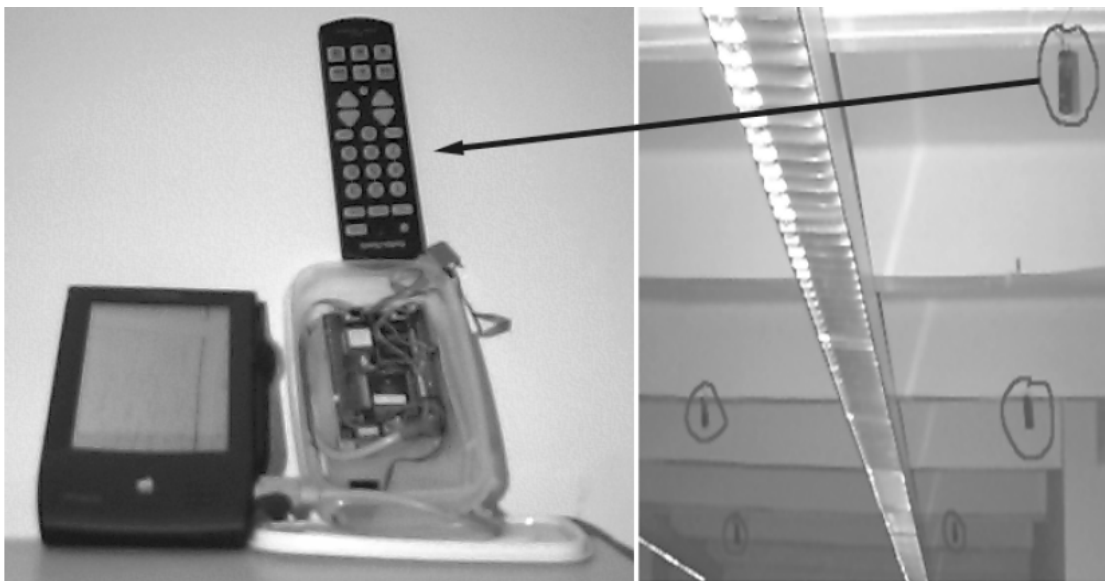
### Teknologi:

Som nevnt benytter CyberGuide PDA'er som klientenheter. For posisjonering benyttes standard GPS koblet til klientenheten for utendørs bruk. For innendørs bruk ble RF-baserte metoder vurdert, men hyllevareløsninger for dette ble ansett som for dyrt ved det tidspunktet. Valget landet derfor på IR-basert sektor/celle posisjonering. Dette ble løst på en ad-hoc måte, ved å benytte vanlige TV fjernkontroller som aktive signalnoder, og en spesiell IR mottaker med en mikrokontroller som kunne kobles til en serieport på PDA'en. Når en bruker da beveger seg mellom IR-nodene vil mottakeren fange opp signalet som inneholder et bestemt mønster og bestemme hvor man befinner seg ut ifra hvilken node som blir identifisert. Dette fungerer greit i relativ liten skala, men blir også fort dyrt da mikrokontrolleren koster omtrent like mye som PDA'en.

Det blir påpekt under "videre arbeid" at CyberGuide benytter forskjellige teknologier for innendørs og utendørs posisjonering, og at å benytte kun én til begge bruksområdene ville vært å foretrekke (allestedsnærværende posisjoneringssystem). CyberGuide påpeker også at det i de fleste tilfeller er ønskelig med en nær kobling mellom posisjonering og kommunikasjon for et slikt system. Når det gjelder CyberGuide systemet er disse separate, noe som bl.a. fører til en økt mengde hardware som kreves av klientenheten, og måten det er løst på fører til at kun klientenheten vet hvor den er hen. For å kommunisere med andre tillater CyberGuide å sende e-post. Dette krevde dog at man koblet seg til en Internet Appletalk forbindelse, og var mulig gjennom nok en ad-hoc løsning hvor det ble laget en applikasjonslevel protokoll som ble lagt over en public domain implementasjon av Appletalk protokollen for Solaris. På denne måten kunne de åpne en datastrøm mellom MessagePad'en og en UNIX plattform, og fikk deres egen gateway til å



Figur 17: Grensesnitt for bar guiden til CyberGuide



Figur 18: CyberGuide IR posisjoneringssystem

gjøre om pakkene til TCP/IP pakker for normal Internett trafikk. Siden dette var ganske tungvindt og hadde ganske begrenset båndbredde ble dette listet under "videre arbeid", hvor de ønsker seg en bedre og mer pålitelig forbindelse med TCP/IP. All informasjonen som tilbys i systemet er derfor også lagret lokalt på klientenheten. Det er ønskelig å ha dette sentralt for å kunne endre og oppdatere informasjon kontinuerlig uten å være avhengig av å gjøre dette for hver enkelt klient. Utviklerne har også satt interoperabiliteten mellom plattformer høyt på agendaen, og legger til rette for at systemet skal kunne brukes uavhengig av plattform. De utviklet derfor to klientapplikasjoner basert på den samme arkitekturen, en for Apple MessagePad og en for Windows baserte klienter.

**Informasjon i systemet:**

Informasjonen som benyttes i CyberGuide er hovedsakelig den samme som man ville finne i en brosjyre som beskriver demonstrasjoner ved en messe eller utstilling. Lokasjonen for severdighetene er lagret og en generell beskrivelse for hver enkelt er lagret sammen med et navn og organisert under en kategori. Når det gjelder bar guiden finnes som nevnt tidligere en egen oppføring for brukerens personlige informasjon om et sted, vurdering, kommentarer, prisrangering og atmosfære. Annen informasjon om barene inkluderer mat og drikke, åpningstider, spesialiteter og rekreasjonsmuligheter.



### 5.1.5 comMotion

comMotion er et lokasjonsbevisst system som kobler personlig informasjon til lokasjoner i en brukers hverdag, som for eksempel handlelisten når du er i den lokale butikken eller påminnelse om et viktig møte når du er på jobben.

Systemet er utviklet av Ph.D. studenten Natalia Marmassee og leder for "Speech Interface Group" Chris Schmandt ved MIT Media Laboratory i Cambridge, USA.

#### **Brukere:**

Brukerne av dette systemet er tiltenkt å være hvem som helst som er mye på farten og som potensielt har mye å holde orden på i hverdagen.

#### **Bruksområde:**

comMotion er designet for å kunne brukes hvor som helst, siden informasjonen blir gitt av brukeren selv heller enn en sentral organisasjon. Dog er systemet begrenset til utendørs bruk når det gjelder lokalisering siden det benytter GPS. Systemet er ment å kunne brukes over store strekninger heller enn meter-for-meter lokalisering. Systemet er tiltenkt som et hjelpemiddel i hverdagen hvor informasjonsmengden i mange tilfeller er overveldende.

#### **Funksjonalitet:**

Funksjonaliteten til dette systemet består hovedsaklig av to-do lister (huskelister/forefallslister), påminnere, abonnementsbaserte informasjonstjenester og 2D kart, i tillegg til å identifisere personlige viktige lokasjoner for brukeren.

Når man først tar i bruk comMotion begynner det å lære seg hvilke lokasjoner man besøker ofte, og ber så brukeren navngi disse lokasjonene som for eksempel "hjem" eller "jobben". Disse virtuelle lokasjonene kan så brukes sammen med de øvrige funksjonene i systemet.

To-do lister kan opprettes av brukeren og knyttes til bestemte lokasjoner. Disse listene inneholder elementer som kan strykes ut etterhvert som man har utført oppgavene. Elementene kan være i form av tekst eller lydopptak. Når brukeren ankommer et sted hvor det er knyttet en liste som inneholder elementer vil han høre et lydsignal som minner ham på at han har ugjorte oppgaver i listen.

Andre brukere kan også sende påminnere til hverandre gjennom comMotion. Disse påminnerne minner om elektroniske post-it lapper, og sendes til en persons virtuelle lokasjoner. Brukeren vil da få opp meldingen når han ankommer lokasjonen som ble spesifisert av avsenderen. Disse påminnerne kan begrenses til både de virtuelle lokasjonene og til bestemte datoer og tidsrom.

Brukere av comMotion kan gjøre forespørsler etter lokasjonen til andre brukeren. Svaret vil være på formen "Bjørn befinner seg på jobben" eller "Bjørn er på vei hjem". For å ivareta personsikkerheten lar systemet brukerne selv bestemme hvem som kan forespørre hvor de befinner seg, og også ved hvilke lokasjoner de skal kunne lokaliseres. Hvis en forespørsel ikke kommer igjennom dette filteret vil man få en melding på formen "Bjørn er incognito!"

comMotion har også en funksjon hvor brukeren kan abonnere på informasjonstjenester som for eksempel nyheter, værmeldinger og oppdaterte kinolister. Abonnementene gjelder per lokasjon og man kan spesifisere forskjellige tidsplaner for tjenestene. For eksempel kan man be om å få levert en liste over hvilke filmer som

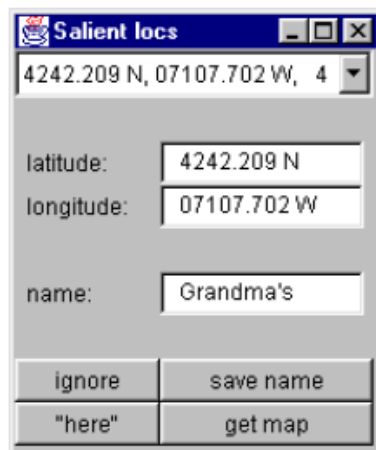
går på den lokale kinoen hver fredag når man går fra jobb, eller få værmeldingen for der man er hver morgen klokken 08.00.

I likhet med de fleste andre lokasjonsbevisste systemer tilbyr også comMotion en kartfunksjon hvor brukeren kan se hvor han befinner seg i forhold til omgivelsene. Her har man mulighet til å se nærliggende steder/bygninger som banker, kinoer og butikker.

### Presentasjon:

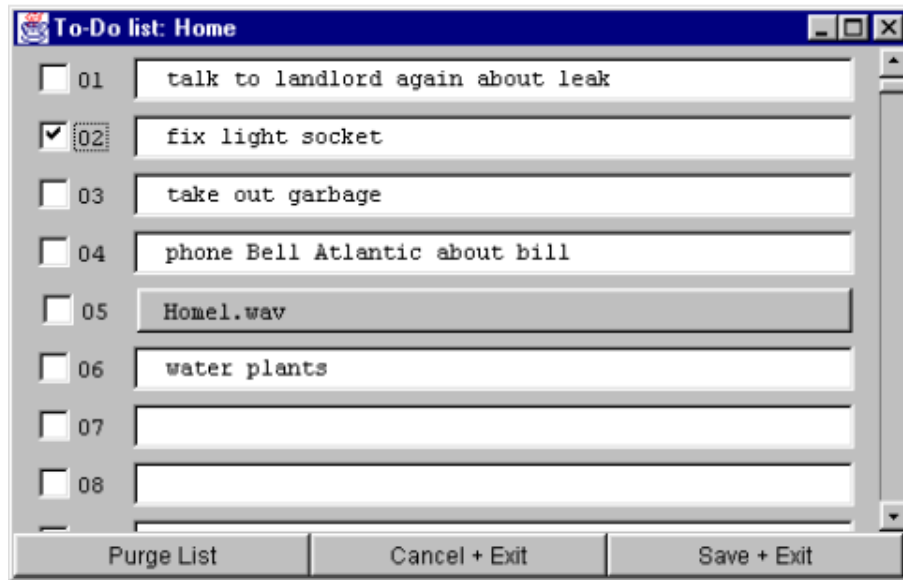
Under utvikling og testing ble comMotion klienten benyttet på en bærbar PC med en tilkoblet GPS mottaker og et modem. For lyd ble det brukt et head-set med mikrofon. Brukergransesnittet ble utviklet for å kunne bruke både visuelle og audielle kommunikasjonsmetoder. Siden systemet er tiltenkt brukt mens man heller tiden er på farten, er det ikke alltid at man har hender eller øyne ledig. Derfor finnes det i tillegg til det vanlige grafiske grensesnittet et audiogrensesnitt som benytter tekst-til-tale for å gi tilbakemeldinger og talegjenkjenning for å interagere med systemet.

Når en ny viktig lokasjon blir identifisert av systemet vil brukeren få opp en beskjed som ber om et navn på lokasjonen. Her kan brukeren navngi lokasjonen hvis dette er en lokasjon som er viktig, eller velge å ignorere lokasjonen hvis han mener systemet har gjort en feiltolkning.

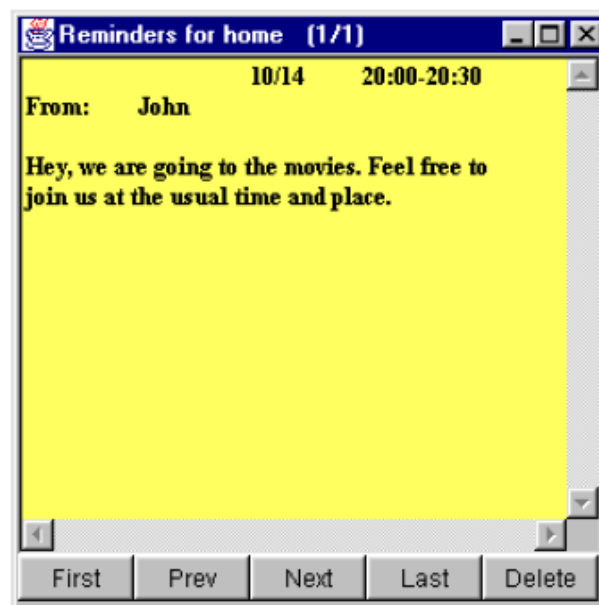


Figur 19: Identifisering av viktig lokasjon i comMotion

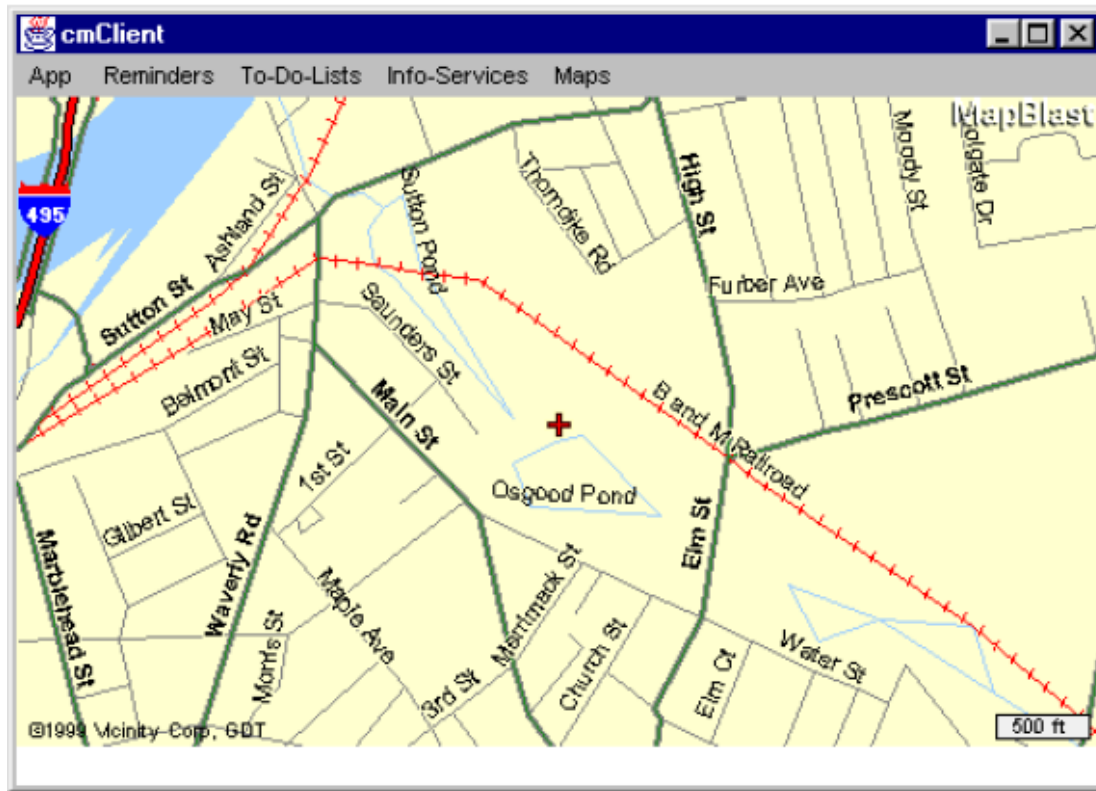
Disse lokasjonene får, som nevnt tidligere, knyttet to-do lister til seg. Disse listene er i form av enkle tekstbokser, og kan krysses av ved hjelp av checkbokser. Påminnerne som kan sendes mellom brukerne av systemet minner mye om klassiske post-it lapper i både størrelse, form og farge.



Figur 20: comMotion GUI for to-do liste



Figur 21: Melding sendt til en bruker ved en bestemt lokasjon



Figur 22: Kart over området hvor brukeren befinner seg

Kartet opptrer som hovedvinduet i klienten med menyvalg for de øvrige funksjonene. Kartet viser hvor brukeren befinner seg i omgivelsene, og kan også vise lokasjonen til en rekke steder/lokasjoner av interesse, som banker, kinoer og butikker. Man kan få informasjon om adresse og telefonnummer på de aktuelle stedene.

comMotion benytter også et avansert talestyringssystem som lar brukeren benytte systemet når både hender og øyne er opptatt med andre ting som for eksempel å kjøre bil, sykle eller man har masse å bære på. Uten å gå for mye i detaljer lar dette brukeren sjekke meldinger og to-do lister gjennom talekommandoer og får disse lest opp med tekst-til-tale funksjoner. Man kan også krysse av for gjennomførte oppgaver og legge inn talemeldinger. Talestyringssystemet er begrenset til de viktigste funksjonene.

### Teknologi:

Bærbare PC'er ble brukt som klientenheter under utvikling og testing av comMotion, med tilkoblet GPS sensor, modem og head-set med mikrofon. Utviklerne ønsker å tilpasse dette til en håndholdt enhet som en PDA eller kanskje også en mobiltelefon i fremtidige versjoner for å gjøre det mer mobilt. Trådløst head-set og mikrofon er også på planen.

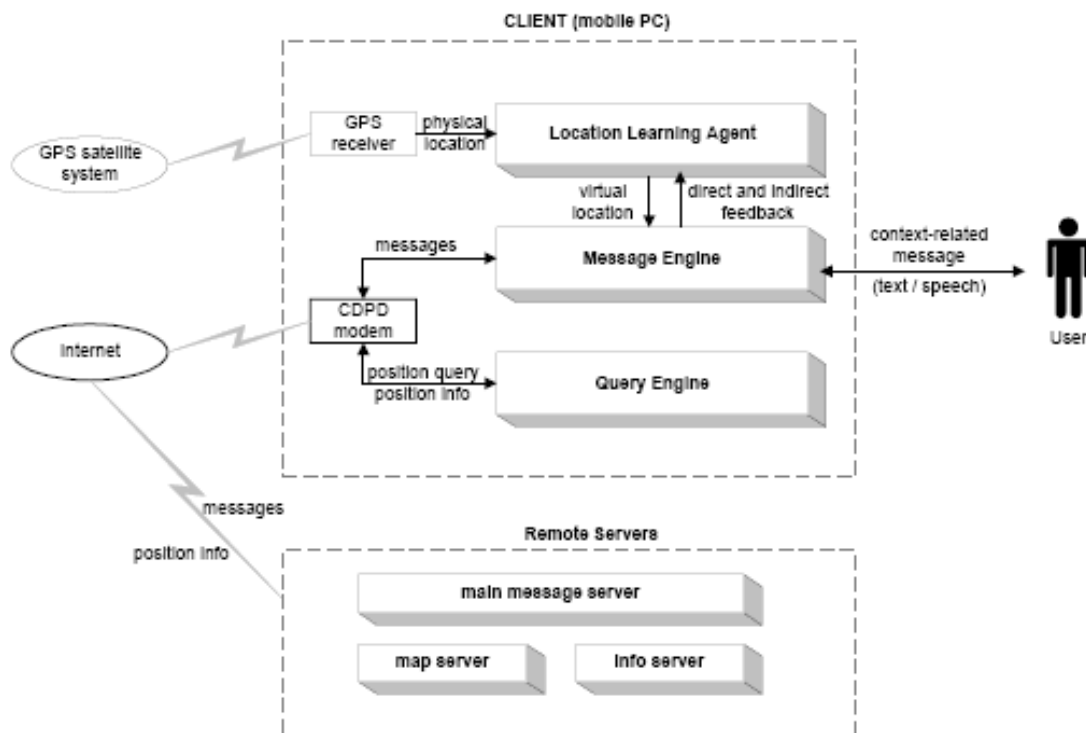
For posisjonering benyttes ganske enkelt GPS med avlesing av lengdegrad og breddegrad ved hjelp av NMEA-0183 protokollen fra GPS mottakeren via serieport til den bærbare PC'en. Høydeinformasjonen blir oversett siden den er regnet som urelevant for dette systemet.

På serversiden benyttes én enkelt servermaskin med programvare for å håndtere beskjeder og to-do lister, kartene som er tilgjengelige og informasjonen til abonnentstjenestene.

### Informasjon i systemet:

Informasjonen som håndteres i dette systemet er hovedsaklig informasjon som er laget av brukerne selv. Viktige personlige lokasjoner lagres for hver bruker, samt to-do lister som knyttes til disse lokasjonene. Påminnelser/meldinger fra andre brukere går også gjennom systemet, men slettes fra serveren når de er overlevert til klienten. Innholdet i to-do listene blir også slettet når de er gjennomført. Oppføringene i listen kan være enten tekstlig eller lydopptak.

Kartene håndteres på serveren og vises for klienten. Informasjon om steder og bygninger som banker, kinoer og butikker blir hentet fra en ekstern web-database, MapBlast [18] (nå Windows Live Local), og sendt til klienten. Informasjonen for abonnentstjenestene blir også hentet fra eksterne web-kilder, formatert og levert til klienten etter behov.



Figur 23: Systemarkitektur for comMotion

### 5.1.6 Andre liknende systemer

#### Stick-E-Notes[26]

Stick-E-Notes er et kontekstbevisst system hvor bl.a. lokasjon er en del av konteksten. Systemet bruker Post-It notes (gule lapper) som metafor for hvordan systemet fungerer. Brukeren kan sette slike elektroniske ”lapper” på steder og situasjoner som vil bli vist for brukeren når konteksten er riktig, for eksempel når man er på et bestemt sted. Hvis man har elektroniske lapper koblet sammen med attraksjoner i Disney World vil disse lappene bli vist for brukeren når man kommer innen en gitt rekkevidde av de aktuelle attraksjonene. Systemet har også sin egen kontekstmodell, Stick-e Note Architecture, som viser hvordan de bygger opp objekter som representerer kontekstrikte situasjoner. For eksempel kan følgende situasjon være et kontekstobjekt: Lokasjonen er kontoret, temperaturen er over 25 grader og tidspunktet på ettermiddagen en lørdag (fredag). Brukeren får da opp en melding om at det er på tide å dra på stranden. Da inngår lokasjon, temperatur og tidspunkt som deler av kontekstobjektet som meldinger er festet til.

#### RADAR[4]

RADAR er et innendørs lokasjons -og sporingssystem utviklet ved Microsoft Research basert på radiosignaler og er for innendørs bruk. Teknikken beskrevet kommer inn under begrepet WiFi Fingerprinting. Basert på denne sporingen tilbyr RADAR lokasjonsbaserte tjenester som blant annet kan skrive ut dokumenter på nærmeste printer i forhold til brukeren, og kan brukes til å navigere gjennom bygninger og rom.

## 5.2 Arkitekturer

Denne delen ser på systemarkitekturer for utvikling av lokasjonsbaserte systemer. Arkitekturen benyttet i systemet CyberGuide blir gjennomgått samt en arkitektur kalt PoLoS.

### 5.2.1 CyberGuide

I sin utvikling av CyberGuide har utviklerne fokusert på et konseptuelt design [1], eller arkitektur, som fanger opp essensen av hva systemet ønsker å kunne gjøre. Dette ble gjort fordi det ble enklere å se systemet både som enkeltkomponenter og som en helhet, og fordi de ønsket å enkelt kunne eksperimentere med forskjellige prototyper og versjoner av CyberGuide. Dette resulterte i en komponentbasert arkitektur hvor de fant det nyttig å presentere komponentene både i form av en generisk funksjon og personifisert i form av mennesker man ville ønsket å ha kontakt med i en "analog" verden. De fire komponentene som ble identifisert for CyberGuide var Kartograf (kartkomponent), Bibliotekar (informasjonskomponent), Navigatør (posisjoneringskomponent) og Budbringer (kommunikasjonskomponent). CyberGuide er hovedsaklig utviklet som en museumsguide.

- **Kartograf (kartkomponent)**

Denne komponenten har kunnskap om de fysiske omgivelsene, så som hvor bygninger, veier og interessante severdigheter inne i bygninger befinner seg. Den består av kart over disse omgivelsene hvor viktige elementer blir merket av, og brukeren kan lokalisere sin egen posisjon i forhold til disse.

- **Bibliotekar (informasjonskomponent)**

Denne komponenten har tilgang til samlingen med informasjon om severdigheter. Dette inkluderer beskrivelser av steder, bygninger og andre interessante severdigheter, samt informasjon om personer assosiert med de aktuelle områdene. Bibliotekaren skal kunne svare på spesifikke spørsmål om lokasjonene, så som "hvem jobber i den bygningen?", "hvilken kunstner malte det bildet?" eller "hvilke andre utstillinger er relatert til den jeg ser på nå?". Komponenten ble realisert gjennom et informasjonslager med relasjoner til objekter og personer av interesse i den fysiske verden.

- **Navigatør (posisjoneringskomponent)**

Siden interessen til brukeren hovedsaklig vil være i de nærmeste fysiske omgivelsene er det viktig å vite hvor han befinner seg til enhver tid for å kunne dekke det umiddelbare informasjonsbehovet. Navigatøren har derfor ansvaret for å holde rede på brukerens fysiske lokasjon. Denne komponenten implementerer derfor posisjonerings teknikker for å finne så presis informasjon som mulig om brukerens posisjon og orientering.

- **Budbringer (kommunikasjonskomponent)**

For å tillate kommunikasjon mellom flere brukere og mellom arrangør eller

administrasjon og brukere finnes en kommunikasjonskomponent. Denne Budbringeren tar seg av denne kommunikasjonen, og legger til rette for tjenester som chat og meldingslevering. I tillegg kan den brukes til å formidle posisjoner mellom brukere for å kunne lokalisere hverandre. Kringkasting av meldinger til et bestemt sett med brukere kan også være ønskelig, for eksempel ved å sende ut meldingen ”bussen vil gå fra stoppestedet om 15 minutter” til turister som kom med samme buss. Komponentene består i CyberGuide av et sett med trådløse kommunikasjonskomponenter for å dekke de forskjellige funksjonene.

Denne komponentbaserte arkitekturen legger til rette for en utvidbar og modulær systemutvikling. Utvidbar i den forstand at det hele tiden enkelt kan legges til nye moduler/komponenter, og modulær fordi de enkelte modulene hele tiden kan byttes ut med nye eller forskjellige versjoner uten at det påvirker de øvrige komponentene. Det ble for eksempel også vurdert å legge til en historiker komponent for å ta vare på informasjon om en brukers aktiviteter, reaksjoner og bevegelser med den hensikt å kunne optimalisere museumsbesøket for brukeren ved senere besøk. Ved å koble disse komponentene sammen på en hensiktsmessig måte legger man til rette for avanserte funksjoner og tjenester, som blant annet de som ble beskrevet under kapittelet om CyberGuide.

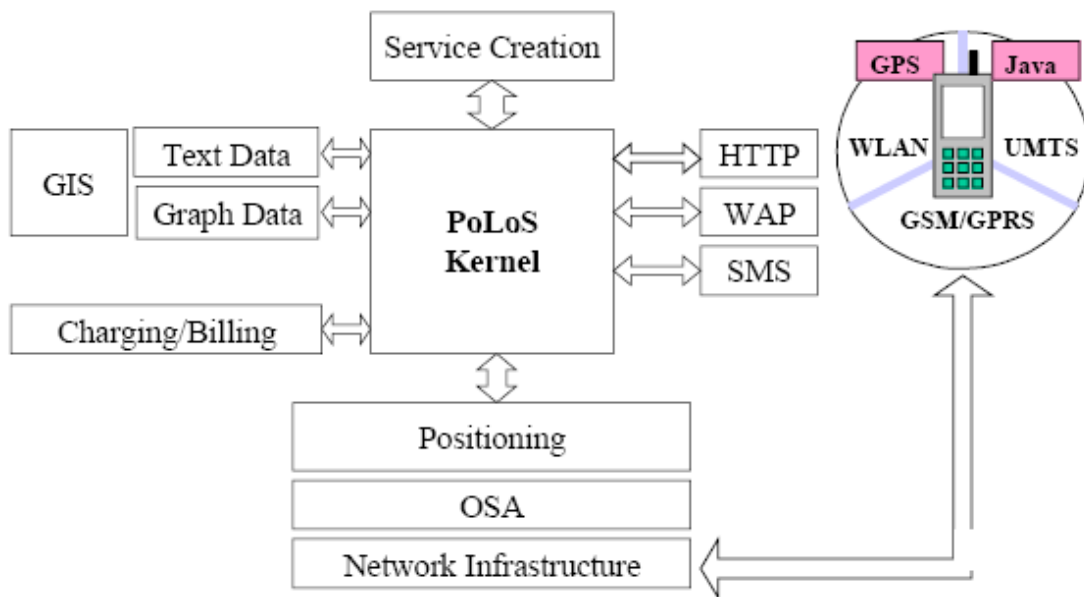
### 5.2.2 PoLoS

I artikkelen ”Using XML and related standards to support Location Based Services” [11] beskrives en komponentbasert LBS arkitektur kalt PoLoS, hvor man bruker XML for å spesifisere tjenestene. Den benytter en Kernel-enhet (moderenhet) som hele systemet bygges rundt. Kernelen koordinerer og utfører kommunikasjonen av tjenestene. Mens den gjør dette har den kontakt med en grensesnittkomponent, posisjoneringskomponent, GIS-komponent og en tjenesteskaperkomponent. Disse skal kunne erstattes av andre komponenter som gjør samme jobben så lenge de kommuniserer med Kernelen på samme måte (plug-in moduler). Dette legger i likhet med arkitekturen i CyberGuide til rette for en utvidbar og modulær systemutvikling. I tillegg er disse komponentene designet på en slik måte at de kan være distribuerte.

- **Grensesnittkomponent**

Grensesnittkomponenten håndterer kommunikasjonen mellom Kernel og eksterne entiteter, som hovedsaklig er brukernes klientenheter. Den støtter tre av de mest vanlige protokollsettene SMS, WAP og WWW/HTTP, men er designet på en slik måte at den støtter enkel implementering og integrering av nye fremtidige protokoller. Komponentene kommuniserer med Kernelens SIM (Service Invocation Module) gjennom en kompakt XML-basert protokoll spesifisert for anledningen. Grensesnittkomponenten ”forbereder” med andre ord informasjonen som blir overført i forespørsler adressert til Kernelen, mens Kernelen sender ut resultatet av tjenesteforespørselen.





Figur 24: Komponentbasert arkitektur i PoLoS

- **Posisjoneringskomponent**

Som navnet tilsier har denne komponenten ansvaret for å formidle posisjonsinformasjon til Kernelen. Den integrerer opptil flere underliggende posisjoneringsystemer og gir de et felles grensesnitt som abstraherer forskjeller mellom teknologiene. Det er implementert posisjonering for 2G/3G nettverk og WLAN, og det er også støtte for GPS gjennom klientbaserte moduler som pusher lokasjonsinformasjonen til posisjoneringskomponentens aktuelle modul.

- **GIS-komponent**

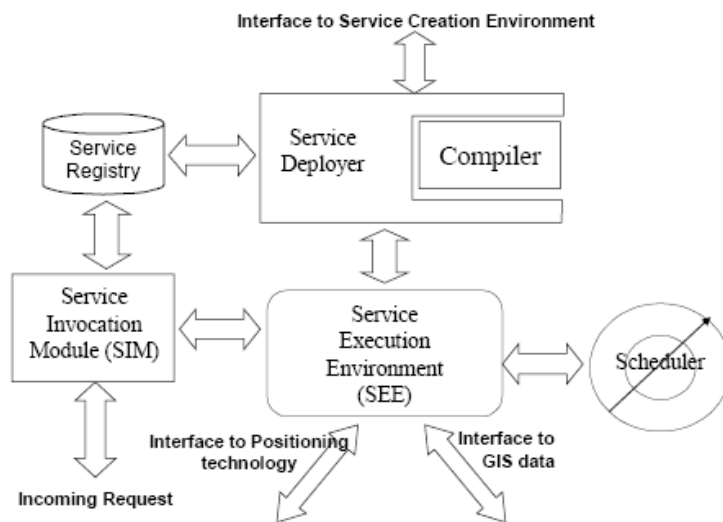
Denne komponenten har ansvaret for samhandlingen mellom Kernelen og eksterne lager med lokasjonsinformasjon. Dette inkluderer tekstlig såvel som grafisk informasjon i form av grafer og kart. Den håndterer også oppgaver som navigering og rutespesifisering. Den er bygd opp av en klientmodul på Kernelsiden og en servermodul på GIS-siden. Klientmodulen kommuniserer forespørsler og svar mens servermodulen tar seg av oppgavene med utenting og beregning mot informasjonslageret. Denne klient-server paradigmen tillater dermed at GIS-komponenten og Kernelen befinner seg på forskjellige lokasjoner og benytter forskjellige teknologier.

- **Tjenesteskaperkomponent**

I dette designet har utviklerne lagt spesielt stor vekt på enkel skaping av tjenester til et lokasjonsbasert informasjonssystem. Denne komponenten skal ta seg av nettopp dette. Den legger til rette for spesifisering, iverksetting og vedlikehold av tjenester gjennom et brukervennlig brukergrensesnitt. Tjenesten og dens logikk spesifiseres gjennom det nye språket SCL (Service Control Language) som er et XML-basert språk beregnet på spesifisering av nextgen (neste generasjons) telefontjenester.

- **Kernelkomponent**

Kernelkomponenten kan betegnes som selve hjertet i hele arkitekturen. Dette er enheten som tar imot og behandler en forespørsel på korrekt måte. Kernelen koordinerer all kommunikasjon mellom komponentene og er den enheten som tar seg av alle oppgavene som skal til for at systemet er operativt og kaller derfor de nødvendige komponentene når en etterspørsel krever det. Kernelen er designet med hensyn til enkel håndtering av de tilgjengelige tjenestene, og legger også til rette for påkalling av tjenester gjennom moduser forskjellig fra de vanlige forespørslene, som for eksempel tids - eller eventbaserte kall. For å håndtere disse oppgavene er Kernelen delt inn i fem mindre moduler som trengs for å gi full funksjonalitet til en tjenestes livssyklus:



Figur 25: Oppbygging av Kernelkomponenten

- The Service Deployer (SD)

Denne modulen tar imot nye tjenester ved å lese XML-filen som blir skapt ved hjelp av tjenesteskaperkomponenten. Gjennom tre steg blir først a) tjenestelogikken ekstrahert, b) selve tjenesten blir kompilert til kode sammen med eventuelle parametre og vedlegg som for eksempel html sider, og så c) registrert som ny tjeneste i systemet gjennom The Service Registry og satt i funksjon i The Service Execution Environment.

- The Service Invocation Module (SIM)

Her mottas forespørsler etter tjenester. Disse forespørslene kan komme fra eksterne klienter via grensesnittkomponenten eller fra kall innad i systemet (for eksempel tids - eller eventbaserte kall, som da vil komme fra The Scheduler). Den kontakter så The Service Registry for å finne den kalte tjenesten.

- The Service Execution Environment (SEE)

Denne modulen utfører selve tjenesten som har blitt kalt gjennom å kontakte de aktuelle komponentene, som i de fleste tilfeller vil være både GIS -og

posisjoneringsskomponenten, og videresende resultatet til grensesnittkomponenten som tar seg av kommunikasjonen med klienten.

– The Scheduler

Dette er modulen som støtter opp om push paradigmen for tjenester. Den innehar en rekke funksjoner for tidsplanlagte tjenestekall (engangsutførelser, daglige eller ukentlige kall osv.) og eventbaserte som for eksempel hvis en klient kommer innenfor et bestemt område eller bygning. Den vil i siste tilfelle motta beskjed om dette fra posisjoneringsskomponenten.

– The Service Registry (SR)

Informasjon om alle tjenestene i systemet er lagret i Service Registry. Etter at en ny tjeneste er implementert av The Deployer blir den registrert her. Denne informasjonen brukes når en tjeneste kalles. En tjeneste opererer med tre forskjellige tilstander i registeret: ”Startet”, ”stoppet” og ”stoppes”. Tjenester som er stoppet har ingen kjørende instanser i SEE. Tjenester som stoppes tillater ingen nye kall, men kan ha instanser allerede kjørende. Tjenesten skifter status til stoppet så snart disse har kjørt ferdig. En ny tjeneste har alltid status som stoppet og må eksplisitt startes av administrator etter den har blitt opprettet gjennom SD.

## 6 Lokasjonsbasert IS for studenter ved NTNU

I dette kapitlet vil det bli beskrevet et helhetlig overordnet systemdesign for et lokasjons-basert system for studenter ved NTNU Gløshaugen. Designet av et slikt system vil inneholde en rekke nyttige tjenester og implementere funksjoner som dekker informasjonsbehovet som nye studenter har. Dette vil selvsagt avhenge mye av selve innholdet i informasjonen og kvaliteten på denne, men gjennom oppgave er det fokusert hovedsaklig på organiseringen og struktureringen av informasjon med disse tjenestene i tankene, og den vil derfor fortsette med det.

### 6.1 Tjenester og funksjoner

Hovedfunksjonaliteten til systemet baserer seg på å kunne tilby studentene informasjon om de forskjellige stedene på campus mens de beveger seg rundt, slik at de blir kjent med omgivelsene, deres beliggenhet, funksjonalitet og generell relevans for dem som studenter. Mulighet for å søke opp ønsket informasjon bør også være til stede. Videre kan annen nyttig informasjon tilbys alt etter hva som egner seg for lokasjonen for å kunne bidra til økt bevissthet og til å dekke studentens informasjonsbehov hva lokasjonen angår. Bruk av interaktive tjenester er også et viktig aspekt som ble avdekket i arbeidet med state-of-the-art, og muligheten for brukerne selv til å påvirke informasjonsinnholdet. Et ideelt systemdesign vil også være såpass åpent i struktur og mulighet for konfigurasjon at det er overførbart til andre applikasjonsområder som involverer lokasjonsbaserte tjenester og levering av informasjon.

Under følger en liste over funksjoner dette systemet bør støtte, sortert i henhold til de identifiserte tjenesteparadigmene. Merk at mange av disse vil kunne passe inn som både push og pull type tjenester. Publish/subscribe paradigmen er ikke benyttet da denne glir rett over i push i denne sammenhengen.

#### PUSH

- Fortelle brukeren hvor han er
- Vise generell info om brukerens lokasjon
- Vise spesiell info om brukerens lokasjon
- Vise brukerens egen info om lokasjon
- Gi brukeren dynamisk oppdatert informasjon om lokasjonen (bl.a. event)
- Vise brukeren kart over område
- Vise brukeren hvor han er på kart
- Bli påmint ting fra huskeliste på stedet
- Bli gjort oppmerksom på oppdaget mål for stedet
- Lovere abonnement til angitt tid
- Lovere abonnement på angitt sted

- Levere abonnement på angitt tid og sted
- Globale meldinger til alle brukere/bestemte brukergrupper
- Globale meldinger til deltakere av event

## PULL

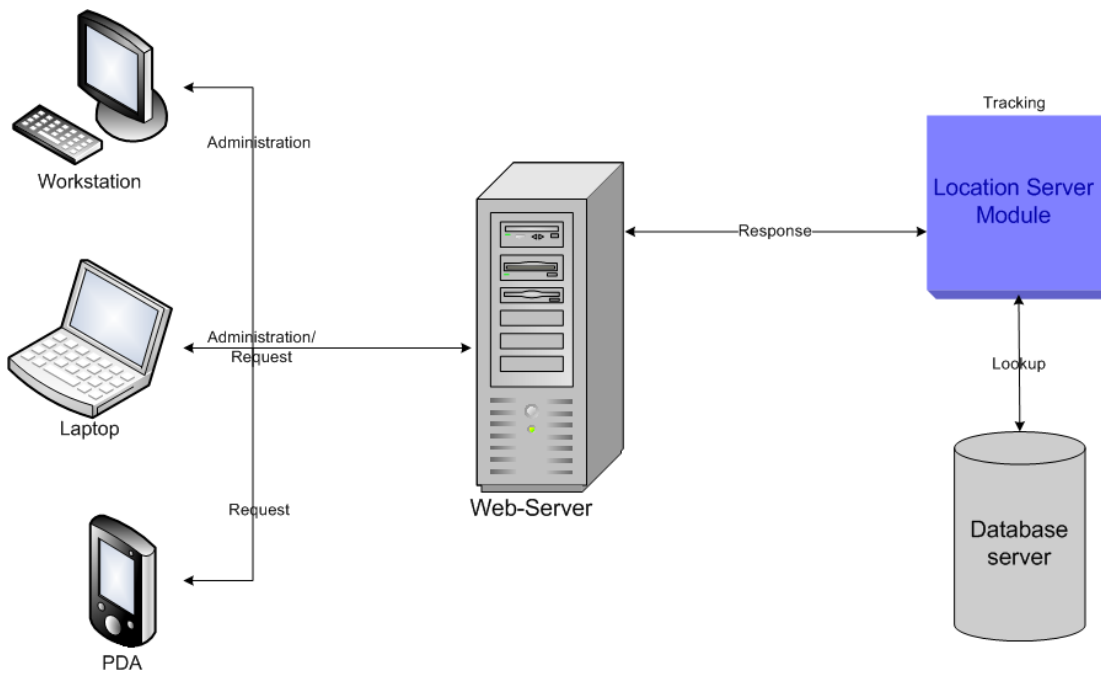
- Søke etter steder (ved navn)
- Søke etter informasjon om steder (ved beskrivelser)
- Søke opp nærmeste type sted
- Søke opp bestemte typer steder
- Vise brukeren kart over oppsøkt område
- Vise brukeren hvordan han kommer til oppsøkt sted
- Navigere fram til steder og info (via GUI)
- Navigere fram til steder og info (via kart)
- Hente fram metadata for steder
- Hente fram utvidet informasjon om steder
- Hente fram informasjon om events for sted
- Vise hvilke steder som kan sees fra brukeren lokasjon
- Vise andre aktuelle relasjoner til lokasjonen
- Finne (spore) andre medstudenter
- Finne (spore) andre deltakere (til event)

## INTERACT

- Interaktive tjenester for hver enkelt lokasjon (reservering o.l.)
- Legge inn egen informasjon om steder
- Lage huskelister for steder
- Lage "mål" (aktive spørringer)
- Dele egen informasjon med medstudenter
- Kommunisere med medstudenter
- La brukeren melde seg på til arrangementer (events) for steder
- La brukeren abonnere på eksterne info-tjenester (værmelding, kino etc)

Det ble på et tidlig stadium i planleggingen laget et sett med use cases som har blitt brukt som en rettesnor underveis til hvordan systemet skal fungere og hvilke tjenester man kan se for seg. Disse er å finne i appendix B.

## 6.2 Teknisk løsning

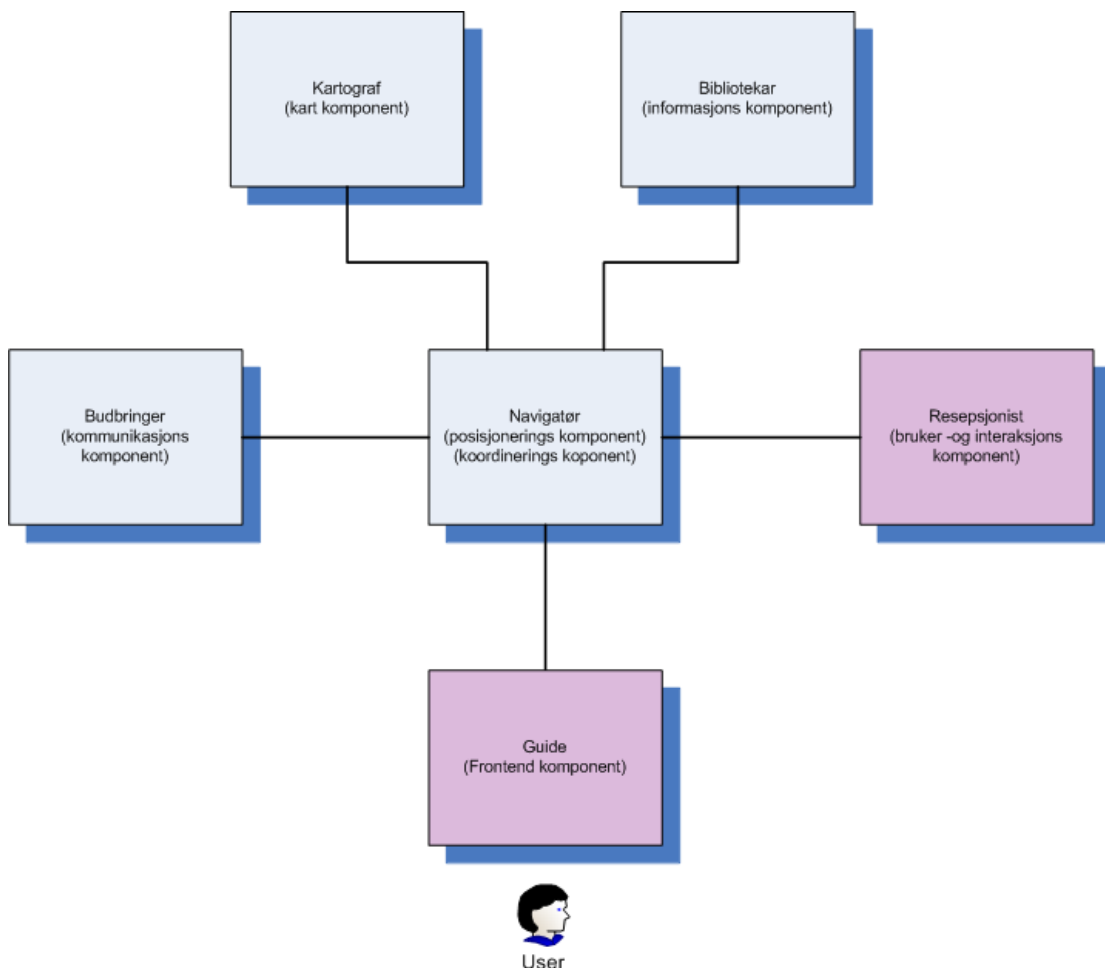


Figur 26: Ideell teknisk løsning for LBS systemet

I løsningen som er vist her har vi et typisk server-klient forhold med klientene til venstre for web-serveren, og serversiden består av alle delene til høyre inkludert web-serveren. Klientene har klient-software for systemet som kommuniserer med server-programvaren (Location Server Module) over det trådløse IP-nettverket. Klientene sender forespørsler som inneholder sensor data for deres lokasjon, det være seg GPS koordinater, node ID eller WiFi-signaler, til web-serveren som viderefremidler dette til Lokasjonsmodulen. Forespørselen blir sporet ved å koble sensor-dataene til lokasjoner i databasen. Videre blir aktuell eller forespurt informasjon slått opp for den aktuelle lokasjonen og sendt tilbake til web-serveren som igjen presenterer informasjonen for klientene.

### 6.3 Systemarkitektur

En veldig god og forståelig systemstruktur ble skissert i CyberGuide [1]. Det er derfor valgt å basere designet på denne, og benytte en lignende konseptuell og logisk struktur, siden den klart definerer oppgaver og samhandling mellom komponentene i systemet og er lett forståelig. Samtidig, slik vi ser i designet av PoLoS [11] er det behov for en styrende enhet som kan koordinere samhandlingen til disse komponentene, samt en enhet som kan ta seg av å interagere med brukeren og fremstå som en enhetlig entitet. Dette er viktig siden brukeren skal slippe å vite noe om systemets struktur, men bare kunne henvende seg til det for å få svar. De fire komponentene i blått tilsvarer mer eller mindre komponentene i CyberGuide. De to siste er delvis hentet fra PoLoS og til dels basert på behovet for å håndtere enkeltbrukere i systemet.



Figur 27: Ideell systemstruktur for LBS systemet

#### Navigator

Posisjoneringskomponenten har, i likhet med tilsvarende komponent i CyberGuide, ansvaret for å holde rede på hvor brukeren befinner seg, og har derfor

implementert de teknikkene og teknologiene som er nødvendig for å gjøre dette. Siden dette er komponenten som hele tiden holder rede på hvor brukeren befinner seg står den også sentralt når det gjelder å koordinere og kalle de øvrige komponentene, og tar derfor også på seg rollen som koordinator. Eventuelt kunne denne koordinerings oppgaven skilles ut i en egen komponent likt kernel-komponenten i PoLoS, men i dette designet sees det som hensiktsmessig å legge denne oppgaven til navigatøren siden denne er såpass sentral.

### **Bibliotekar**

Informasjonskomponenten har kontroll over all informasjonen i systemet og er koblet sammen med posisjoneringskomponenten for å finne informasjon om lokasjonen brukeren befinner på eller forespør. Den kan skaffe til veie all informasjon når den trengs i forhold til de andre komponentene, f.eks. fingerprint eller GPS data for en lokasjon når navigatøren skal posisjonere, informasjon om en lokasjon når guiden skal presentere den eller arkivere brukerskapt informasjon når resepsjonisten leverer den.

### **Kartograf**

Kartkomponenten har ansvaret for alle grafiske fremvisninger av kart og plotting av posisjoner på kartene. Den har også ansvaret for veivisning og beregninger av mulige veivalg og å finne bestemte lokasjoner som brukeren forespør.

### **Budbringer**

Kommunikasjonskomponenten har ansvaret for meldings -og kommunikasjons-tjenester mellom brukere og mellom systemet og brukerne. Dette kan være alt fra chat til e-post og globale kunngjøringer. I tillegg ville denne komponenten være sentral i utveksling av informasjon mellom brukere i systemet, så som deling av annotasjoner, bilder, huskelister og lignende.

### **Resepsjonist**

Bruker -og interaksjons komponenten håndterer abonnement på eksterne informasjonstjenester og henter disse og leverer de til brukerne når aktuelle kriterier er møtt. Den har også ansvaret for å holde rede på informasjon om brukeren og hans personlige informasjon i systemet, både personalia og skapt og eventuelt delt informasjon. Denne er farget lilla fordi det er en av komponentene som ikke tilsvarer noen av komponentene fra CyberGuide.

### **Guide**

I tillegg anses det som nødvendig med en Frontend komponent som har ansvaret for å formatere og presentere informasjonen på en god og hensiktsmessig måte for brukeren, og kunne ta imot og videresende input som brukeren kommer med til systemet. Denne er koblet rett til Navigatøren (eller koordinatoren) som igjen vil kalle de nødvendige komponentene i hvert tilfelle. Denne er også farget lilla for å illustrere at den er forskjellig fra CyberGuide designet.



## 6.4 Teknologi

I state-of-the-art kapittelet fikk vi en oversikt over teknologi benyttet i eksisterende lokasjonsbaserte systemer. Her vil det diskuteres valg av teknologi til et aktuelt lokasjonsbasert system.

### 6.4.1 Posisjonering

Valg av posisjoneringsteknologi er svært viktig når man planlegger å tilby lokasjonsbaserte tjenester, både med tanke på pris, ytelse, presisjon, kostnad med tanke på implementasjon og andre kompatibilitetshensyn. Et ideelt system bør, om ikke implementere, så i det minste støtte flere posisjoneringsteknologier for å kunne være både fleksibelt og utvidbart.

I et fullstendig system ville det blitt implementert WiFi Fingerprinting teknologi med tanke på innendørs posisjonering, og GPS eller AGPS for utendørs posisjonering. WiFi Fingerprinting er hensiktsmessig for denne type applikasjon fordi det er billig å implementere og har vist god ytelse og presisjon [10], og er kompatibelt med hvilket som helst allerede eksisterende WiFi utstyr. GPS eller AGPS er noe dyrere med tanke på utstyr, og vanlig GPS har en forholdsvis lang initieringstid fordi den må finne signal fra minst fire satellitter når den kommer utendørs. AGPS vil hjelpe på dette, men koster også mer. For å hjelpe på mulige dødszoner eller områder som er vanskelig å posisjonere for de andre valgte teknologiene kan beaconbasert bruk av Bluetooth enheter benyttes. Disse vil være ganske billige og kunne knyttes til tråd-baserte nettverk. De vil også være kompatible med WiFi Fingerprinting siden de bruker radiosignaler.

### 6.4.2 Klientutstyr

Klientutstyret er en viktig del av systemet, siden det er her brukeren vil møte og interagere med det. Siden et slikt system er ment å brukes mens man er i bevegelse bør dette utstyret være så lett og portabelt som mulig. Det første alternativet som da dukker opp er mobiltelefon. Mobiltelefonene blir stadig bedre og inkorporerer stadig flere teknologier. Det er også noe de fleste i dag har med seg over alt. Dog de fleste telefoner folk bruker til daglig har et display som fort kommer til kort når det gjelder å benytte denne typen innholdsrike tjenester, og mangler deler av teknologien som man ønsker å benytte til posisjoneringen. Det finnes en rekke svært gode smartmobiler med tilstrekkelig display og deler av teknologien man ønsker, så som WiFi, men disse er også kostbare og ikke noe de fleste benytter per i dag. Utsiktene for dette bare et års tid inn i fremtiden er likevel gode. Det neste naturlige alternativet er dermed PDA som i dag er det smartmobilene er på vei til å bli. De har forholdsvis store display, er kraftige og integrerer ofte de ønskede og nødvendige teknologiene som WiFi, GPS og Bluetooth. Disse er dog også ganske kostbare, men dette er stort sett en fellesnevner for alt slikt utstyr.

De to PDA'ene vi ser avbildet har begge store display, kraftige prosessorer, WiFi,



Figur 28: Ideelt klientutstyr for systemet: PDA

GPS og Bluetooth. Den ene av dem er også faktisk en mobiltelefon, men er tilgjengelig også den mest kostbare. Disse enhetene er ideelle for de fleste lignende formål som denne oppgave er rettet mot.



Figur 29: Ideelt klientutstyr for systemet: Laptop

Et siste alternativ er en laptop (bærbar PC) som svært mange studenter og andre benytter i det daglige. Disse er litt mer bryssomme å benytte mens man er i bevelse, men er i det minste portable. Det finnes mange små og enkle varianter i dag som egner seg bedre til dette formålet enn de kraftigste variantene som ofte blir litt mer bryssomme. Det er derfor også aktuelt at systemet bør kunne benyttes på bærbare PC'er som benytter aktuell teknologi ved kun å installere klientsoftware.

## 6.5 Info i systemet

Ideelt sett ville systemet inneholde all informasjon som kan synes nyttig for en student i hans eller hennes hverdagslige situasjon, med spesiell fokus på informative elementer ved lokasjoner. Siden dette kan være mer eller mindre hva som helst må dette konkretiseres med noen forslag.

Først og fremst ville denne informasjonen inkludere beskrivende informasjon om alle steder, dette være seg bygninger, rom, saler eller uteområder, som kan hjelpe studentene til å forstå hva som er viktig eller relevant for dem på det aktuelle stedet, eventuelt at det ikke er viktig for dem. I tillegg skal det være informasjon som hjelper studentene til å finne fram til disse stedene ved behov, og her er kartfunksjoner et naturlig valg. Videre kan man knytte annen relevant metadata til disse stedene som kan hentes fram ved behov, så som timeplaner, åpningstider, eventuelle kontaktpersoner, tilhørigheter, funksjoner, tilgjengelig utstyr og andre data man anser nyttig å kunne finne informasjon om.

Tilgang til eksterne informasjonskilder av både statisk og dynamisk art, som for eksempel vær og føre, trafikk og fremkommelighet, teater og kinoforestillinger, nyheter og lignende som kan knyttes opp i mot den øvrige informasjonen i systemet er ideelt.

Det ville også være nyttig med informasjon forfattet eller påvirket av brukerne selv i forskjellige former, som annoteringer, bilder, huskelister og så videre, og mulighet for å utveksle dette med andre.

Et annet aspekt er at informasjonen som gis til brukerne er individuelt tilpasset, og man trenger derfor i det minste basiskunnskap om brukeren. Derfor bør systemet benytte en eller annen form for tilpasset brukerprofil. Denne kan tilpasses av brukeren. I tillegg hadde mulighet for dynamisk oppdatering og systemlæring basert på brukerens valg og oppførsel i bruk av systemet vært ideelt, men dette grenser igjen over i funksjonalitet. Dette medfører også at noe informasjon om steder i systemet må finnes i flere versjoner slik at den mest aktuelle versjonen kan vises for en bestemt bruker.

En del informasjon finnes allerede på web og kan benyttes, blant annet informasjon om institutter og fakulteter og deres kontorer og tjenester, og informasjon om forelesningssaler og utstyr. Men en del informasjon må også forfattes spesielt for anledningen og tilpasses det begrensede formatet.

Det meste av informasjonen vil være være statisk, men det kan også være behov for dynamisk informasjon som hele tiden oppdateres. Det kan være info for hva som serveres til middag i de forskjellige kantinene, arrangementer som arrangeres i forskjellige bygninger for forskjellige linjer, bussruter for busstoppene og lignende.

## 7 Prototype

I dette kapittelet vil det bli gjennomgått et løsningsforslag i form av en prototype for å tilby lokasjonsbaserte tjenester for nye studenter ved NTNU Gløshaugen. Først kommer en analyse av informasjon som er blitt lagt til grunn for valg av elementer i lokasjonsmodellen, eller dataformatet om man vil. Videre lokasjonsmodellen blir presentert i sin helhet sammen med en databaseimplementasjon av denne. Til slutt presenteres en implementasjon av en klientapplikasjon koblet sammen med lokasjonsmodellen.

### 7.1 Analyse av informasjon

For å finne ut hva slags informasjon som er mulig, interessant og nyttig å knytte til lokasjoner med tanke på slike tjenester, er det gjort en analyse av informasjonselementer/metadadataelementer som kan inngå i en slik informasjonsstruktur.

Hva slags informasjon (stort og smått) finnes om en lokasjon?

- koordinater/posisjon
- type sted
- fasiliteter
- størrelse
- formål (hva gjør stedet/hva kan man gjøre på stedet?)
- forhold/miljø/klima (dynamiske forhold)
  - vær
  - trafikk
  - føre
- tilgang
- historie/fremtid
  - arrangementer
  - begivenheter
- tilstand (åpen/stengt/i drift/ødelagt etc)
- eierskap
- beboere/brukere

Basert på disse punktene, gazetteers og ADL GCS og på informasjonen fra systemene i State-of-the-art er det plukket ut et sett elementer som har potensiale til å kunne inngå i et lokasjonsbasert system for nye studenter. Disse er diskutert og beskrevet i mer detalj i de to neste underkapitlene.

### 7.1.1 Lokasjonsinformasjon

Lokasjonsinformasjon er informasjon som kan kobles til en lokasjon for å si noe om lokasjonen og også si noe om selve informasjonen. De tre førstnevnte, lokasjonsnavn, posisjon og lokasjonstype, tilsvarer de tre kjerneelementene i gazetteers. De fire neste, informasjon, informasjonsart, fasiliteter og nøkkelord, er beskrivende informasjon om en lokasjon. De to påfølgende, kart og veier, kan gi grafisk oversikt over et område og hvordan man skal finne fram. Måltype, målgruppe, prioritet og type/mime kan beskrive informasjon og hvem den er tiltenkt slik at brukere kan differensieres. De tre neste elementene, del av, synlige lokasjoner og relevante relasjoner, er alle ment å representere relasjoner mellom lokasjoner. Event representerer dynamisk informasjon for en lokasjon. Eksterne infotjenester kobler inn andre tjenester som kan relateres til en lokasjon. Til slutt har vi tidsaspekt som kan reflektere temporære forhold i informasjonen.

#### Lokasjonsnavn

Mennesker definerer steder og lokasjoner ved hjelp av navn. Akkurat som stedsnavnselementet i gazetteers bør vi derfor ha en post som holder på navnet på steder/lokasjoner som eksisterer. Dette skal ifølge en gazetteer være offisielle navn som representerer et sted/område. Her kan det også finnes alternative navn som stedet er kjent under. Det er derfor aktuelt å lenke dette til en tesaurus med alternative stedsnavn både for å kunne presentere disse som informasjon til brukerne, og for mulige gjenfinningsoppgaver hvis for eksempel en bruker søker etter et sted han har hørt nevnt ved navn.

#### Posisjon

Posisjon er en angivelse av hvor et sted eller en lokasjon befinner seg, altså informasjon om en lokasjons beliggenhet på jordens overflate. Dette er et av hovedelementene i gazetteers og et informasjonselement som må eksistere i alle slike systemer for å kunne knytte lokasjonen til en bruker opp mot et bestemt sted og dermed også tilhørende informasjon eller ressurs. I gazetteers angis dette som koordinater for lengdegrad, breddegrad og høyde over havet. LBS kan bruke disse koordinatene til å bestemme at en bruker befinner seg på et bestemt sted. Men for å kunne støtte alternative teknologier til GPS posisjonering bør også andre former for posisjonsangivelse kunne benyttes, noe som vil gjøre bruken av og bruksområdene til slik informasjon mer fleksibelt. Dette vil innebære velkjente GPS data i form av koordinater, sensoridentifikatorer for sensorsystemer som Bluetooth og IR, og også mobilteknologi som GSM.

#### Lokasjonstype

Lokasjonstype gir en indikasjon på hva slags sted en lokasjon fremstår som, for eksempel en bygning, et rom, et uteareal o.l. Dette tilsvarer det som kalles en navnetype i gazetteers, altså en kategorisering av typen sted. Det er en fordel for slik informasjon som dette at det spesifiseres ut i fra et kontrollert vokabular eller predefinert skjema slik at man unngår tvetydighet og lettere kan gjenfinne typer

lokasjoner man leter etter.

### **Informasjon**

De tre førstnevnte informasjonselementene tilsvarer de tre kjerneelementene vi finner i gazetteers. I tillegg til disse kan gazetteers inneholde ekstra informasjon om aktuelle steder. "Informasjon" beskriver altså selve nytteinformasjonen om en lokasjon - en mer utfyllende forklaring på ting som er relevant for og aktuelt med dette stedet. Denne typen fritekstbeskrivelse av et sted benyttes av de fleste lokasjonsbaserte turistguidesystemer, inkludert GUIDE [7], TouristGuide [30] og CyberGuide [1].

### **Informasjonsart**

Siden selve informasjonen som er knyttet til lokasjonene er noe av det som gir verdi til lokasjonsbaserte tjenester, bør denne også håndteres på en hensiktsmessig måte. Informasjonsart indikerer hva slags type informasjon som er beskrevet for en lokasjon, og legger opp til at en lokasjon kan ha et variert utvalg aktuell informasjon knyttet til seg. Det kan være ting som veibeskrivelse, generell opplysning, reklame, kunngjøring o.l. Dette bør spesifiseres ut i fra et kontrollert vokabular eller egnet skjema for å bevare integritet i dataene. Dette kan brukes til for eksempel å filtrere ut informasjon som skal vises for en bruker, eller aktivt for å finne bestemt informasjon.

### **Fasiliteter**

En bruker kan være interessert i å få en rask oversikt over en bestemt type fasiliteter i et område eller ved sin egen lokasjon med henvisning til eksakt lokasjon for fasilitetene, eller ønske å finne den nærmeste fasiliteten av aktuelle type. Denne informasjonen kan representeres gjennom f.eks. en tekstlig liste over fasiliteter, eller som ikoner merket av på et kart basert på hvor de befinner seg. CyberGuide [1] benytter slik informasjon i sitt system, men med begrensning til å bare se de forutbestemte lokasjonsfasilitetene som systemet er designet for å vise. En utvidning av dette ville være å la brukeren bestemme hvilken type fasilitet han ønsker å få oversikt over. Dette tillater TouristGuide [30].

### **Nøkkelord**

Nøkkelord er et element som forekommer i de fleste metadataformat, og er ment som en hjelp til informasjonsgjenfinning. Keywords inneholder derfor aktuelle nøkkelord som kan beskrive en lokasjon slik at den er lettere å finne i et typisk pull-basert søk. Her kan det være en fordel med et kontrollert vokabular med nøkkelord, men man bør ikke utelukke muligheten for fritekstbeskrivelser heller.

### **Kart**

I så og si alle eksisterende lokasjonsbaserte tjenester tilbys et kart i en eller annen form som viser brukerens lokasjon i forhold til omgivelsene, og mange gir også muligheten til å bruke dette kartet til å lete opp informasjon om disse omgivelsene

gjennom å velge steder og områder på selve kartet. I klassiske gazetteers som mange kjenner fra atlas, kan man slå opp stedsnavn i et register for så å finne hvilket kart og på hvilken side man kan finne det aktuelle stedet. I et elektronisk system vil dette fungere tilsvarende, og holde en peker til aktuelt kart over en lokasjon. Denne bør også kunne holde aktuelle "layers" eller versjoner av kartet (for eksempel inntegnede veier på kartet) for den gitte lokasjonen.

### **Veier**

Informasjon om hvordan man lettest kan komme seg fra en lokasjon til en annen i form av en veibeskrivelse eller opptegning på kart kan være nyttig i mange situasjoner. Dette kan dog fort bli ressurskrevende. Det finnes GIS-funksjoner som kan gjøre dette automatisk og i realtime fra tilfelle til tilfelle, noe som virker mye mer plausibelt enn å lage veibeskrivelser mellom alle lokasjoner. myCampus [28] benytter en slik funksjon i sin restaurantveileder, og GUIDE[7] systemet kan generere sightseeingruter for brukerne.

### **Måltype**

Når man snakker om informasjon som publiseres har man alltid en målgruppe som budskapet er rettet mot. Dette kan vi også benytte når det gjelder lokasjonsbasert informasjon, siden steder kan ha forskjellig betydning fra bruker til bruker. Måltype vil derfor kunne angi en eller annen form for gruppering av brukere som informasjonen er rettet mot, for eksempel studenter, ansatte, besøkende, turister, arbeidere o.l. Hvis det er snakk om generell informasjon til alle kan man for eksempel spesifisere en måltype "generell" eller "alle". Måltypene må være kontrollerte for at dette skal kunne fungere i praksis. Man kan spesifisere slike grupper nærmere ved å lage flere undergrupper av hver type, noe som leder oss videre til neste punkt, målgruppe.

### **Målgruppe**

Målgruppe angir hvilke brukergrupper innenfor brukertypene informasjon er rettet mot. Dette kan være for eksempel informatikk eller statsvitenskap innenfor måltype student. Innenfor typen arbeider kan det være for eksempel renholder, konstruksjonsarbeider eller vaktmester. Målgruppene må også være kontrollerte. Eksempel på bruk av disse inndelingene kan være hvis en "event" blir arrangert kun for informatikkstudenter. Hvis en informatikkstudent da er i nærheten av der hvor dette blir arrangert kan han få opp en melding om dette, mens en arkitekturstudent som passerer ikke vil få noen melding. Denne typen kontekstbevissthet er et av hovedpunktene i systemet myCampus [28].

### **Prioritet/begrensning**

For å kunne unngå overflod av informasjon som pushes til en bruker kunne det vært hensiktsmessig med en eller annen form for kontroll som brukeren selv er med på å definere etter ønske og behov. Et mulig forslag til dette kunne vært prioriteringsnivåer på informasjon som begrenser hvor høyt informasjonen skal være

rangert før den slipper igjennom, og brukeren selv skal kunne sette sitt personlige begrensingsnivå. Det kunne gjøres ved hjelp av poeng for viktighet på en skala fra 1 til 10, eller gjennom tekstlige benevninger som standard, middels, høy osv. Problemet med dette er at forfatterens oppfatning av viktighet ikke nødvendigvis er den samme som brukerens oppfatning, og vil derfor ikke alltid gi det ønskede resultatet hverken for forfatteren eller brukeren.

### **Type/Mime**

For å kunne begrense hvilke media som skal pushes til en bruker ville det være hensiktsmessig å kunne angi hvilken type som finnes i de forskjellige oppføringene for lokasjoner. Hvis en bruker har begrenset båndbredde eller kapasitet til å ta-  
le bestemte typer media kan disse filtreres ut. Det kan også benyttes dersom en bruker rett og slett ikke ønsker å se f.eks. filmklipp eller høre lyd.

### **Del Av**

Gazetteers inneholder, i tillegg til kjerneelementene, relasjoner som kan uttrykkes eksplisitt. For å kunne koble sammen steder og informasjon på en fornuftig måte trenger vi å kunne vite hvilke områder som omslutter hverandre og hvilke som inngår som del av et annet område. "DelAv" er derfor informasjon som forteller om en lokasjon er en del av et annen. Denne relasjonen vil virke begge veier, og dermed også fortelle hvilke deler som utgjør en lokasjon. F.eks. ville forelesningssal "F1" være en del av bygning "IT-bygget", eller sal "KJL1" en del av bygning "Kjelhuset". Denne typen relasjoner reflekteres bl.a. i GUIDEs[7] Environment Model hvor severdigheter lenkes sammen basert på beliggenhet i forhold til hverandre.

### **Synlige Lokasjoner**

Et annet sted-til-sted forhold som er aktuelt å representere er hvilke steder/lokasjoner som kan sees fra en gitt lokasjon. Dette blir påpekt i [6], hvor brukere ofte kunne tenkt seg å spørre en guide om ting som "hva er det der?" eller "Hva er det vi ser der borte?". Det er det derfor interessant å kunne uttrykke slike menneskelige oppfatninger av lokasjoner og deres forhold til hverandre. Med en slik relasjon ville det være mulig å få informasjon om andre steder/lokasjoner som er synlige fra denne lokasjonen.

### **Andre relevante relasjoner**

Det burde være mulig å representere andre sted-til-sted forhold som dukker opp som behov etter hvert, og kan representeres gjennom pekere til andre lokasjoner sammen med en beskrivelse av forholdet og kanskje en egen klassifisering av typen forhold. Den kan også for eksempel knyttes til bestemte brukertyper eller brukergrupper for å vise at et sted er relevant til et annet for nettopp denne brukeren. Forfatterne av informasjonen i systemet bør ha full frihet til å definere slike relasjoner, eller la de stå åpne.



### **Event**

En event er en form for tilstelning eller arrangement som kan annonseres på forhånd for et gitt sted. Informasjon om at noe vil foregå på en lokasjon kan leveres til brukere som befinner seg ved lokasjonen i et aktuelt tidsrom, omtrent som å henge opp en virtuell plakat. Ved å opprette et eget element for en slik event kan man også knytte annen informasjon til det, f.eks. brukerinformasjon ved at brukere kan melde seg på til arrangementet. Man får da også mulighet til utvidet funksjonalitet ved at man får segmentert temporære brukergrupper og kan sende ut meldinger spesielt rettet mot deltakere. Man kan også tillate sporing av brukere som er påmeldt arrangementet slik at man lettere kan finne igjen f.eks. en turgruppe eller seminardeltakere.

### **Eksterne infotjenester**

Informasjon av eksternt art kan hentes fra web eller RSS-postinger og knyttes sammen med de øvrige tjenestene. Slike tjenester kan være nyheter, vær og trafikkmeldinger, bussruter, kinoprogram og lignende som kan knyttes til lokasjoner. For eksempel kan man vise værmeldingen på Gløshaugen, eller oppgi bussruteinformasjon ved en busstopp basert på den aktuelle stoppen for daværende tidspunkt. comMotion [19] systemet benytter seg av slike tjenester og tilbyr planlagte værmeldinger og kinoprogram.

### **Tidsaspekt**

Siden spatial informasjon er temporær av natur, derav benevnelsen spatiotemporær informasjon, må man håndtere dette ved å angi gyldighetstider for informasjon og steder. Det kan tenkes at man registrerer informasjon om et sted som man vet vil være ugyldig etter en viss tid eller bare gyldig i bestemte perioder, og man bør derfor kunne angi dette slik at det kan tas hensyn til hva slags informasjon som presenteres i forskjellige situasjoner eller til forskjellige tider. En slik tidsbestemmelse ble benyttet i GUIDE [7] systemet for å tilby brukeren informasjon om severdigheter som var innenfor åpningstid. Det ble imidlertid påpekt at dette var kontekstinformasjon man måtte tenke nøye igjennom hvordan man benyttet, siden erfaringene fra GUIDE viste at brukere ofte gjerne ønsket informasjonen selv om severdigheten ikke var innenfor åpningstiden. Denne informasjonen kan uansett i minste grad benyttes som tillegginformasjon til en bruker for å gjøre ham oppmerksom på informasjonens gyldighet. Gazetteers angir vanligvis også tid - Når en lokasjon med et gitt navn begynte å eksistere og når den eventuelt sluttet å eksistere eller gikk over til å benytte et annet navn eller fikk omdefinert sin spatiale beliggenhet. Tid vil også kunne brukes til å angi "events" (hendelser eller arrangementer) som kan tenkes å annonseres og som kun vil være gyldige i korte perioder.

### 7.1.2 Brukerinformasjon

For å kunne tilby kontekstbasert informasjon utover lokasjonen til brukeren må man også ha en viss kunnskap om brukeren. En typisk måte å gjøre dette på er med en brukerprofil som inneholder alt fra demografisk informasjon til personlige preferanser. Dette kan kobles sammen med eventuell stedsinformasjon for å utvide kontekstbevisstheten og tilby bedre og mer beriket informasjon og funksjonssinnhold. Viktigheten av å utvide kontekstbevisstheten utover lokasjon blir påpekt i artikkelen "User needs for location-aware mobile services" [13].

De fire første elementene brukernavn, brukertype, brukergruppe og brukerinteresser er med på å gi brukerne en egenartet profil som kan brukes til å differensiere innhold i tjenester fra bruker til bruker. De fem neste elementene, deltakelse, annotasjon, 2-do liste, mål/dagsmål og abonnement, gir brukeren mulighet til "interact" tjenester og kan bli en mer aktiv deltaker i tjenestene. De fire siste, aktivitet/rolle, presisjon, informasjonsart og prioritering, gir mulighet for å kontrollere og begrense push-basert informasjon.

#### **Brukernavn**

For å ha unike brukere og kunne identifisere enkeltbrukere må man ha noe som entydig skiller de fra hverandre. Det er vanlig å benytte en bruker ID eller et brukernavn for å oppnå dette.

#### **Brukertype**

En enkel og god måte å starte på er å segmentere brukerne i mindre grupper slik at informasjon kan rettes mot spesielt mot eller benyttes til å filtrere bort urelevant informasjon. Ved å angi hvilken type en bruker er, f.eks. student, ansatt, besøkende, turist, arbeider e.l. er vi et godt stykke på vei til å dele opp brukerne.

#### **Brukergruppe**

Brukersegmenteringen kan gå over flere nivåer og dele brukerne inn i enda mindre grupper. Brukergruppe kan angi hvilken gruppe innenfor brukertypen en bruker hører til, f.eks. informatikk eller statsvitenskap innenfor typen student, eller rengjørere, konstruksjonsarbeider eller vaktmester innenfor typen arbeider.

#### **Brukerinteresser**

Dette er et typisk element for en brukerprofil, enten det dreier seg om web-profiler for surfing eller mailinglister og så videre. GUIDE [7] benytter også et slikt element i sitt User Environment for å kunne matche personlige interesser mot interessante turistlokasjoner. Denne informasjonen består som oftest av en liste over interesser som en bruker har, f.eks. data, musikk, idrett o.l., og kan blant annet knyttes opp mot lokasjoner og events for bestemte interesseorganisasjoner, bedrifter, foreninger o.l.

### Deltakelse

Dette er også en slags måte å segmentere brukere på, men er mer avhengig av sted og tid og vil dermed være mer dynamisk. En deltakelse vil svare til en "event", hvor en bruker kan være påmeldt som en deltaker, f.eks. til en forelesning, et seminar, møter eller lignende. Det kan også skape mulighet for segmentering basert deltakelse til mer faste anordninger som fag, medlemskap og lignende.

### Annotasjon

I artikkel [13] blir det påpekt at brukere alt for ofte blir sett på som passive informasjonskonsumenter, mens de ofte har både et behov og et ønske om å bidra til informasjonen i systemet. Dette blir også beskrevet som et ønske for videre arbeid i flere systemer, er delvis implementert i CyberGuide [1] og hovedformålet til Stick-e-Notes [26]. Annotasjon er derfor ment for holde personlig informasjon om et sted som brukeren selv kan manipulere. Denne informasjonen skal kunne aksesserer når brukeren befinner seg på det aktuelle stedet eller ved manuelt oppslag av brukeren, og det kan være aktuelt å la brukere dele annotasjoner med hverandre.

### 2-Do liste

I hverdagen har man ofte mye man skal gjøre og huske på, og mange bruker allerede elektroniske agendaer og påminnere for å lette denne byrden. Men man må likevel ofte huske sjekke selve agendaen eller huskelisten. MyCampus [28] har implementert en agenda som minner brukeren på noe han har å gjøre på aktuelle lokasjon når han befinner seg der, og noe av det samme gjør comMotion [19] gjennom en forefallsliste for lokasjoner. Brukerspesifiserte 2-Do lister (forefallsliste) som er knyttet til bestemte lokasjoner angitt av brukeren selv er derfor hensiktsmessig i et slikt system for å gi økt funksjonalitet og innhold for brukeren. Denne listen kan bli automatisk vist for brukeren når han befinner seg ved en aktuell lokasjon, eller innenfor et avgrenset område. Dette kan i tillegg til lokasjon begrenses av tidsbestemmelser for å øke kontekstbevisstheten og unngå forglemmelser.

### Mål/dagsmål

Det er ofte slik at man i løpet av en dag eller en uke har bestemte ting man ønsker å enten gjøre eller finne av en eller annen grunn, enten det er å få kjøpt en bok eller en ekstra perm, eller finne sted som har en scanner man kan bruke, men man tar det når det passer seg heller enn å løpe dit sporenstreks. Mål er derfor tiltenkt som et aktivt query (spørring) over ting som brukeren ønsker å finne i løpet av dagen eller uken, f.eks. sko, klokke, utested/restaurant, datasal, grupperom, institutt o.l. Hvis dette queryet/målet matcher noen av stedene hvor brukeren befinner seg i løpet av dagen kan det bli gitt beskjed om dette. MyCampus [28] benytter en slags utvidet versjon av dette i sin semantiske modell hvor en brukers online agenda brukes til å opplyse om at man befinner seg på et sted hvor man har ting å gjøre. Denne vil ikke være like kritisk som en 2-do liste og vil ikke være knyttet

til en bestemt lokasjon, men hjelpe å identifisere steder som kan møte målet.

### **Abonnement**

For å kunne benytte eksterne informasjonstjenester [19] må brukeren kunne spesifisere hvilke informasjonstjenester han er interessert i og når og hvor disse tjenestene skal leveres. Eksempler på dette kan være å få værmeldingen for lokasjonen brukeren befinner seg på hver morgen klokken 07.00, få vist de siste nyhetene til lunsjtid hver dag, eller få levert kinoprogrammet for helgen når brukeren forlater campus fredager etter klokken 16.00.

### **Aktivitet/Rolle**

[13] nevner muligheten for at brukere kan ha forskjellige preferanser ved forskjellige lokasjoner. Det er også mulig at brukere vil ha forskjellige behov alt etter hva slags situasjon de er i og hva de gjør på. For å øke kontekstbevisstheten rundt dette kan man la brukerne velge hva slags aktivitet eller rolle de er i, f.eks. skole, jobb, shopping, fritid, uteliv, sightseeing osv. Dette vil være noe av det samme som de fleste kjenner som profiler fra mobiltelefonen, hvor man kan skifte profil ut i fra om man er i møte, hjemme, på ferie og så videre for å begrense oppringinger, lydforstyrrelser med mer.

### **Presisjon**

Det kan være ønskelig for en bruker å påvirke presisjonen av lokasjonsutslag på forskjellige funksjoner, spesielt for push-baserte tjenester. For eksempel om brukeren ikke ønsker å få bestemt informasjon om et sted bare fordi han passerer forbi der innen en viss rekkevidde. Det kan derfor være hensiktsmessig å la brukeren spesifisere denne graden i en eller annen form.

### **Informasjonsart**

Enkelte typer informasjon kan være spesielt ønsket av en bruker, og et typisk eksempel vil være reklame. Det bør derfor være mulig å kunne blokkere enkelte typer informasjon og la andre slippe igjennom. Eventuelt kan det også være typer en bruker er spesielt interessert i og ønsker å finne mer enn andre.

### **Prioritering/begrensning**

Dette svarer til prioritet for viktighet på informasjon for lokasjonene for å kunne filtrere bort ønsket informasjon og unngå overflod av push-baserte leveringer. Som nevnt tidligere er dette vanskelig å bruke riktig i praksis, men er likevel nevnt her for å kunne ta det i betraktning.

## 7.2 Lokasjonsmodell

Lokasjonsmodellen/dataformatet spesifiserer hvordan informasjonen i systemet henger sammen, og danner grunnlaget for logikken i systemet. Den viser hvordan steder er strukturert i forhold til hverandre og i forhold til den øvrige informasjonen. Modellen er bygd rundt en gazetteer, og bygd videre basert på elementer og konsepter identifisert gjennom systemer i state-of-the-art og den foregående analysen.

### 7.2.1 CULBIS

På figur 30 ser vi en konseptmodell over lokasjonsmodellen/dataformatet som er blitt døpt CULBIS (Context-User-Location Based Information Structure). Her kan vi se hvordan informasjonselementene er koblet sammen for å danne grunnlaget for tjenestene som skal kunne tilbys.

Lokasjonsmodellen er basert på og bygd rundt en digital gazetteer-struktur, da gazetteers allerede er et anerkjent og mye brukt rammeverk for lokasjonsavhengig informasjon. I tillegg kan gazetteers i sin enkleste form svare på mange av spørsmålene som er aktuelle for ønsket funksjonalitet i et slikt system, i følge det vi kom fram til i kapittel 3.3.

I en typisk digital gazetteer representerer man lokasjonens beliggenhet gjennom koordinater i form av et punkt, en linje eller et areal representert ved flere avgrensede koordinater. Dette gjør at vi vet hvor på jorden vi skal lete etter en aktuell lokasjon. Noe som imidlertid ikke er tatt hensyn til er muligheten for flere posisjonsangivelser enn nettopp disse koordinatene. Slike koordinater er svært virkningsfulle fordi de både er menneskelig forståelige og kan måles/peiles gjennom utstyr som GPS. Men med tanke på å kunne tilby lokasjonsbaserte tjenester har vi ofte behov for å kunne uttrykke dette forholdet mellom geografisk navn og posisjon på flere måter for å kunne støtte flere teknologier. Som tidligere diskutert finnes det i dag mange teknologier og teknikker for å posisjonere en bruker, og svært mange av disse benytter ikke koordinater. Hvis disse skal kunne benyttes sammen med en gazetteer for å tilby informasjon og tjenester vil de være avhengige av en avansert oversettingsmodul eller lignende som derivierer koordinater ved hjelp av annen teknologi eller ferdigdefinerte matriser. Dette kan fort gi avvik og unøyaktigheter i posisjoneringen. En annen løsning er å utvide lokasjonsbegrepet i gazetteeren til å inkludere flere måter å angi posisjon på. For beaconbaserte lokasjonssystemer vil dette inkludere identifikatorer eller sett av disse, for mobilnett-posisjonering vil det inkludere celleidentifikator og eventuelle seksjonsidentifikatorer innenfor cellen, og for WiFi-basert fingerprinting selve fingeravtrykket for en lokasjon. Dette vil ikke gi noe større tap av presisjon enn det som allerede er reelt for teknologien, men krever at denne informasjonen er tilgjengelig på forhånd. Det er også store fordeler med dette ved at man kan anvende forskjellige teknologier på samme datagrunnlag og skifte mellom bruk av teknologien i realtime og etter behov, for eksempel når en person beveger seg mellom utendørs og innendørs omgivelser hvor en teknologi kanskje slutter å virke men

en annen kan overta. Dette er det derfor tatt hensyn til i den foreslåtte lokasjonsmodellen/dataformatet hvor det er støtte for både WiFi Fingerprints og GPS koordinater i henhold til ADL GCT. Det skal også være mulig å koble til nye lokaliseringselementer dersom det skulle bli ønskelig, uten øvrig omstrukturering. Lokasjonsmodellen er også designet for å kunne benyttes på tvers av flere applikasjonsområder ved at den implementerer tabeller som må ha definerte skjema for hvordan de skal fungere. Tabellen "Relasjon" er en av disse, hvor man spesifiserer hvilke typer relasjoner man ønsker å benytte for å kunne representere en mengde ulike relasjoner mellom lokasjoner. Man vil også i hvert tilfelle kunne spesifisere hvilken type bruker en gitt relasjon gjelder for, slik at man kan representere en relasjon som kun gjelder for denne brukeren. I prototypeimplementasjonen er de to relasjonene "er\_del\_av" og "kan\_sees\_fra" lagt inn. Førstnevnte relasjon vil stort sett være en nødvendighet for å få et slikt system til å ha tilstrekkelig sammenheng mellom lokasjoner, men dette vil selvfølgelig variere alt etter hva man ønsker å oppnå. "Brukergruppe" og "Brukertype" kan også tilpasses applikasjonsområdet. I prototypen er brukertypene ansatt, arbeider, student og besøkende lagt inn, med hver sine underklasser i brukergruppe-tabellen. Se forøvrig appendix del D for skjema. Lokasjonstyper kan også defineres alt etter hva man ønsker å representere av lokasjoner. I prototypen er det valgt å bruke et skjema som representerer lokasjoner som henholdsvis campus, bygning, etasje og rom for innendørs, og campus, plass og vei for utendørs. Disse er satt opp i ønsket hierarki. Funksjoner er også definert per lokasjonstype, hvor for eksempel rom kan ha funksjon som kontor, grupperom, arbeidsrom, kantine, auditorium og så videre. Bygning kan for eksempel være bibliotek, kantine eller lignende. Tabellen InfoArt kan også forhåndsdefineres for å kategorisere typer info som er tilgjengelig for hver lokasjon. For eksempel kan dette være "turistinfo" spesielt beregnet på turister, "standardinfo" for generell opplysning, "reklame" hvis det skulle være aktuelt, "veibeskrivelse" eller lignende. Tabellen "Kontekst" er også til stede men er ikke benyttet i prototypen. Denne var ment å være et valg brukeren kunne ha for å ha forskjellige preferanser alt etter hvilken situasjon han var i, som for eksempel på skole, på jobb, fritid og så videre.

Tabellen "Tid" skal representere tidsaspekter i modellen. Dette kan bl.a. være for å holde rede på når et sted er åpent eller stengt, hvor lenge informasjon av forskjellig art skal være gyldig, hvor lenge et arrangement varer osv. Følgende tidsaspekter er derfor identifisert:

- **Åpningstid**

Tidsrommet når en lokasjon eventuelt er åpen. Tider utenfor den angitte åpningstiden kan dermed regnes som stengt.

- **Gyldighetstid**

Dette kan være den tiden et informasjonsobjekt er gyldig fra det er lagt inn i systemet, og etter utgått tid rett og slett skal kasseres eller byttes ut.

- **Varighet**

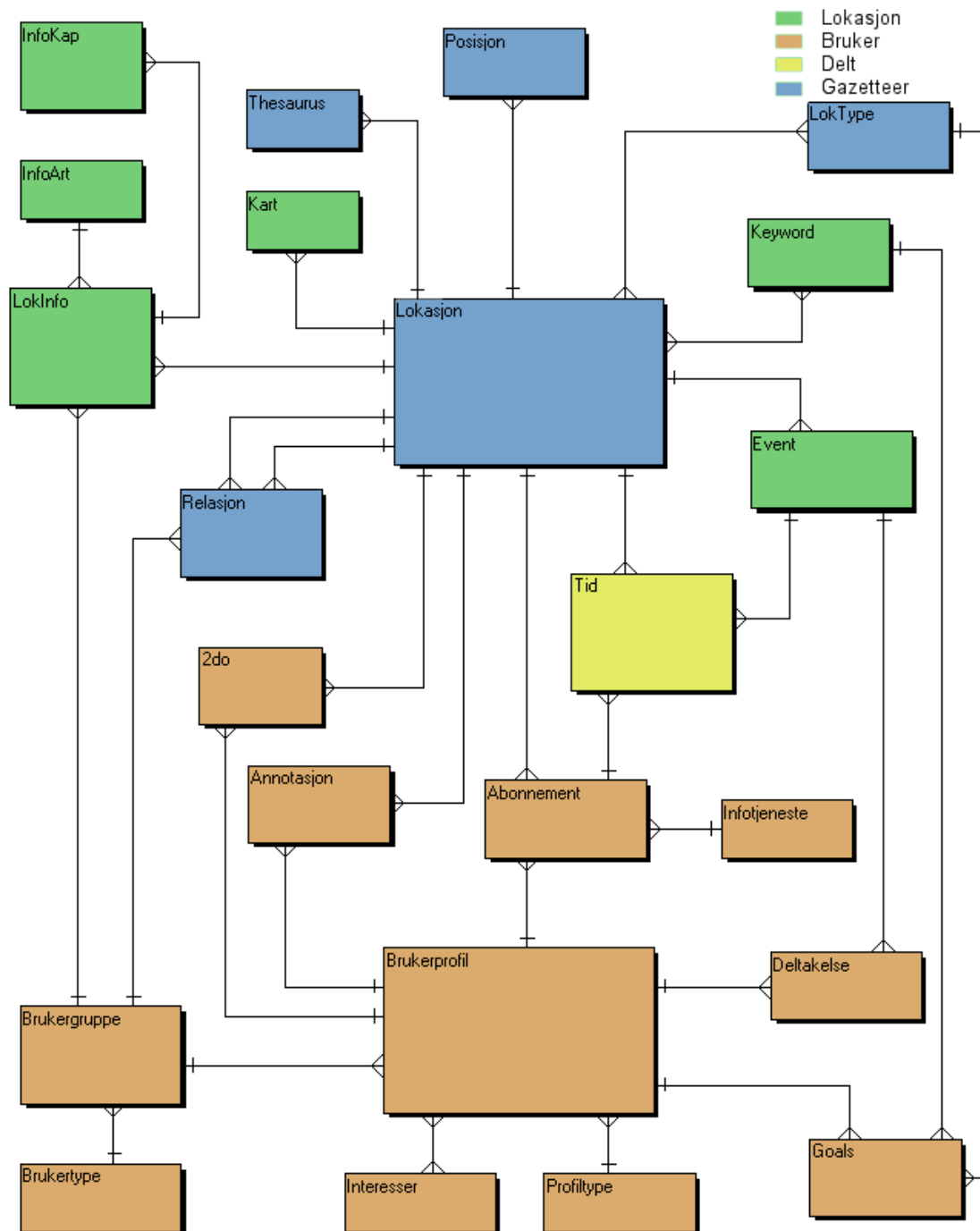
En hendelse eller arrangement vil ha en varighet, både når det gjelder tid for start og slutt, men også for tidsrommet informasjonen skal være aktiv. Man

kan da på forhånd legge inn et arrangement i systemet med spesifisering om at det skal annonseres fra og med en uke før arrangementet starter, og opplyse om tidsrommet for start og slutt.

- **Leveringstid**

Bestemte informasjonsobjekter eller tjenester kan ha bestemte tider for levering, noe som kan spesifiseres ved en leveringstid for aktuelle tjeneste.

I figur 30 ser vi hvilke elementer som representerer den rene gazetteer delen merket med blått. Lokasjon tabellen kan sammenlignes med geografisk navn, den holder informasjon som identifiserer en bestemt lokasjon. Posisjon tilsvare koordinater i gazetteers, men er her også i stand til å håndtere flere typer lokaliseringsdata. LokType inneholder en liste over typer steder en lokasjon kan være, kjent som navnetyper i gazetteers, organisert i et hierarki og koblet til Lokasjon for å plassere en gitt lokasjon i hierarkiet. De tre nevnte tabellene tilsvare kjerneelementene i gazetteers. I tillegg har vi en Thesaurus som ofte følger gazetteers, som holder alternative navn på steder. Digitale gazetteers holder også ofte relasjoner mellom steder som for eksempel forteller at et sted "er del av" et annet. Dette er her representert gjennom tabellen Relasjon. Denne tabellen har mulighet for å spesifisere sitt eget skjema over forskjellige typer relasjoner som kan utvides etter hvert hvis ønskelig. Hvis man da har spesifisert relasjonene "er\_del\_av" og "kan\_sees\_fra" kan man bruke disse relasjonene til å lage koblinger mellom steder. Eksempel på dette kan være "Auditorium F1 er del av IT-bygget" eller "IT-bygget kan sees fra Hangaren kantine".



Figur 30: Konseptmodell over informasjonen i CULBIS formatet. Her er den delt inn etter informasjon for lokasjon, bruker, delt informasjon og standard gazetteer elementer.





### 7.3 Tjenester og Funksjoner

Både lokasjonsmodellen og implementasjonen av klientapplikasjonen til systemet skal støtte et antall funksjoner og tjenester. I listen som følger finner vi igjen de samme funksjonene som spesifisert i kapittel 6. Lokasjonsmodellen er modellert for å kunne støtte opp om samtlige av disse funksjonene. Funksjonene som er implementert og testet i klientapplikasjonen er merket med en asterisk (\*). Noen funksjoner er også merket med et tjenestenummer. Disse vil inngå i eksempel som følger. Funksjonene er også her gruppert etter PUSH, PULL og INTERACT paradigmene.

#### PUSH

- Fortelle brukeren hvor han er (PUSH tjeneste 1) \*
- Vise brukeren kart over område (PUSH tjeneste 1) \*
- Vise brukeren hvor han er på kart (PUSH tjeneste 1) \*
- Vise generell info om brukers lokasjon (PUSH tjeneste 2) \*
- Vise spesiell info om brukers lokasjon (PUSH tjeneste 2) \*
- Vise brukers egen info om lokasjon \*
- Gi brukeren dynamisk oppdatert informasjon om lokasjonen (bl.a. event) \*
- Bli påmint ting fra huskeliste på stedet \*
- Bli gjort oppmerksom på oppdaget mål for stedet
- Levere abonnement til angitt tid
- Levere abonnement på angitt sted
- Levere abonnement på angitt tid og sted
- Globale meldinger til alle brukere/bestemte brukergrupper
- Globale meldinger til deltakere av event

#### PULL

- Søke etter steder (ved navn) \*
- Søke etter informasjon om steder (ved beskrivelser) \*
- Søke opp nærmeste type sted (PULL tjeneste 2 a og b)
- Søke opp bestemte typer steder \*
- Vise brukeren kart over oppsøkt område \*
- Vise brukeren hvordan han kommer til oppsøkt sted
- Navigere fram til steder og info (via GUI) \*
- Navigere fram til steder og info (via kart)
- Hente fram metadata for steder
- Hente fram utvidet informasjon om steder \*
- Hente fram informasjon om events for sted

- Vise hvilke steder som kan sees fra brukerens lokasjon (PULL tjeneste 1)
- Vise andre aktuelle relasjoner til lokasjonen
- Finne (spore) andre medstudenter \*
- Finne (spore) andre deltakere (til event)

## INTERACT

- Interaktive tjenester for hver enkelt lokasjon (reservering o.l.) (INTERACT tjeneste 2)\*
- Legge inn egen informasjon om steder (INTERACT tjeneste 1)\*
- Lage huskelister for steder \*
- Lage ”mål” (aktive spørringer)
- Dele egen informasjon med medstudenter
- Kommunisere med medstudenter
- La brukeren melde seg på til arrangementer (events) for steder
- La brukeren abonnere på eksterne info-tjenester (værmelding, kino etc)

De implementerte funksjonene er valgt ut fordi de representerer det vi kan kalle hovedfunksjonaliteten til systemet. I tillegg er noen ekstra funksjoner implementert for å teste ut og illustrere hvordan de virker sammen med lokasjonsmodellen. De som er valgt bort og ikke implementert i prototypen ble valgt vekk på grunn av manglende tid og ressurser i kombinasjon med kompleksitet, og i enkelte tilfeller på grunn av manglende datagrunnlag som for eksempel mangel på koordinater. Funksjoner som ville inngått i kartkomponenten er blant annet ikke implementert av sammen grunnene som at komponenten ikke er fullstendig implementert.

For å demonstrere hvordan selve dataformatet støtter disse funksjonene er det valgt ut et subset fra hver kategori som vil bli illustrert og gitt eksempler på. Noen av disse eksemplene inkluderer mer enn én funksjonalitet av gangen for å vise litt av sammenhengen. Hvilke funksjoner som inngår i hvilke tjenester kan sees i listen over.

Tjenestene som vil bli presentert er som følger:

- PUSH tjeneste 1: ”Du er her” (navn og kartangivelse)
- PUSH tjeneste 2: ”Informasjon om lokasjon”
- PULL tjeneste 1: ”Hva kan sees fra her/hva er der borte?”
- PULL tjeneste 2a: ”Hvor er nærmeste type lokasjon/fasilitet?” (koordinater)
- PULL tjeneste 2b: ”Hvor er nærmeste type lokasjon/fasilitet?” (relasjoner)
- INTERACT tjeneste 1: ”Legge inn min informasjon om her”
- INTERACT tjeneste 2: ”Bruke lokasjonens webtjeneste”

### **PUSH tjeneste 1: ”Du er her” (navn og kartangivelse)**

Dette er en av de mest elementære tjenestene og skal hjelpe brukeren i å finne ut hvor han befinner seg ved å gi ham navnet på lokasjonen han befinner seg, og et kart som angir hans posisjon i forhold til omgivelsene. Illustrasjon for tjenesten ser vi på figur 32.

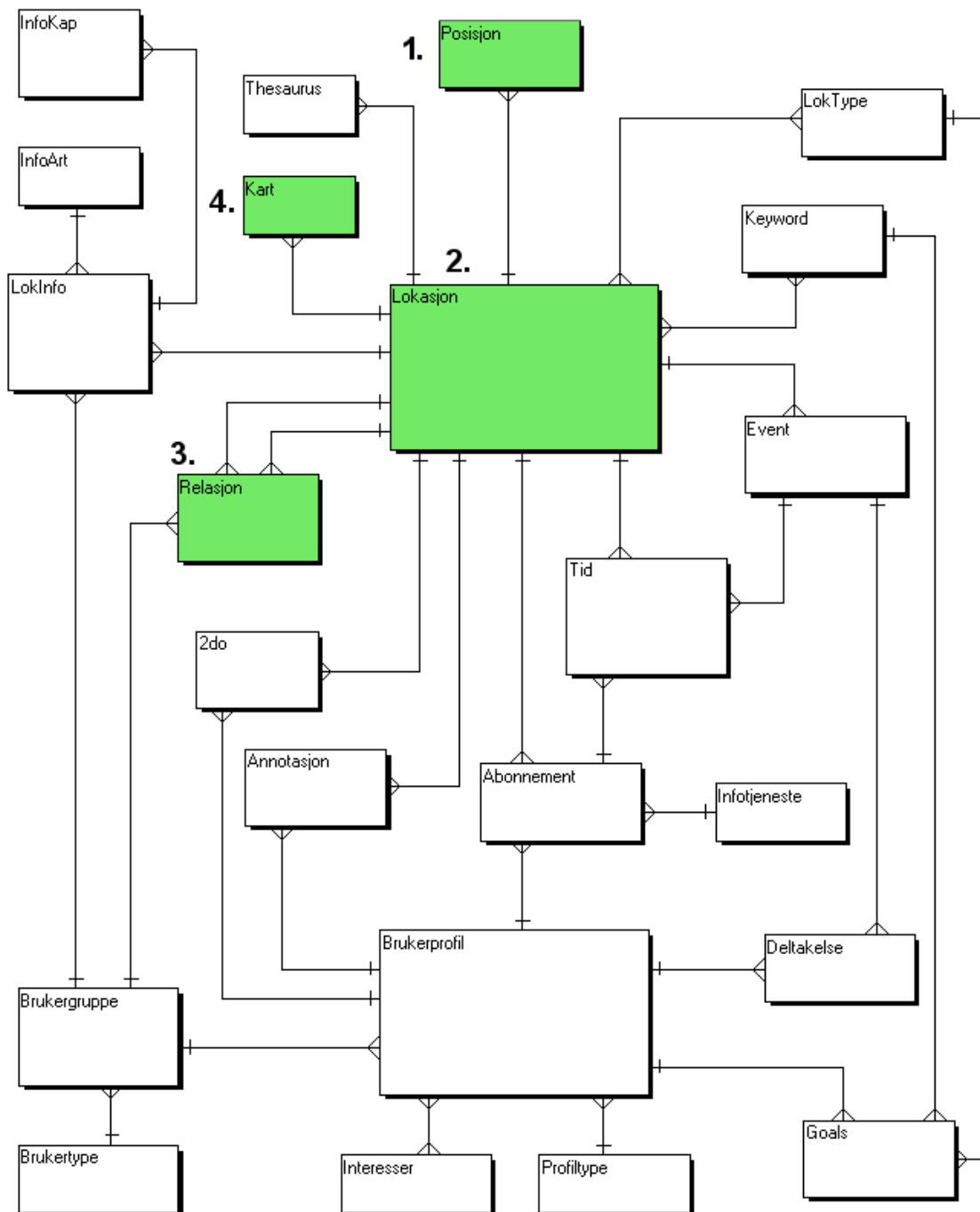
Denne tjenesten vil benytte følgende deler av CULBIS formatet:

1. Fingerprint fra brukerens lokasjon matches mot fingerprints i databasen.
2. Lokasjon for brukeren er identifisert og lokasjonsnavn hentes ut for å vises for brukeren og for videre oppslag.
3. For å angi sti for tilhørighet for lokasjon (bygning >etasje >rom) slå dette opp i Relasjon tabellen.
4. Kart for angitte lokasjon hentes ut og vises for brukeren.

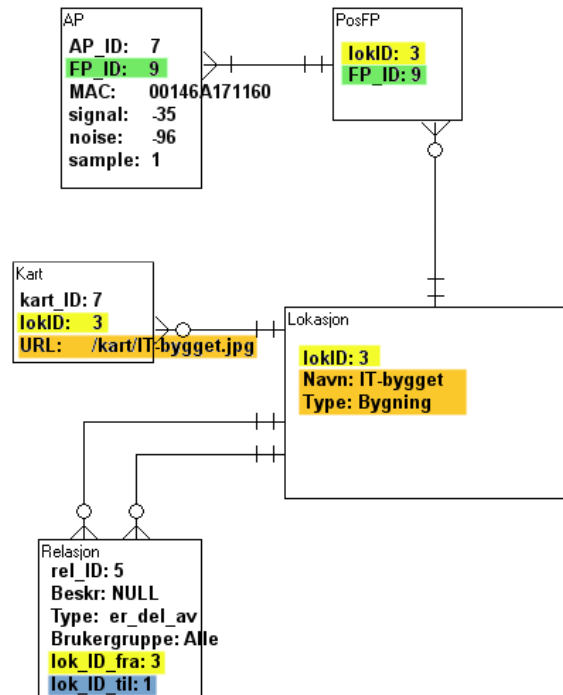
#### **Eksempel:**

En informatikkstudent, la oss kalle ham Lars, vandrer rundt på Gløshaugen Campus. I det han går inn i ”IT-bygget” spretter det opp informasjon som forteller Lars hvor han er og viser ham et kart av området og plasserer hans lokasjon på kartet.

Systemet vil forsøke å lete opp ”fingeravtrykket” som ligner mest på avtrykket som blir lest fra brukerens klient. Systemet vil gå igjennom alle aksesspunktene som danner fingeravtrykket og matche dette mot fingeravtrykket lest fra klienten. Fingeravtrykkene vil få en score basert på likheten mellom forespurt avtrykk og de i databasen. Hvis ett eller flere avtrykk er innenfor en forhåndsbestemt grenseverdi vil systemet regne den beste som en match. Et fingeravtrykk har gjerne flere ”oppføringer” som brukes for å forbedre presisjonen ved at denne kan gjenkjennes fremfor andre oppføringer i forskjellige situasjoner. ID’en merket med grønt (se figur 33) lenker et WiFi aksesspunkt til fingeravtrykket. Hvis eller når en match blir funnet vil systemet lete opp hvilken lokasjon som tilhører fingeravtrykket gjennom ID’en merket med gult. Den vil så hente ut informasjonen merket med oransje fra Lokasjon-tabellen og bruke ID’en merket med gult videre for hente adressen til kartet merket med oransje, og presentere denne informasjonen for brukeren. For å kunne vise tilhørighet for lokasjonen gjør den et oppslag i Relasjon tabellen og finner ut hvilken lokasjon den er en del av. Dette gjør den inntil den kommer til øverste nivå. Den kan da vise en ”spøringssti” av typen (bygning >etasje >rom) for lokasjonene.



Figur 32: Delene av CULBIS som benyttes for "Du er her" tjenesten.



Figur 33: Illustrasjon av informasjonen som brukes for "Du er her" tjenesten.

## PUSH tjeneste 2: "Informasjon om lokasjon"

Denne funksjonen går mer eller mindre hånd i hånd med den forrige tjenesten, og gir en kortfattet presentasjon av lokasjonen, beregnet på å vises på en håndholdt enhet av begrenset størrelse. I tillegg gir den mulighet for brukeren til å velge videre utdypende informasjon hvis det er ønskelig og tilgjengelig. Det er mulig å begrense og personlig tilpasse denne informasjonen basert på deler av brukerprofilen. Illustrasjon for tjenesten ser vi på figur 35.

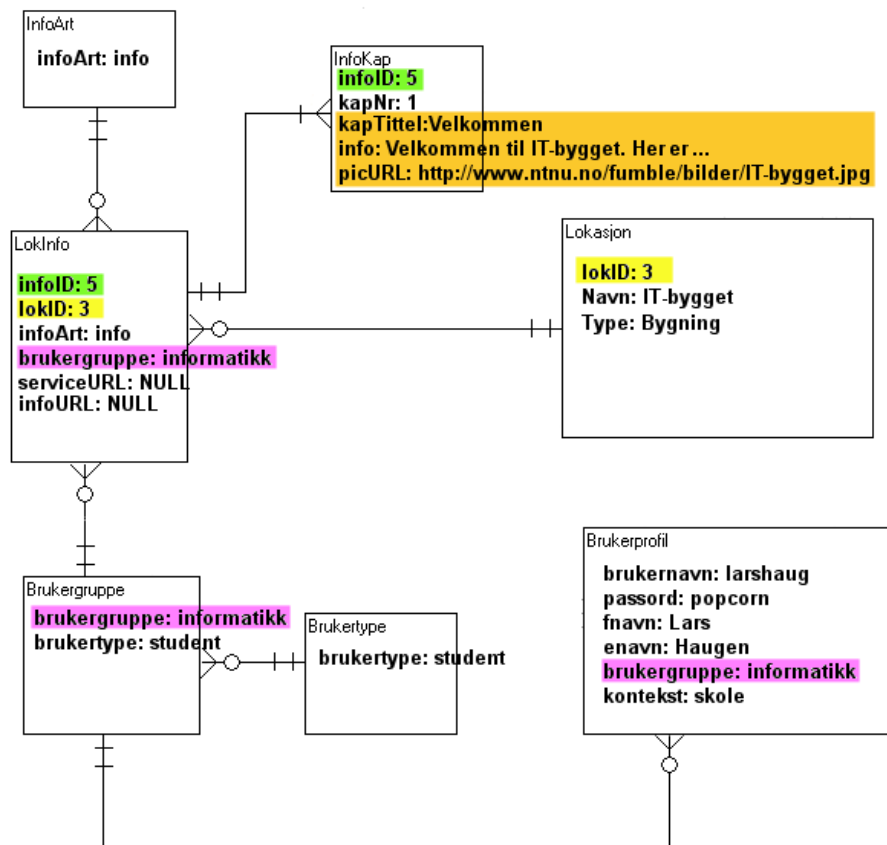
Denne funksjonen vil benytte følgende deler av CULBIS formatet:

1. Lokasjons ID for nåværende lokasjon hentes ut for videre oppslag.
2. Informasjonsobjekter for gjeldende lokasjon hentes.
3. Siden denne funksjonen skal gi nytteinfo sjekkes typen info (andre typer kan være reklame, veibeskrivelse, kontaktinfo etc).
4. For både å personlig tilpasse og begrense informasjonen filtreres den i forhold til brukerens brukergruppe og brukertype.
5. Initielt vises bare en kort beskrivelse, så hvis det eksisterer mer info og brukeren ønsker denne kan han velge å få vist ytterligere informasjon som ligger lagret som "kapitler".

**Eksempel:**

Lars har fått vite hvor han er og sett et kart over omgivelsene. Videre vil han få oppgitt litt informasjon om hva slags sted dette er og eventuelle ting som er interessant for ham her, og kan velge å finne mer informasjon hvis han ønsker det.

Denne tjenesten forutsetter at brukeren allerede har blitt posisjonert. Systemet leter opp informasjon som tilhører lokasjonen, se figur 34 (merket gult), og sjekker om det finnes elementer som er spesielt myntet på Lars ved å sjekke hvilken brukergruppe Lars tilhører og sammenligne dette med informasjonsobjektene (merket lilla). Hvis det finnes informasjon spesielt for informatikkstudenter velges denne, og informasjon fra første kapittel hentes ut fra "infoKap" tabellen, ID merket med grønt, infoen som hentes er merket oransje). Hvis Lars ønsker mer informasjon hentes neste kapittel fra tabellen.



Figur 34: Illustrasjon av informasjonen som brukes for "Informasjon om lokasjon" tjenesten.





**PULL tjeneste 1: ”Hva kan sees fra her/hva er der borte?”**

Denne tjenesten illustrerer hvordan relasjonsfunksjonene er tenkt, og i dette tilfellet skal det hjelpe brukeren i å orientere seg i omgivelsene ved å kunne stille spørsmål som ”Hva ser jeg der borte?”, ”hvilke bygninger ser jeg herfra?” o.l., og kunne velge å se lokasjonsinformasjonen om disse stedene. I andre tilfeller kan det være aktuelt å legge inn relasjoner som kan reflektere det man ønsker å gjenspeile. Illustrasjon for tjenesten ser vi på figur 36.

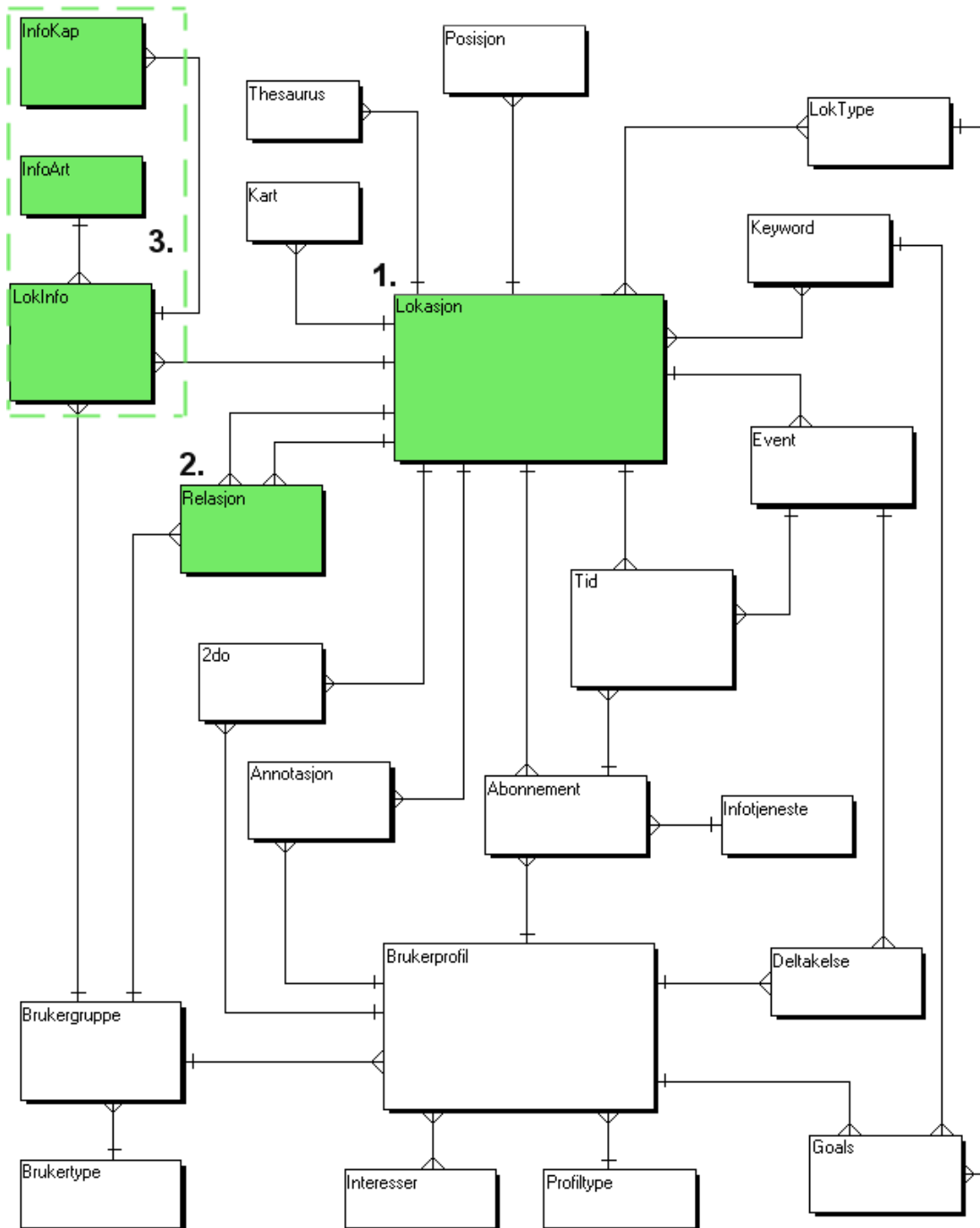
Denne funksjonen vil benytte følgende deler av CULBIS formatet:

1. Lokasjons ID for nåværende lokasjon hentes ut for videre oppslag.
2. Lokasjoner koblet sammen med relasjon av typen ”synlig\_fra” (evt. andre relasjonstyper) hentes fram.
3. Info om lokasjonene kan hentes og vises for brukeren hvis han ønsker det.

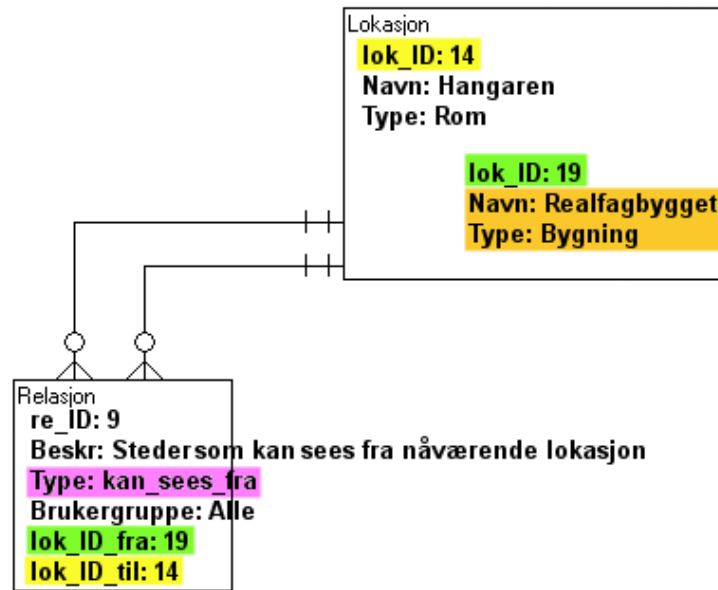
**Eksempel:**

Lars sitter i Hangaren kantine i Hovedbygget og ser ut av vinduet. Han ser sørover og ser Kjemiblokkene og inngangen til Realfagbygget. Han blir nysgjerrig og slår opp i systemet. Det forteller ham at fra Hangaren kan han se IT-bygget, Kjemiblokkene og Realfagbygget. Han velger Realfagbygget og får se informasjon om det.

Denne tjenesten forutsetter at systemet har posisjonert brukeren. Den leter opp relasjoner av typen ”kan\_sees\_fra” merket med lilla i figur 37, hvor lokasjonen man befinner seg på finnes i ”til” feltet merket med gult. Når dette er funnet kan man slå opp lokasjonen som er relatert til der man befinner seg, her merket med grønt, og presentere lokasjonen for brukeren, merket oransje. Når brukeren velger mer info om en lokasjon kan dette hentes frem på sammen måte som illustrert i de foregående eksemplene for informasjon, kart og lignende.



Figur 36: Delene av CULBIS som benyttes for "Hva kan sees fra her/hva er der borte?" tjenesten.



Figur 37: Illustrasjon av informasjonen som brukes for ”Hva kan sees fra her/hva er der borte?” tjenesten.

### PULL tjeneste 2a: ”Hvor er nærmeste type lokasjon/fasilitet?”

Denne tjenesten skal hjelpe brukeren å finne nærmeste lokasjon/fasilitet av en bestemt type i forhold til sin egen posisjon. Det er flere måter å gjøre dette på, alt etter hva slags informasjon som er tilgjengelig om lokasjonene, med varierende grad av presisjon. Illustrasjon for tjenesten ser vi på figur 38.

Ved bruk av GPS koordinater vil denne tjenesten benytte følgende deler av CUL-BIS formatet:

1. Lokasjons ID for nåværende lokasjon hentes ut for videre oppslag.
2. Lokasjoner av ønsket type blir funnet.
3. Koordinater for nåværende lokasjon og aktuelle steder blir hentet for å finne den nærmeste lokasjonen.
4. Nærmeste riktige lokasjon blir funnet og vist for brukeren.



**PULL tjeneste 2b: ”Hvor er nærmeste type lokasjon/fasilitet?”**

Hvis ikke GPS eller andre former for koordinater er til stede for lokasjonene kan man benytte en litt annen fremgangsmåte for å finne nærmeste lokasjon av en bestemt type. Denne er dog potensielt mindre presis, og benytter relasjoner mellom lokasjoner. Ved å benytte relasjonen ”er\_del\_av” kan man finne lokasjoner av valgt type som er del av samme lokasjon/scope [12] på neste hierarkiske nivå i lokasjonstype-tabellen. Hvis dette ikke gir noe resultat kan man gå ut til neste hierarkiske nivå og lete der. Denne metoden vil være mindre presis enn hva som er mulig med koordinater, men illustrerer godt hvordan denne typen relasjoner kan utnyttes. Se figur 39 for illustrasjon.

Ved bruk av relasjoner/scope vil denne funksjonen benytte følgende deler av CULBIS formatet:

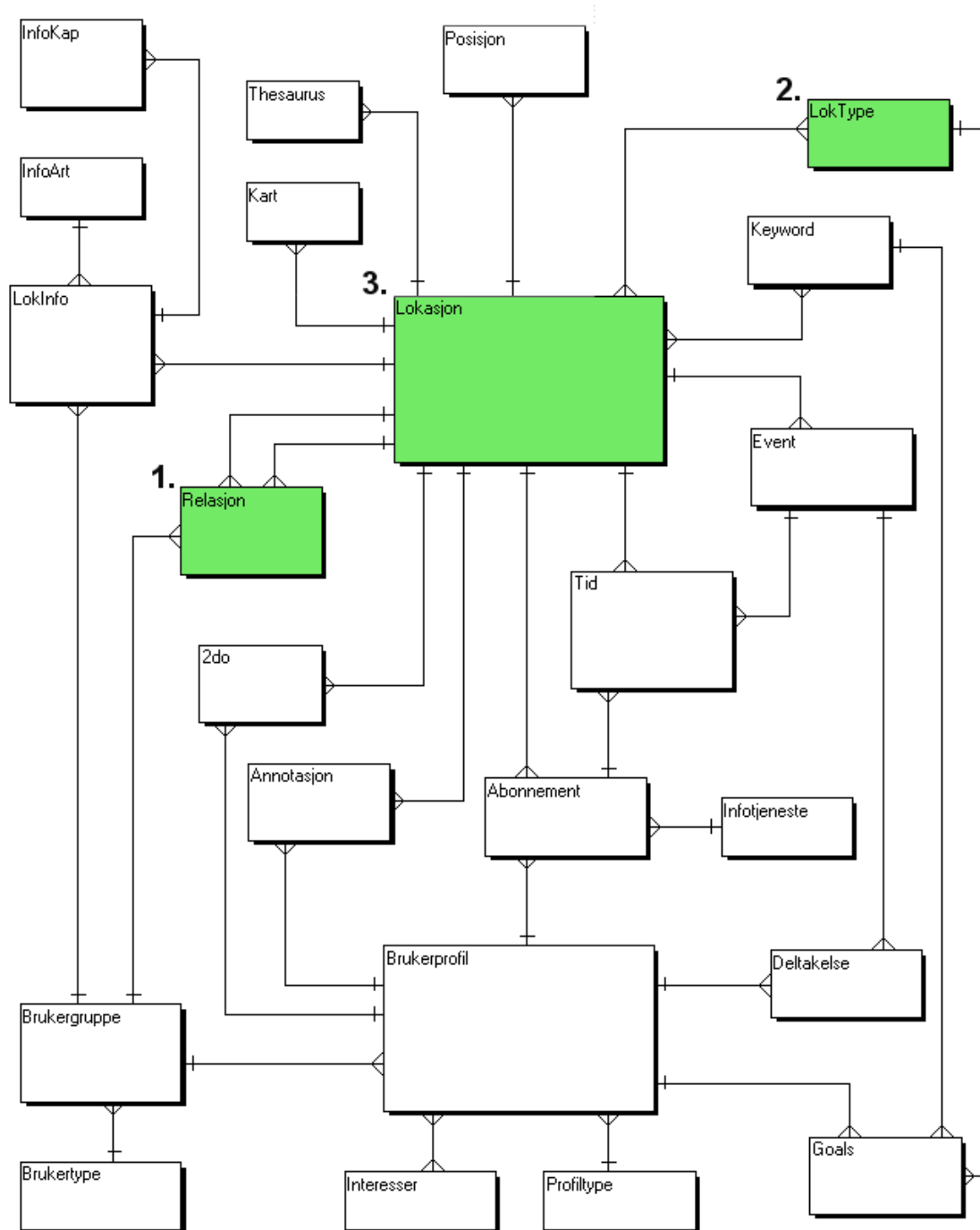
1. Finner relasjoner til lokasjoner under brukerens lokasjon/under overliggende lokasjoner til brukerens lokasjon.
2. Sjekker om etterspurt type lokasjon finnes på dette nivået.
3. Returnerer funnet lokasjon eller leter videre på neste nivå inntil den finner riktig type.

Denne tjenesten vil lete i syklor på hvert nivå inntil den finner en lokasjon som matcher forespørselen, eller inntil den ikke finner noen i hele tatt. Den finner først ut om typen lokasjon som etterspørres finnes på et hierarkisk nivå under brukerens lokasjon. Hvis ikke, finner den hvilke lokasjoner brukerens lokasjon er en del av på neste hierarkiske nivå, og finner lokasjoner av riktig type under disse. Hvis ikke den finner noe her heller leter den videre på neste hierarkiske nivå. En enda bedre måte å gjøre dette på ville sannsynligvis være å indeksere lokasjonene på forhånd og angi tilhørigheter som en litterær streng. Deretter kunne man mye enklere hente ut lokasjoner som matchet deler av denne strengen. Denne indekseringen måtte i såfall ha benyttet en lignende fremgangsmåte som beskrevet.

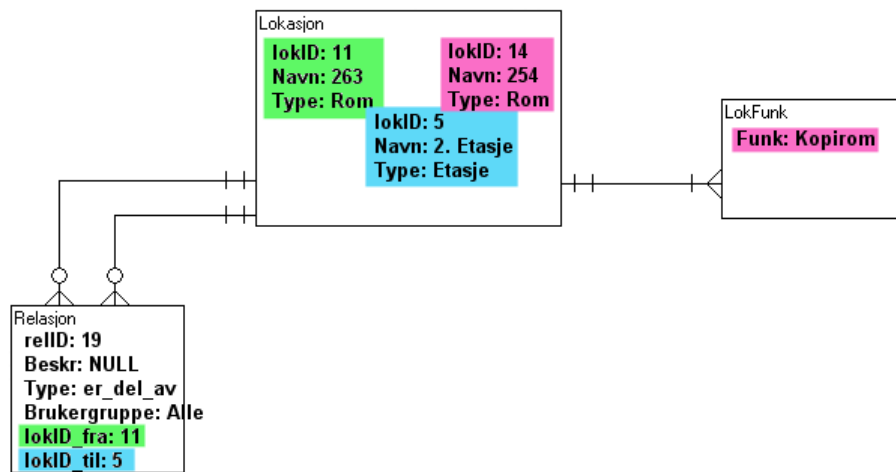
**Eksempel:**

Lars trenger å ta noen kopier, og vil derfor gjerne finne nærmeste kopirom. Han benytter systemet for å spesifisere at han ser etter nettopp et kopirom. Systemet vet at Lars befinner seg på arbeidsrom 263, og begynner så å lete etter nærmeste kopirom.

Systemet tar utgangspunkt i hvor brukeren befinner seg, merket med blått på figur 40. Den slår opp i relasjonstabellen og finner ut om dette rommet innehar andre lokasjoner av funksjonstypen ”kopirom”. Når den ikke finner dette på noen underliggende nivåer går den et nivå opp, i dette tilfellet til ”2. Etasje”, merket med grønt. Den sjekker dermed om denne lokasjonen har lokasjoner under seg av funksjonstypen ”kopirom”. Her vil den finne ”254”, merket med lilla, anta at dette er det nærmeste kopirommet til ”263” og vise dette for Lars. Eventuelt vil den finne flere kopirom på denne etasjen og vise samtlige for Lars.



Figur 39: Delene av CULBIS som benyttes for "Hvor er nærmeste type lokasjon/fasilitet" funksjonen med relasjoner/scope.



Figur 40: Illustrasjon av informasjonen som brukes for "Hvor er nærmeste type lokasjon/fasilitet?" tjenesten.

### INTERACT tjeneste 1: "Legge inn min informasjon om her"

Denne tjenesten lar brukeren legge inn egendefinert informasjon om lokasjonen som han finner nyttig, og som kan hentes frem igjen ved senere anledninger. Se illustrasjon på figur 41.

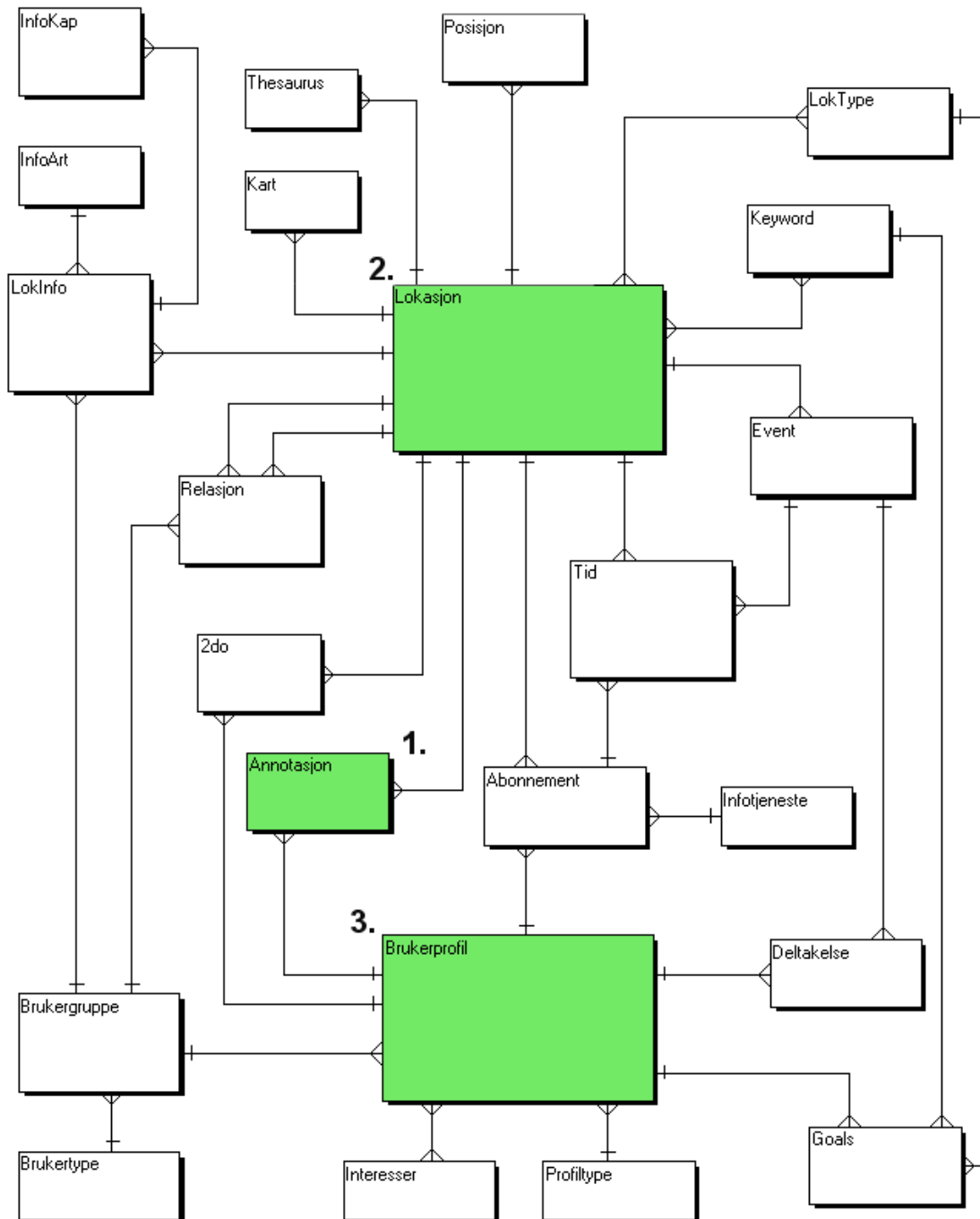
Denne tjenesten vil benytte følgende deler av CULBIS formatet:

1. Tar imot notater fra brukeren angående lokasjonen og lagrer det som en annotasjon.
2. Kobler annotasjonen til lokasjonen brukeren befinner seg på.
3. Kobler annotasjonen til den bestemte brukeren.

#### Eksempel:

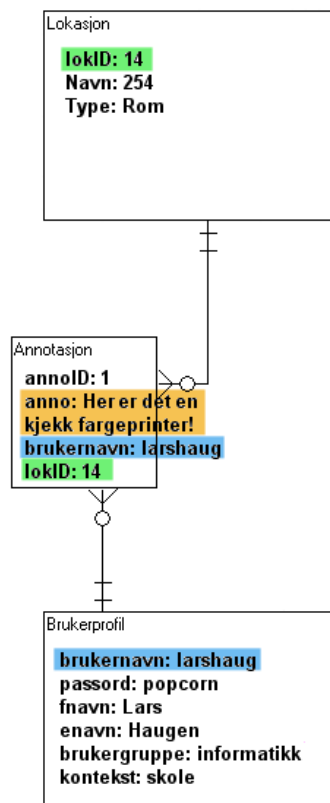
Da Lars hadde funnet kopirommet fant han ut at det var en veldig fin fargeprinter på dette rommet. Han ville merke seg dette og kanskje tipse noen av sine kompis om det, og la derfor inn en annotasjon på dette stedet.

Systemet tar utgangspunkt i hvor brukeren befinner seg merket med grønt på figur 42, tar i mot annotasjonen til Lars fra klienten, merket med oransje, og kobler den til denne lokasjonen. Videre kobler den annotasjonen til Lars' brukerprofil, merket med blått.



Figur 41: Delene av CULBIS som benyttes for "Legge inn min informasjon om her" tjenesten.





Figur 42: Illustrasjon av informasjonen som brukes for ”Legge inn min informasjon om her” tjenesten.

## INTERACT tjeneste 2: ”Bruke lokasjonens webtjeneste”

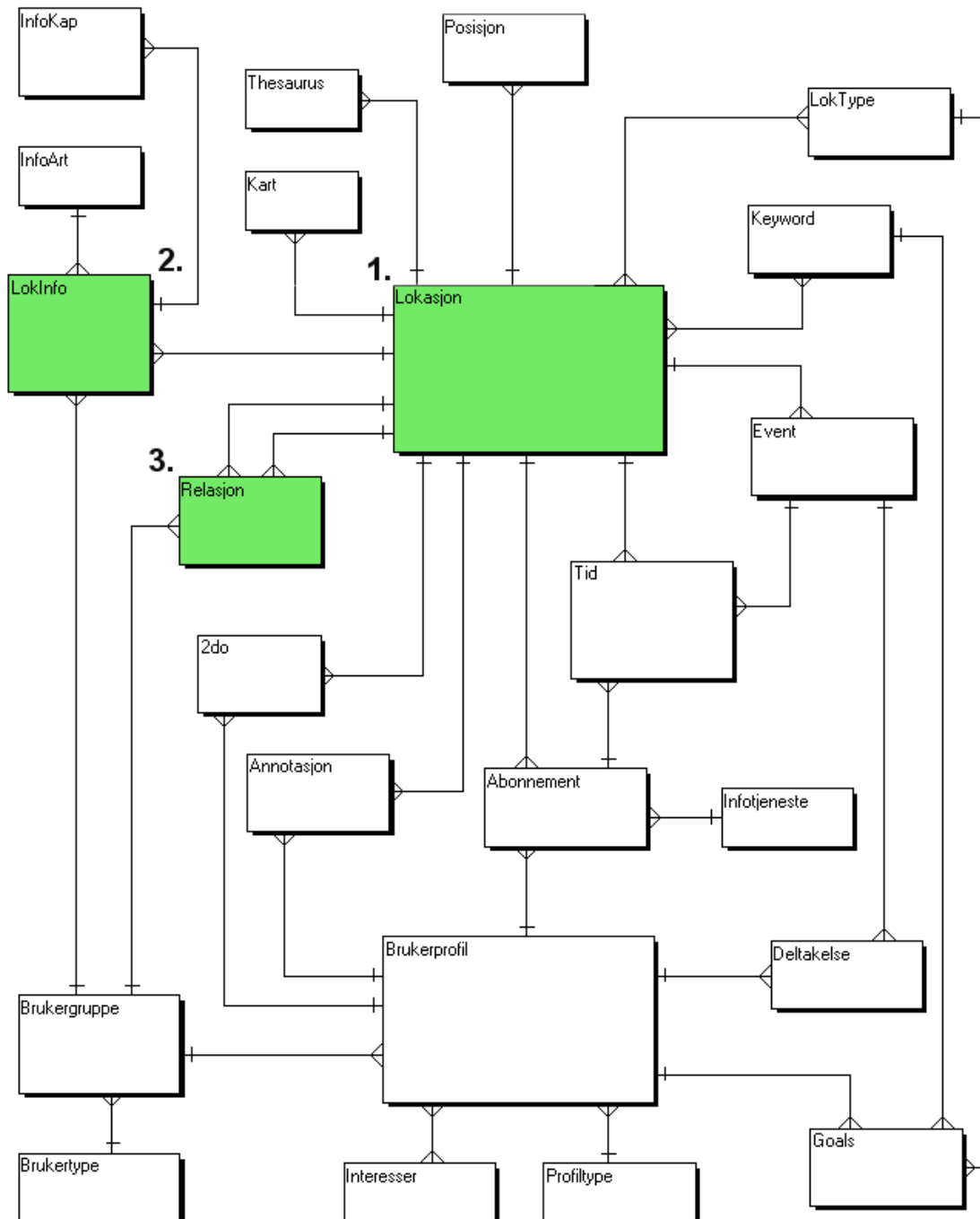
Denne tjenesten lar brukeren benytte en lokasjons webtjeneste hvis det eksisterer. Funksjonen henter da ut tjeneste-URL og viser denne i klientenhetens hovednettleser. Derfra er ansvaret overlatt til de forskjellige lokasjoners egne webservere. Se figur 43.

Denne tjenesten vil benytte følgende deler av CULBIS formatet:

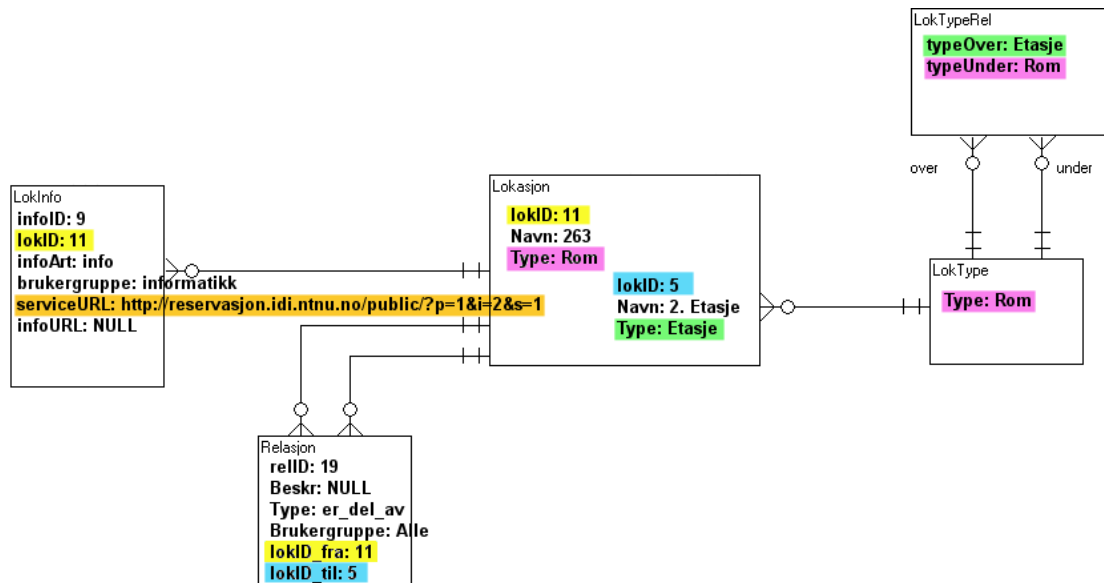
1. Finner lokasjonen brukeren befinner seg på.
2. Henter ut tjeneste-URL hvis det finnes og åpner denne i klientens hovednettleser.
3. Alternativ hvis tjeneste URL ikke finnes: Finn overordnet lokasjon og bruk dennes webtjeneste.

### Eksempel:

Lars ønsker å reservere arbeidsplass på rom 263 sammen med noen av klassekameratene sine. Han har blitt med dem inn på rommet, og velger å bruke lokasjonens webtjeneste i systemet.



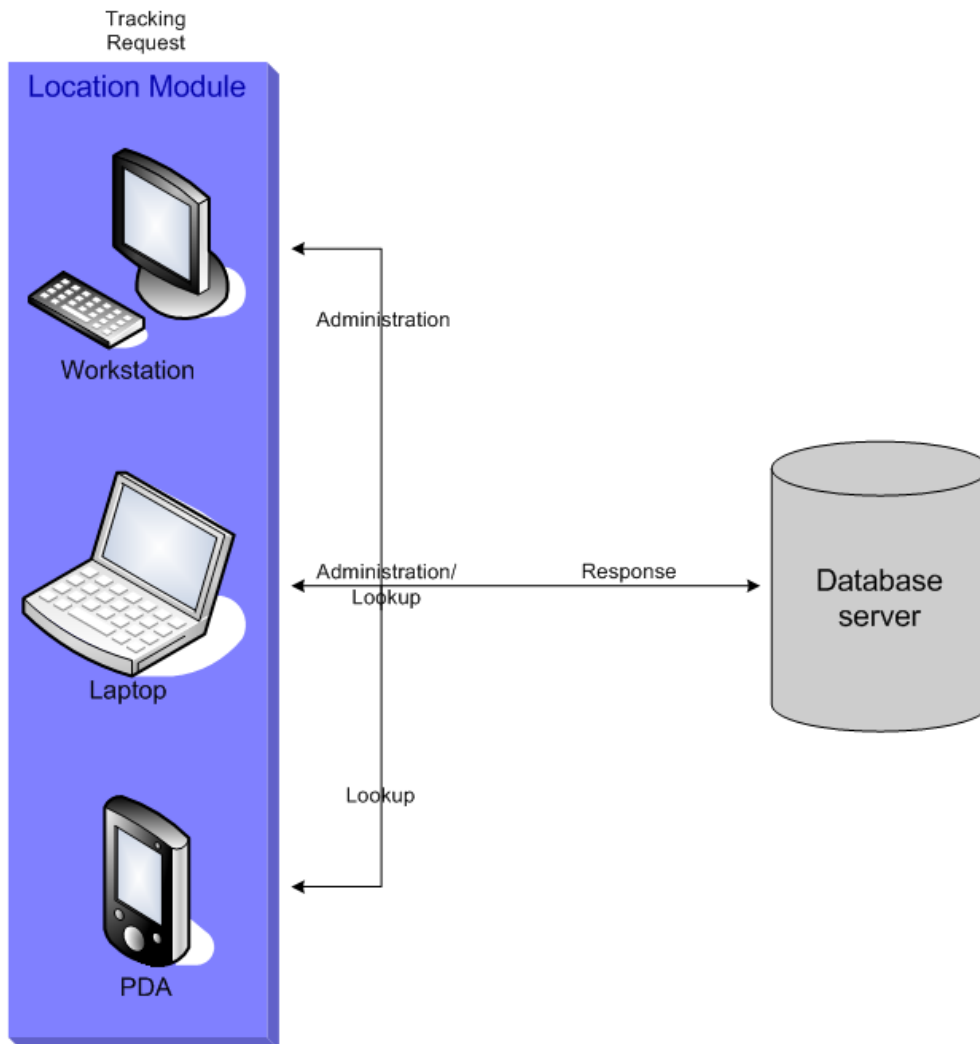
Figur 43: Delene av CULBIS som benyttes for "Bruke lokasjonens webtjeneste" tjenesten.



Figur 44: Illustrasjon av informasjonen som brukes for "Bruke lokasjonens webtjeneste" tjenesten.

Systemet tar utgangspunkt i hvor brukeren befinner seg, merket med gult på figur 44. Systemet finner informasjonsobjektet for lokasjonen, henter ut URL for webtjeneste, merket med oransje, og åpner dette i klientenhetens web-browser. Hvis det ikke eksisterer en webtjeneste for aktuelle lokasjon kan systemet gjøre tilgjengelig webtjenesten for den overstående lokasjonen, i dette tilfellet for etasje eller eventuelt bygningen. Dette gjøres ved å slå opp i relasjonstabellen og finne lokasjonen som rommet er en del av. Den overstående lokasjonen i relasjonstabellen vil selvfølgelig være av den overstående typen i lokasjonstype-tabellen (her illustrert med lilla og grønt. Merk at disse ikke trenger å sjekkes i dette tilfellet da en integritetssjekk skal sikre at dette er riktig når lokasjoner registreres). Hvis en webtjeneste finnes her kan systemet la brukeren benytte denne.

## 7.4 Teknisk løsning

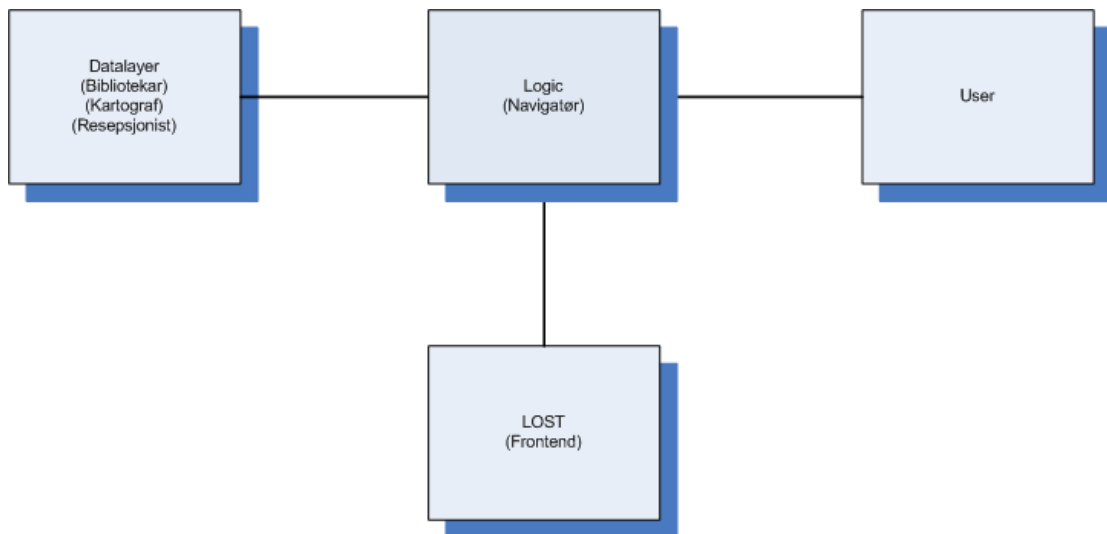


Figur 45: Teknisk løsning for prototypen av LBS systemet

I prototypen ble det kun implementert en klientdel av systemet for enkelthets skyld og for å spare tid og ressurser. Dermed hviler alt ansvaret på denne klienten og klientenhetene må ta seg av prosesseringen. Det vil i alle tilfeller være aktuelt å legge en viss mengde prosessering hos klienten for å unngå overbelastning av serversiden.

Klienten vil her gjøre oppslag direkte mot databasen for å finne ut hvor den befinner seg. Deretter vil den kunne slå opp den aktuelle informasjonen i databasen og presentere denne for brukeren.

## 7.5 Systemarkitektur



Figur 46: Her ser vi hvordan systemstrukturen ble implementert i prototypen

I prototypen ble systemstrukturen implementert på en litt annerledes måte enn det opprinnelige designet. Dette beror til dels på tid tilgjengelig til å gjennomføre implementering og til dels på hva som var praktisk å gjennomføre.

### **Logic (Navigator, Koordinator)**

Som vi ser av diagrammet ble det implementert en logic komponent som fungerer som en koordinator for de andre komponentene og er den komponenten som tar seg av posisjoneringen. Dette tilsvarer altså navigator-komponenten fra det ideelle designet. Den står også derfor sentralt i forhold til de andre komponentene.

### **Datalayer (Bibliotekar, Kartograf, Resepsjonist)**

Datalayer komponenten har tilgang til informasjonsbasen og fungerer dermed som bibliotekar-komponenten fra det opprinnelige designet. Det ble ikke implementert en egen kartograf-komponent da dennes oppgaver ville vært for kompliserte og tidkrevende for denne oppgaven. Geografiske Informasjonssystemer (GIS) er en kilde til hvor man kan finne løsninger for en slik komponent. I stedet fikk nevnte datalayer komponent dette ansvaret i form av tilgang til enkle statiske kart som kan hjelpe en bruker til å få oversikt over området og sin posisjon i forhold til dette. Likeledes ligger også resepsjonist-komponentens ansvar hos datalayer-komponenten da dette for øyeblikket kun består av lenker til eventuelle webtjenester lokasjonene måtte ha. Bestilling av abonnement på eksterne infotjenester er ikke implementert.

### **User (brukeren)**

Det ble opprettet et user-objekt som representerer brukeren som til enhver tid er innlogget i systemet slik at brukerens profil alltid er tilgjengelig for de andre

komponentene.

### **LOST (Guide)**

Med utgangspunkt i hvordan utviklingen av applikasjoner foregår i Visual Studio ble det laget en frontend-komponent som representerer brukergrensesnittet og tar seg av kommunikasjonen mellom systemet og brukeren. Denne formaterer og presenterer informasjonen for brukeren, og videreformidler forespørsler til logic komponenten som også er koordinator. Denne tilsvare Frontend-komponenten fra det opprinnelige designet.

Budbringerkomponenten er ikke implementert da det ikke har vært fokus på gjennomføring av slik direkte kommunikasjon i denne oppgaven. En mulig løsning som kan være aktuell i et slikt tilfelle er å bruke en åpen ”instant messaging” protokoll slik det ble gjort ved University of Twente [14] i tillegg til å overføre slik informasjon mellom entiteter i det sentrale informasjonslageret slik det ble gjort i CyberGuide [1].

## 7.6 Posisjonering

For å kunne lokalisere en bruker på campus trengs teknologi som kan finne og ekstrahere posisjonen til den trådløse enheten og dermed også brukeren, slik at en tjeneste kan nyttiggjøre seg av den. Forskjellige metoder for lokalisering ble presentert tidligere i oppgaven.

### 7.6.1 WiFi Fingerprinting

Her presenteres en implementasjon av en løsning som egner seg bra for formålet med en lokasjonsbasert tjeneste for studenter, kanskje spesielt med tanke på begrensninger som pris og tilgjengelighet, og hva som er både mulig og plausibelt med dagens teknologi. En annen fordel er også at det muliggjør utvikling av lokasjonsbaserte tjenester nesten ”rett ut av esken”.

#### Nibble

Nibble [5] er et lokaliseringssystem som benytter trådløs LAN (WiFi) ”fingerprinting” for å spore lokasjonen til en bruker. Nibble er ekstrahert fra det større systemet Muse [22] hvor det inngikk som en lokaliseringskomponent. Muse beskrives som en arkitektur for smarte sensor-omgivelser, hvor man forsøker å utnytte den økende graden av teknologi rundt oss i hverdagen til å samle data og informasjon om omgivelsene og brukte dette til vår fordel, f.eks. ved å spore hvor brukere befinner seg ved hjelp av trådløse aksesspunkt og nettverkskort. Nibble henter ut signalmålinger fra nettverkskortet angående signalstyrke og signalforstyrrelser for aksesspunktene og sammenligner dette mot data allerede lagret i systemet. Gjennom denne sammenligningen bestemmer systemet om man befinner seg på noen av de lagrede lokasjonene. Nibble benyttet funksjoner i Orinoco drivere for å hente ut målingene, og var derfor låst til å benytte denne typen kort og drivere på klientene.

#### NetStumbler

For å kunne måle signaler fra trådløse aksesspunkt trenger man å kunne hente ut informasjon som plukkes opp av det trådløse nettverkskortet på klientenheten. I de fleste tilfeller krever dette at man får tak i data fra nettverkskortets drivere som vet hvordan dette skal hentes ut, og er forskjellig fra leverandør til leverandør av slike kort. NetStumbler [21] er en applikasjon laget for å kunne overvåke trådløse nettverk, og har den fordelen at den støtter et stort antall forskjellige typer trådløse nettverkskort. Det tillater også å kjøre egendefinerte script for å håndtere WLAN informasjonen slik man selv ønsker. Et forholdsvis enkelt VBScript henter ut de dataene vi er interessert i, legger det i en database og holder de kontinuerlig oppdatert.

#### Fumble

Fumble ble resultatet av et samarbeid med Geir Marius Gjøl [10] hvor det ble benyttet samme prinsipp som lå til grunn for Nibble med WiFi Fingerprinting. NetStumble ble benyttet for å hente ut målinger fra det trådløse nettverkskortet uten å måtte programmere direkte mot spesifikke trådløskort og for å slippe å

utvikle egen kompatibilitet for noe slikt på dette tidspunktet da det kan være svært omfattende. Det ble skrevet et VBScript som kjøres sammen med NetStumbler som kontinuerlig oppdaterer målinger fra det trådløse nettverksskottet i en databasefil. Scriptet er å finne i appendix. Dette er grunnlaget for å lage fingerprint av signalet for forskjellige lokasjoner, og er det som brukes til å sammenligne stedet man befinner seg på mot tidligere registrerte fingerprint.

Teknikken bak metoden som er implementert for WiFi Fingerprinting er ganske enkel og rett fram, men fungerer svært bra og med god presisjon. I følge tester gjort av Geir Marius Gjul [10] har det en innendørs lokasjonspresisjon på ned til 1 meter med 80% riktig posisjonering og hele 98% riktig posisjonering innenfor en radius av 3 meter. For å lage fingerprint av lokasjon samles dataene fra nettverksskottet inn over tre intervaller, og den beste avlesningen blir tatt vare på som et fingerprint. I tillegg kan flere fingerprint lages for hver lokasjon for å øke presisjonen. Når systemet så sammenligner sin nåværende lokasjon med databasen med lokasjoner vil systemet matche mot beste treff av alle fingeravtrykkene, også for samme lokasjon, og returnere beste treff.

Fingerprint XX		Client reading		
MAC	signal	MAC	signal	
00146A3D9D40	-81	00146A3D9D40	-85	= 4
00146A3D98B0	-86	00146A3D98B0	-79	= 7
00146A3D9CD0	-78	00146A3D9CD0	-75	= 3
020E350000B3	-61	020E350000B3	-61	= 0
00146A3D9D60	-77	00146A3D9D60	-78	= 1
00146A3D9D90	-83	00146A3D9D90	-80	= 3
00146A3D9DC0	-81	00146A3D9DC0	-83	= 2
000E8382A370	-87	000E8382A370	-87	= 0
00146A171180	-71			

Signalstyrkedifferanse = 20
Straffepoeng for ulike aksesspunkt = 10
<b>Total score = 30</b>

Figur 47: Eksempel på utrekning av score for Fingerprint på lokasjon i Fumble

Denne beregningen, illustrert over, gjøres ved å gi poeng for likhet med signalet til hvert fingerprint. Minste poengsum betyr beste treff, derfor legges det til poeng for uoverensstemmelser mellom målingen og et gitt fingerprint. Når utregningen starter sammenlignes hver enkelt måling av et trådløst aksesspunkt hos klienten med tilsvarende aksesspunkt i fingeravtrykket ved å matche MAC adressen. Signalstyrken som blir målt hos klienten blir sammenliknet med signalstyrken for aktuelle aksesspunkt i fingeravtrykket (noise og signal-to-noise ratio kan også brukes, men har i tidligere studier [4] ikke gitt nevneverdig utslag). Scoren for likhet mellom signalstyrken på målingene av aksesspunktet blir satt til den samme som differansen mellom de to signalstyrkene, så jo mindre differanse det er dess bedre, altså lavere, blir scoren. Hvis aksesspunktet ikke finnes i fingeravtrykket



gis det ”straffepoeng” for denne ulikheten. Likeledes for omvendt tilfelle. Når score for alle lokasjoner er utregnet antas det at brukeren befinner seg på den lokasjonen med lavest score, men kun dersom score kommer under en gitt verdigrænse. Denne verdien har vi satt til 50 som standard, men implementert real-time justering av denne for å kunne øke eller avta lokaliseringsfølsomheten for systemet dersom det skulle være nødvendig, for eksempel ved bruk av dårligere nettverkskort.

### 7.6.2 GPS

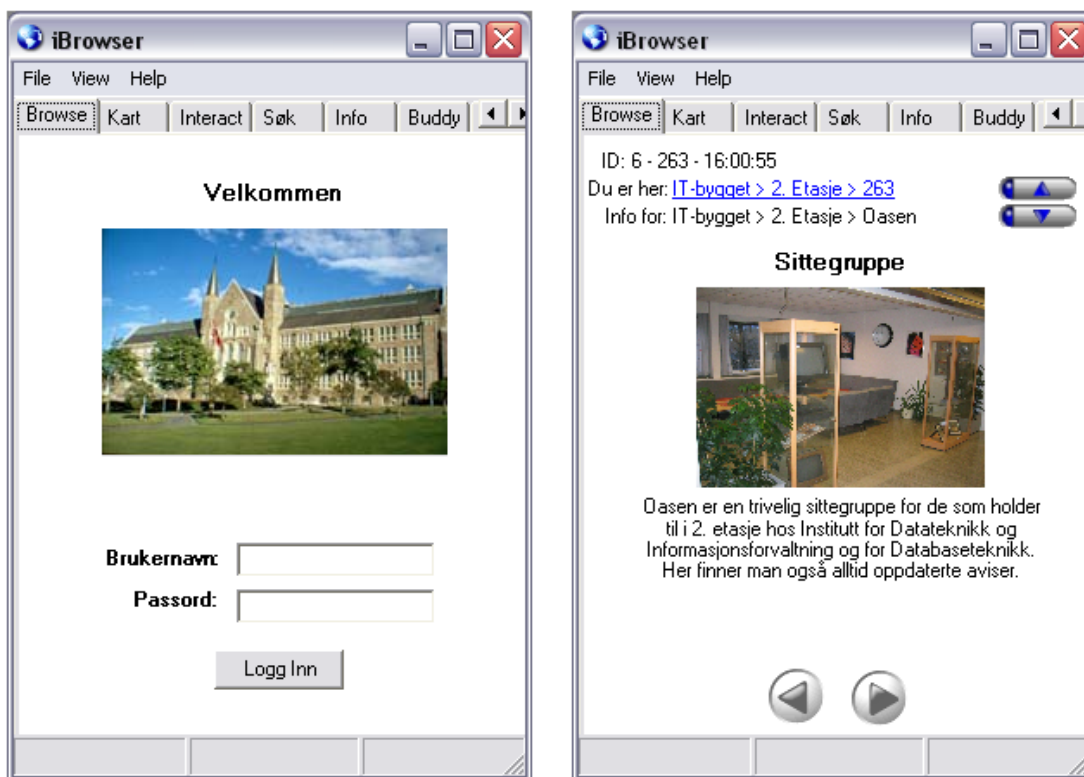
GPS posisjonering er ikke implementert i denne prototypen da det ikke var tilgang på GPS-utstyr hverken integrert i klient eller eksternt tilkoblet, noe som stort sett begrenser posisjonering til innendørs bruk, bortsett fra enkelte tilfeller hvor det trådløse nettets rekkevidde strekker seg ut fra bygningene de er plassert i. Det er dog tatt hensyn til mulighet til bruk av GPS koordinater i lokasjonsmodellen i henhold til spesifikasjonene i ADL GCS [9].

## 7.7 Prototype klientapplikasjon - LOST

Klientapplikasjonen er utviklet i Visual Studio .NET 2003 med VisualBasic.NET som utviklingsspråk. Dette ble brukt fordi det er raskt og enkelt å utvikle i og genererer gode brukergrensesnitt. I tillegg er det gratis tilgjengelig på NTNU sine studentmaskiner, og tidligere erfaringer med dette verktøyet og språket er gode. Det er også en fordel at det er enkelt å kommunisere direkte med databaser fra koden. Prototypen er laget for å kjøre på en laptop pc med WiFi da det ikke var noen PDA eller andre lignende enheter tilgjengelig. Applikasjonen har ganske enkelt blitt døpt LOST (LOcation SysTem).

Prototypen implementerer WiFi Fingerprinting teknikken fra Fumble og benytter en implementering av CULBIS dataformatet som beskrevet tidligere. Databasen i prototypen består av en Microsoft Access database. Dette ble benyttet for å spare tid og ressurser under utvikling samt at det er god støtte for dette i .NET rammeverket som ble benyttet. Et mer fullstendig system ville implementert denne videre i SQL-Server, Oracle, DB2 eller lignende som støtter flere brukere.

Klientapplikasjonen ble utviklet i to utgaver, en for å administrere informasjonen i systemet, og en for studenter som skal benytte seg av systemet.



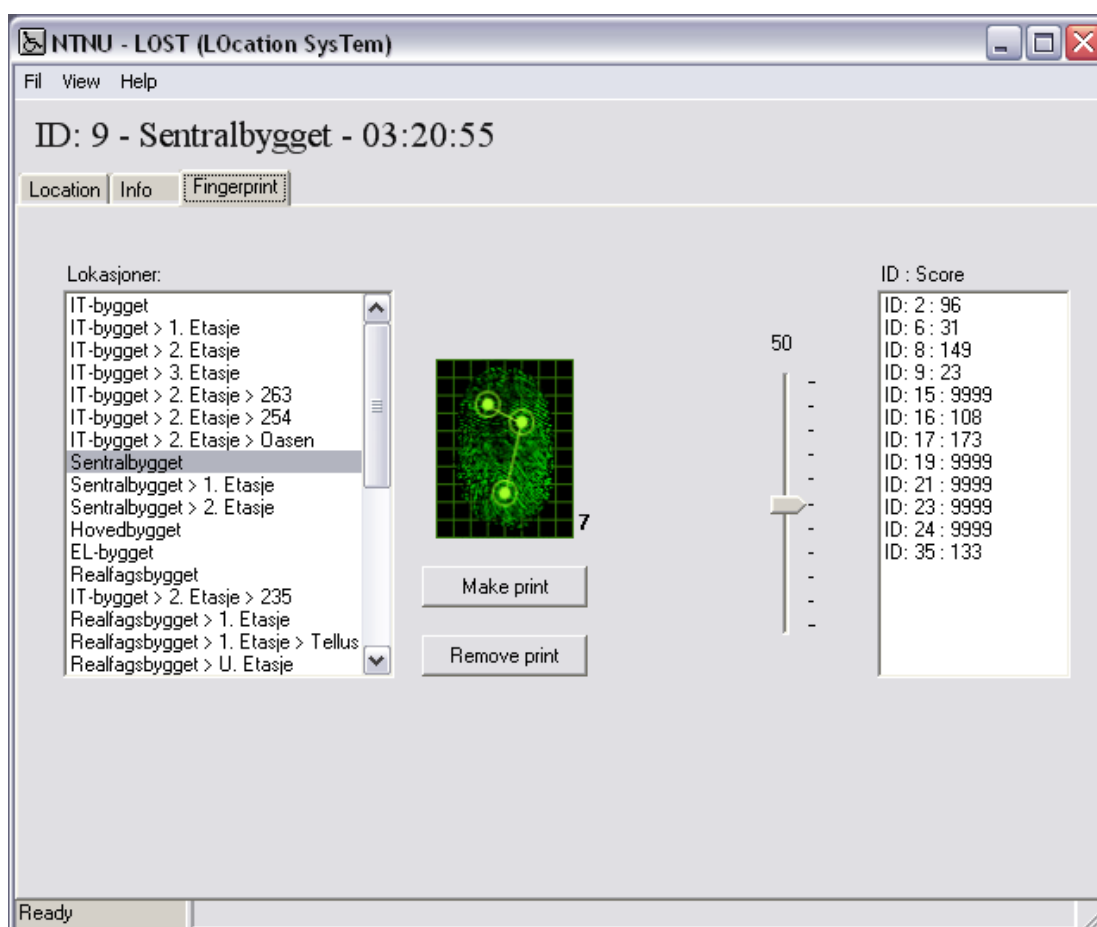
Figur 48: Skjerm bilde av innloggingsskjermen og infoskjerm for brukerutgaven av prototypen

### 7.7.1 Bruererklient

Denne utgaven er ment for den gjengse bruker, altså studenten. Den lar brukeren logge seg inn i systemet og starter så sporingen. Informasjon vil bli presentert for brukeren basert på hvor han befinner seg, og han får også tilgang til andre tjenester for lokasjonen. For å ikke låse brukeren til kun de ressursene som er tilgjengelig for akkurat hvor han er er det mulighet for å navigere rundt i lokasjonene og informasjonen uten å fysisk måtte forflytte seg dit. Systemet vil hele tiden holde rede på hvor brukeren befinner seg og gjøre det enkelt å navigere tilbake til opprinnelseslokasjonen. Se forøvrig brukermanual i appendix [A.2.2](#).

### 7.7.2 Administrator

Denne utgaven lar en administrator skape lokasjoner i systemet og legge inn informasjon som brukerne i sin tur vil få vist når de benytter systemet. Denne brukes også til å opprette fingerprints ut fra WiFi signalene for hver enkelt lokasjon, og disse kan forbedres ved å lage flere fingerprints. Se forøvrig brukermanual i appendix [A.2.1](#).



Figur 49: Skjermbilde av administratordelen til prototypen

### 7.7.3 Tjenester

Tjenestene som er implementert benytter funksjonene som beskrevet i kapittel 7.3. Tjenestene benytter med andre ord en eller flere funksjoner. Merk at tjenestene i stor grad er integrert i hverandre slik at en tjeneste kan benytte deler av en annen for å tilby sin tjeneste. Dette er bare en kort gjennomgang av tjenestene. For mer fylldig gjennomgang av funksjonalitet, se appendix A.

#### Spring/info

Springstjenesten er den som holder rede på hvor brukeren befinner seg og gir informasjon om lokasjonen brukeren befinner seg på, og fasiliterer mange av de øvrige tjenestene gjennom springen. Den kan også vise info om steder som for eksempel er søkt frem. Denne tjenesten benytter funksjonene:

- Fortelle brukeren hvor han er
- Vise generell info om brukerens lokasjon
- Navigere fram til steder og info (via GUI)
- Hente fram utvidet informasjon om steder

Denne tjenesten sier også i fra hvis brukeren har elementer i huskeliste for lokasjonen, eller om det finnes event informasjon. Se aktuelle tjenester ”Huskeliste” og ”Event”.

#### Kart

Karttjenesten viser brukeren kart for lokasjonen han befinner seg på, eller eventuelt for lokasjon han har søkt opp. Tjenesten benytter følgende funksjoner:

- Vise brukeren kart over område
- Vise brukeren hvor han er på kart

#### Søk

Søketjenesten lar brukeren søke etter lokasjoner gjennom type lokasjon, navn på lokasjon og beskrivelse av lokasjon. Den benytter følgende funksjoner:

- Søke etter steder (ved navn)
- Søke etter informasjon om steder (ved beskrivelser)
- Søke opp bestemte typer steder
- Vise brukeren kart over oppsøkt område

Tjenesten gir tilgang til ”Spring/info” tjenesten for oppsøkte lokasjon.

#### Annotasjon

Annotasjonstjenesten lar brukerne legge inn annotasjoner/notater om den valgte lokasjonen, som kan hentes frem igjen ved en senere anledning eller når brukeren befinner seg på lokasjonen. Den benytter følgende funksjoner:

- Legge inn egen informasjon om steder

- Vise brukerens egen info om lokasjon

### Huskeliste

Denne tjenesten lar en bruker legge inn elementer i en huskeliste for hver enkelt lokasjon. Denne listen kan hentes frem igjen ved en senere anledning eller når brukeren befinner seg på samme lokasjon. Brukeren blir minnet på elementer i huskelisten for en lokasjon når sporingstjenesten sporer brukeren til aktuelle lokasjon eller når brukeren navigerer til lokasjon gjennom søk eller andre metoder.

- Lage huskelister for steder
- Bli påmint ting fra huskeliste på stedet

### Webtjeneste

Denne tjenesten gir brukeren enkel tilgang til eksisterende webtjenester en lokasjon kanskje har. For eksempel reservering av rom eller lignende.

- Interaktive tjenester for hver enkelt lokasjon (reservering o.l.)

### Buddy

Buddytjenesten lar brukeren spore personer i kontaktlisten for å finne ut hvor de befinner seg. Tjenesten simulerer hvordan dette kan fungere, men implementerer ingen virkelige funksjoner. Lokasjon for sporing av personer i kontaktlisten gjøres ved å velge en tilfeldig lokasjon for å simulere virkelig funksjonalitet. Hvis dette var blitt fullstendig implementert ville den benyttet følgende funksjoner:

- Finne (spore) andre medstudenter
- Finne (spore) andre deltakere (til event)

### Event

Event tjenesten gjør brukeren oppmerksom på at noe foregår eller kommer til å foregå på aktuelle lokasjon og gir informasjon om dette.

- Gi brukeren dynamisk oppdatert informasjon om lokasjonen (bl.a. event)

### Settings

Dette er i og for seg ikke en tjeneste, men har en funksjon som lar brukeren justere følsomhet for lokasjonsutslag. Denne er lik funksjonen som ble forespeilet i informasjonsanalysen, men har ingen tilknytning til lokasjonmodellen og de data som finnes der. Den er basert på parametre i forbindelse med fingerprinting teknikken som har en verdigrænse for utslag på gjenkjent lokasjon. Det er denne som kan justeres til å være strengere eller mildere i sin vurdering av gjenkjent lokasjon.

## 7.8 Info i systemet

Prototypeimplementasjonen ble fylt med et sett med lokasjoner, fingerprints og informasjon om lokasjonene. Denne delen vil si litt om informasjonen som finnes i dette systemet.

### 7.8.1 Testlokasjoner

Nesten samtlige bygninger som blir benyttet av studenter på Gløshaugen er lagt inn i systemet med en minimumsbeskrivelse for å ha noe informasjon å kunne vise for disse bygningene. Hovedsaklig er det lagt inn lokasjoner i IT-bygget, Sentralbygget og Realfagbygget. Utendørs lokasjoner har blitt utelatt da disse er mye vanskeligere å posisjonere uten GPS eller annen mer egnet utendørs posisjoneringssystem.

Følgende liste forteller hvilke lokasjoner som eksisterer i systemet. Lokasjoner merket med asterisk (\*) har også fingerprints for lokalisering knyttet til seg.

```
IT-bygget *
IT-bygget > 1. Etasje
IT-bygget > 2. Etasje
IT-bygget > 1. Etasje > F1
IT-bygget > 2. Etasje > 235 *
IT-bygget > 2. Etasje > 254 *
IT-bygget > 2. Etasje > 263 *
IT-bygget > 2. Etasje > Oasen *
Realfagbygget *
Realfagbygget > 1. Etasje
Realfagbygget > U. Etasje
Realfagbygget > 1. Etasje > Realfagsbiblioteket *
Realfagbygget > 1. Etasje > Tellus *
Realfagbygget > U. Etasje > Café Sito *
Sentralbygget *
Sentralbygget > 1. Etasje
Sentralbygget > 1. Etasje > Hangaren *
Sentralbygget > 1. Etasje > Café Sito
Sentralbygget > 1. Etasje > Tapir Storkiosk *
Sentralbygget > 1. Etasje > Studentservice *
Sentralbygget > 1. Etasje > SiT Studentinfo *
EL-bygget *
EL-bygget > 1. Etasje
EL-bygget > 1. Etasje > Terra *
Kjelhuset *
Kjemiblokkene
Geologi
```

Gruvedrift  
Materialteknisk  
Verkstedteknisk  
Metallurgi  
Produktdesign \*



Figur 50: Her ser vi et eksempel på plassering av fingerprint for posisjonering i IT-bygget, 2.etasje.

### 7.8.2 Testinfo

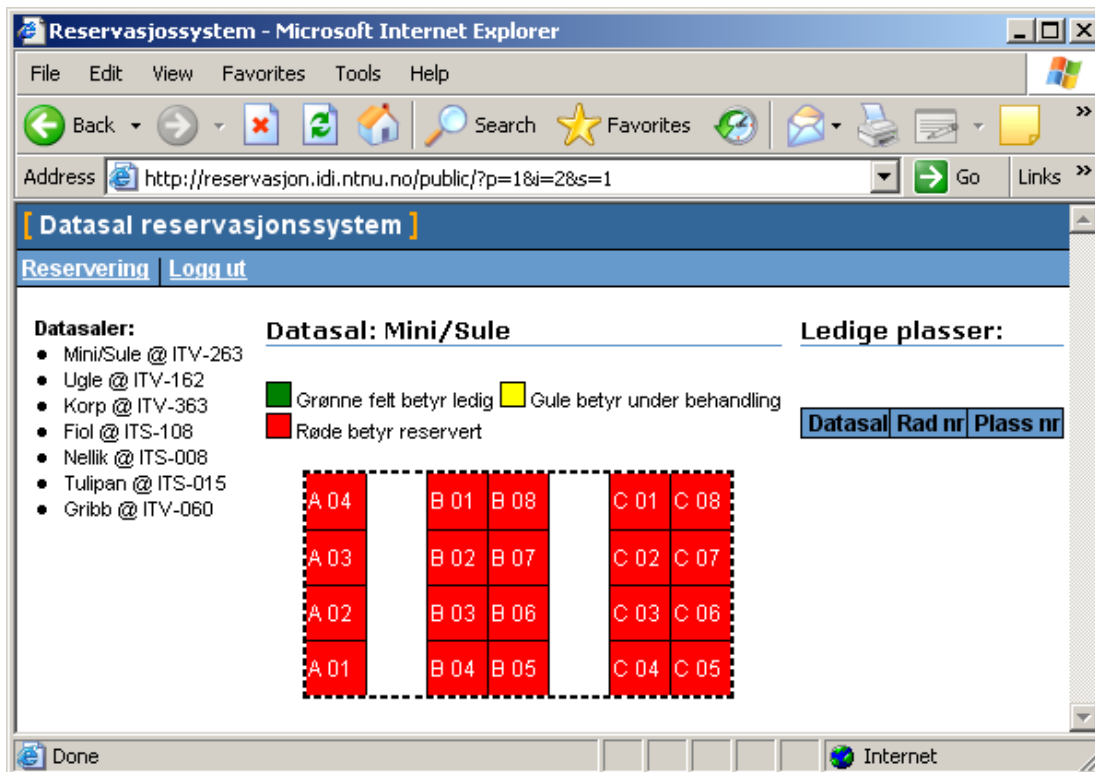
Det er lagt inn beskrivende info om nær sagt samtlige lokasjoner i systemet for å kunne teste og illustrere hvordan denne informasjonen kan fremstilles. Unntaket er for de fleste lokasjonene av typen "etasje" som fungerte mer eller mindre som holdere av lokasjoner. For å få systemet opp å kjøre med denne lokasjonsmodellen måtte et sett med nødvendig data legges inn. Dette gjelder de skjema-baserte tabellene som skal kunne tilpasses til applikasjonsområdet. Alle lokasjoner er selvfølgelig klassifisert etter type og funksjon. Det samme gjelder for brukerne som er lagt inn systemet samt andre klassifiseringer og skjemaer. Se appendix D for oversikt over predefinerte skjema.

Hver lokasjon er gitt en enkel beskrivelse basert på hva slags lokasjon det er. Der det er behov for litt mer utdypende informasjon er det lagt til mer enn ett kapittel. De fleste lokasjoner har også et bilde som representerer lokasjonen. Bygningene fikk en beskrivelse av hvem som har tilhørighet der og hva man kan gjøre der. Kantiner har info som opplyser om ting som dagens middag, menyer og priser og så videre, og enkeltrom har info som beskriver hva de brukes til og hva man kan gjøre der. Lokasjoner med egne websider eller webtjenester ble lagt inn med lenker, men det var få av disse. Rom 263 i IT-bygget fikk for eksempel lagt inn webtjenesten for å reservere arbeidsplass for det aktuelle rommet. Printerrommet 254 fikk lagt til webtjeneste i form av link til tjenesteside for fargeprinterne på rommet.

Det er lagt inn statiske kart for godt over halvparten av lokasjonene for å illustrere kartfunksjonen. Disse varierer i granularitet fra oversikt over hele Gløshaugen campus ned til hver enkelt bygning med angivelse av lokasjon inne i bygningen. Det er lagt inn event informasjon på enkelte bygninger for å teste og illustrere



Figur 51: Eksempel på info om lokasjonene.



Figur 52: Eksempel på webtjeneste for en lokasjon.



event funksjonen hvor man blir opplyst om hendelser eller arrangement som skjer ved lokasjonen.

Testpersonene fikk selv muligheten til å legge inn annotasjoner og huskelister for å teste funksjonen av disse. Det er også lagt inn et par huskelister og annotasjoner på forhånd for hver testperson for å hjelpe dem i gang med tjenestene.

## 7.9 Oppsummering

På grunnlag av State-of-the-art og øvrig litteratur, gazetteers, ADL Gazetteer Content Standard og gjennomført analyse av informasjon er det foreslått en lokasjonsmodell/dataformat for å støtte en rekke lokasjonbaserte funksjoner og tjenester. Lokasjonsmodellen er modellert med utgangspunkt i Gazetteer strukturen. Det ble her foretatt et utvalg av elementer som skulle inngå. Dette utvalget var noe stort, men for å imøtekomme flest mulig funksjoner var dette nødvendig og burde også kanskje ha vært enda større.

Lokasjonsmodellen er designet for å støtte samtlige funksjoner fra kapittel 6, men bare et utvalg av disse er implementert i prototypeapplikasjonen. Grunnen til dette er at det hadde blitt for omfattende for denne oppgaven å implementere og teste samtlige. Utvalget er gjort på grunnlag av hva som er regnet som kjernefunksjonalitet i systemet, for å få representert alle tjenesteparadigmene og for å spare tid i enkelte tilfeller.

Den tekniske løsningen på prototypen ble noe annerledes enn beskrevet i kapittel 6. Dette var et valg som ble gjort for å korte ned implementeringstiden. I all hovedsak ble serverdelen av systemet slått sammen med klientdelen slik at klienten bærer hele ansvaret, og er dermed koblet rett mot databasen. Dette var ikke noe problem da testenheten, en laptop pc, hadde nok kapasitet til å håndtere dette. Med all sannsynlighet ville også en moderne PDA kunne gjort det samme. Det er derfor også god grunn til å spekulere i å legge mer av ansvaret på klientene for å spare serversiden for unødig last.

Systemarkitekturen består av en komponent som tar seg av posisjoneringen og koordinering mellom de andre komponentene, en komponent som har kontroll på datalageret, en komponent som kommuniserer med brukeren samt et objekt for å representere brukeren i systemet. Ansvaret for øvrige komponenter beskrevet i kapittel 6 ble fordelt mellom disse komponentene der dette passet seg best, men med ganske begrenset funksjonalitet i forhold til hva som var ønskelig, eller rett og slett ikke implementert. Dette ble gjort fordi hver komponent kan tilsvare et eget system i seg selv og det ville blitt for komplisert og tidkrevende for denne oppgaven å implementere alle.

Valg av posisjoneringsteknologi ble gjort på grunnlag av hva som egner seg til innendørs bruk og fordi oppgaven ikke har budsjett til kostbar teknologi. GPS er derfor ikke benyttet da dette ikke var tilgjengelig. WiFi Fingerprinting, derimot, kunne implementeres ut i fra enkle prinsipper, hadde allerede god tilgang til trådløst nettverk og var i så måte en kostnadsfri teknologi. Den fungerer også svært bra og tåler sammenligning med flere mer avanserte og mer kostbare teknologier

til sitt bruk.

Selve applikasjonen ble laget i to varianter, en for å administrere informasjonen i systemet og en for sluttbrukere. Dette ble gjort for å adskille disse to delene slik at sluttbrukerne ikke trenger å forholde seg til hvordan systemet fungerer, bare at det faktisk fungerer. Det er også behov for de som vedlikeholder systemet å ha enkle måter å skape, publisere og vedlikeholde innholdet.

## 8 Testing og evaluering

Testingen som er gjennomført har fokusert på å validere funksjonaliteten til et lokasjonsbasert informasjonssystem for studenter ved store campus som NTNU Gløshaugen og hvordan slik funksjonalitet og informasjon bør presenteres for brukerne. I tillegg er det også sett noe på studentenes innstilling til å bruke et slikt system både i og utenfor en skolesetting.

Testingen ble gjennomført ved at en testgruppe fikk en rask gjennomgang av hvordan de skulle benytte systemet (brukerklienten), og deretter fikk teste applikasjonen ved NTNU Gløshaugen campus slik de selv følte for.

Testgruppen gjennomførte etter endt testing en evaluering gjennom et spørreskjema som hadde nevnte fokus. I tillegg ble det gjennomført uformelt intervju av hver enkelt testperson hvor systemets funksjonalitet og presentasjon ble diskutert, og hvorvidt brukerne kunne tenke seg å benytte et slikt system i forskjellige situasjoner. Testpersonene var fri til å komme med egne innspill.

### 8.1 Testgruppen

Testgruppen besto av totalt 6 personer. Tre av testpersonene var informatikkstudenter, en var datateknikkstudent, en var kybernetikkstudent og en studerte humanistiske fag. Tre av testpersonene var seniorstudenter, en var tidligere student og to var nye studenter. To av studentene hadde også benyttet lignende tjenester før i en eller annen form. Testpersonene ble valgt ut tilfeldig blant bekjente som hadde tid og anledning til å delta.

### 8.2 Testoppsett

Testoppsettet som testgruppen benyttet var så åpent som mulig for at de skulle få et best mulig inntrykk av hvordan et ferdig system skulle kunne bli benyttet. De fikk med andre ord full tilgang til klienten slik den er beskrevet i brukermanualen i kapittel [A.2.2](#), med alle funksjoner og tjenester slik de er beskrevet. Testpersonene fikk en rask gjennomgang av hvordan klienten fungerer og fikk fritt teste funksjonaliteten. De hadde hele tiden en person som var kyndig på systemet tilgjengelig hvis de hadde spørsmål. Testpersonene fikk selv velge når de mente de hadde testet systemet tilstrekkelig. Hver gjennomføring varte i mellom 30 minutter til 1 time. Testingen ble gjennomført i henholdsvis IT-bygget, Sentralbygget og Realfagbygget hvor det er lagt til en del testlokasjoner rundt omkring. Noen testpersoner valgte å bevege seg mellom to av bygningene mens andre holdt seg innad i en av bygningene.

### 8.3 Testdata

Testdata som ble benyttet var det sammen som beskrevet for prototypen i kapittel 7.8. Det består av en rekke lokasjoner i forskjellige bygninger med lagrede fingerprints for posisjonering, kart, bilder, beskrivelser og øvrige informasjonselementer.

### 8.4 Spørreskjema

Spørreskjemaet som ble benyttet hadde totalt 30 spørsmål som etterhvert ble redusert til 29 da ett av spørsmålene, spørsmål 2, ble funnet overflødig siden det ble bestemt at testpersonene skulle få en rask innføring i bruk av systemet. Skjemaet stiller en del ja/nei spørsmål og ber i en del tilfeller testpersonen begrunne svarene gjennom kommentarer og forklaringer. For fullstendig spørreskjema, se appendix F.

Spørreskjema	
1. Har du benyttet en lignende tjeneste eller system før?	Ja / nei
2. Hvilke tjenester (funksjoner) benyttet du?	
3. Hvilke tjenester (funksjoner) syntes du var nyttige?	Søk   annotering   webtjeneste   huskeliste   event   buddy   kart
4. Har du noen kommentarer til sporsingsfunksjonen?	
5. Har du noen kommentarer til søk funksjonen?	
6. Har du noen kommentarer til buddy funksjonen?	
7. Har du noen kommentarer til huske liste funksjonen?	
8. Har du noen kommentarer til annoteringsfunksjonen?	
9. Har du noen kommentarer til event funksjonen?	
10. Har du noen kommentarer til kart funksjonen?	
11. Har du noen kommentarer til webtjeneste funksjonen?	
12. Var det noen av funksjonene du gjerne ville skulle fungert annerledes?	Ja / nei
a. Hvis ja, hvilken/hvilke, og hva burde vært annerledes?	
13. Var det tjenester (funksjoner) du savnet som du kunne tenkt deg?	Ja / nei
a. Hvis ja, hvilke?	
14. Var det informasjon du savnet som du synes burde være tilgjengelig?	Ja / nei
a. Hvis ja, hva?	
15. Var det ting som var vanskelig å forstå?	Ja / nei
a. Hvis ja, hva?	
b. Hvis ja, hvordan kunne dette blitt gjort mer forståelig?	
16. Var funksjonene enkle å bruke?	Ja / nei
a. Hvis nei, hva var vanskelig å bruke, og hvorfor? (Søk, annotering, huskeliste, webtjeneste, , event, buddy, kart)	
17. Var det greit å navigere rundt i informasjonen?	Ja / nei
a. Hvis nei, hvordan kunne det gjøres bedre?	
18. Ville du lært deg å kjenne campus raskere med en slik tjeneste?	Ja / nei
19. Er dette noe du ville brukt som ny student?	Ja / nei

Figur 53: Spørreskjema som ble benyttet for testing, side 1 og 2.

### 8.5 Intervju

Intervjuet med testpersonene ble utført rett etter spørreskjemaet hadde blitt besvart, og tok form som en uformell diskusjon om systemet og testingen av den, samt diskusjon i rundt svarene som nettopp hadde blitt gitt i spørreskjemaet. Dette fanget godt opp meninger og synspunkter som var vanskelig å få frem på et papirbasert spørreskjema til tross for at en del av svarene var i fritekst form.

## 8.6 Resultater

Sett bort i fra selve sporingsfunksjonen som pusher info var de mest populære tjenestene event, kart og søk som ble listet som nyttig av henholdsvis seks, seks og fem av de totalt seks testpersonene. Webtjeneste og huskeliste ble listet som nyttige av fire, og til slutt annotering med tre, men med visse forbehold om funksjonalitet for sistnevnte.

Tjenesten som pusher informasjon om lokasjonene ble av samtlige testpersoner sett på som svært nyttig til bruk ved store steder eller områder hvor man er dårlig eller ikke kjent. Den samme funksjonaliteten ble sett på som mindre nyttig hvis man allerede var godt kjent. Det ble også kommentert at et slik system ville være enda mer aktuelt for enda større områder som for eksempel en hel by.

Nesten samtlige, fem av seks, mente det måtte finnes en søk-funksjon i et slikt system. To av testpersonene etterlyste mulighet til å finne nærmeste sted av en bestemt type, for eksempel nærmeste datasal, printer eller toalett. En person ville gjerne kunne søke etter alternative navn på steder, siden enkelte steder har flere navn eller kallenavn, og det kan være vanskelig å vite hva som er riktig. Det ble også etterlyst flere mulige måter å søke på, så som søk på nøkkelord, finne utstyr, søk etter event-informasjon og lignende. En person ønsket også å kunne søke etter for eksempel et fag ved navn eller kode, og så få opp som resultat hvor forelesning for dette faget foregikk. Dette kunne også gjerne strekke seg til flere typer tidsbestemte hendelser.

Fire av de seks testpersonene syntes buddy-tjenesten var nyttig, men det ble kommentert at den gjerne var mer en morsom funksjon man ønsket å ha heller enn en ”matnyttig” funksjon i de fleste tilfeller. To testpersoner stilte seg skeptiske i forhold til personvern når det gjelder at andre vet hvor de befinner seg. Det ble fremmet forslag om å endre eller forbedre presentasjon av tjenesten ved å kunne gruppere kontakter/buddies etter hvor de befinner seg, og også å vise buddies på aktuelle områdekart.

Huskelisten ble sett på som en kjekk og nyttig funksjon som fire av testpersonene ville benyttet. To av testpersonene bemerket at de gjerne ville sett tid inkludert i huskelisten i en eller annen form, slik at man ikke trenger å få påminnelse kun for lokasjonen, men kombinert med et bestemt tidspunkt eller et gitt tidsrom. En av testpersonene var litt skeptisk til å knytte huskeliste til en bestemt lokasjon, men syntes det var nyttig hvis han i tillegg kunne få alle huskelister oppgitt hvis han ønsket det, uten måtte være ved lokasjonen eller browse fram lokasjonen. E-post ble nevnt som et alternativt medium til å samle huskelister.

Tjenesten for annoteringer på lokasjoner var den minst likte tjenesten blant testpersonene, hvor tre syntes den var nyttig men kun med visse forbehold. De som

ikke så på tjenesten som nyttig mente at dette var som en litt annerledes versjon av huskelisten og ville derfor heller benyttet den. Tjenesten ble stort sett bare sett på som nyttig hvis man kunne dele annotasjonene med andre ved for eksempel å sende bestemte annotasjoner til sine venner, eller hvis annotasjonene hadde fungert som et slags lokasjonsbasert forum slik at alle kunne se hva som ble notert. Dermed kunne det fungere som tips mellom studenter og brukere. Alle testpersonene bemerket at de ikke visste hva annotasjon betyr, og ville gjerne heller at tjenesten skulle blitt kalt "gule lapper" da dette gir mye større mening og riktige assosiasjoner for gjennomsnittlige brukere. En av testpersonene som er ny ved campus mente at han ville benytte tjenesten til å merke seg hvilke steder han hadde forelesninger til bestemte tider. Igjen ser vi at tidsaspektet dukker opp.

Event tjenesten var den tjenesten det ble gitt mest positive tilbakemeldinger på. Samtlige seks testpersoner syntes den var nyttig og ville benyttet den, ikke bare som nye studenter, men også som seniorstudenter. Det ble ytret ønske om å kunne få ikke bare informasjon om "arrangement", men også annen aktuell ny og tidssiktig informasjon angående de forskjellige lokasjonene. For eksempel nyheter, oppdateringer og ellers annet nytt som foregår.

Kart ble sett på som nyttig av samtlige testpersoner, og enkelte mente også at det var en nødvendig tjeneste eller funksjon i et lokasjonsbasert system. Halvparten av testpersonene etterlyste større grad av detaljrikhet i kartene for å bedre kunne skille bygninger, rom og andre elementer fra hverandre. De etterlyste også mulighet for å kunne zoome inn og ut på selve kartet. En testperson bemerket at man ikke alltid visste hvilken vei man var orientert i forhold til kartet, og at det kunne vært nyttig med noe som kunne hjelpe på dette. Fem av seks mente man burde kunne trykke på lokasjoner på kartet for å få info om dem. To av testpersonene ønsket også enda større grad av interaktivitet i kartet ved for eksempel å kunne vise navn på lokasjoner når man peker på dem, merke av bestemte typer lokasjoner på hele kartet, og kontinuerlig kunne vise sine egne bevegelser på kartet slik man er vant til fra håndholdte GPS enheter.

Tilgang til webtjenester for hver enkelt lokasjon ble sett på som nyttig av fire av testpersonene. Det ble beskrevet som en "kjekk" funksjon som kunne spare tid i mange tilfeller. Mesteparten av nyttedelen av denne tjenesten ligger dog i selve webtjenesten som en eventuell lokasjon tilbyr.

Testpersonene ble bedt om å komme med forslag til andre tjenester de kunne tenkt seg til et slikt system, som ikke eksisterte i systemet de testet. Det kom tre forslag fra tre forskjellige testpersoner. En foreslo en printerfunksjon som lot brukeren skrive ut dokumenter ved fysisk nærmeste printer. En annen kunne tenkt seg mulighet for å kommunisere med andre brukere gjennom tekstlig "instant messaging" chat. En tredje ville gjerne hatt en veivisertjeneste som kunne vise veien fra sin lokasjon til en annen på kartet. Da testpersonene senere i spørreskjemaet

ble spurt direkte om de kunne tenkt seg en rekke bestemte tjenester svarte fem at de ville hatt en veivisertjeneste. Fem sa at de gjerne ville kunne dele annotasjoner med andre. Fem testpersoner ville også gjerne få beskjed hvis en venn i kontaktlisten var i nærheten. Ytterligere fem ville hatt en chat funksjon, og kunne trykke på kartet for å få mer info om lokasjonene de trykket på. På spørsmål om de kunne tenkt seg å automatisk få opp ting som forelesningsnotater, et fags hjemmeside, faglige meldinger og lignende når de kom inn i et auditorium hvor de hadde forelesning, svarte fem av testpersonene at de gjerne ville det. Det ble dog bemerket at det kanskje var like greit å gjøre det lett tilgjengelig heller enn å la ting "poppe opp". Det ble spurt om det var andre kommunikasjonsformer enn chat testpersonene kunne tenkte seg integrert i et slik system. Følgende ble nevnt: E-post, SMS, kamera/bilde/video, tale og mulighet for filoverføring. Det ble også nevnt at det var ønskelig å kunne benytte hele systemet fra en mobiltelefon hvor man allerede har flere kommunikasjonsformer. Da de ble spurt om deres innstilling til å få levert viktige meldinger gjennom et lokasjonsbasert system slik studentene ved NTNU per i dag gjør via samarbeidssystemet "It:s Learning" svarte en testperson blankt nei, mens fem svarte at det var greit, men med visse forbehold som at da måtte alle bruke det, bare de viktigste meldingene burde komme på denne måten og det måtte ikke virke for påtrengende.

På spørsmål om det var informasjon de savnet i systemet svarte testpersonene at de savnet informasjon om forelesninger og hvor disse ble holdt til hvilke tider. Åpningstider for visse lokasjoner var også noe de kunne tenkt seg, så som når kantiner og kiosker er åpne, og når instituttkspedisjonene og andre servicekontor stenger. Ellers ble det sagt at oppdatert informasjon generelt var noe de kunne tenkt seg mer av.

Samtlige seks testpersoner svarte at de trodde de ville lære seg campus raskere å kjenne ved å benytte slike tjenester, og alle svarte også at de ville benyttet et slikt system som nye studenter hvis det var tilgjengelig. Samtlige trodde også at de fleste nye studenter ville hatt stor nytte av et slikt system, også utover bare det å bli kjent og finne frem. De begrunnet svarene med at det var ideelt for å få svar på det man lurer på når man ikke er kjent, man får informasjon der man trenger det, det samler nyttig informasjon på ett sted, og det kan også hjelpe på å unngå "overflod" av informasjon ved at man blir fortalt bare det man trenger å vite der og da. Et krav måtte likevel være at systemet ble presentert gjennom en håndholdt enhet. Det ble også ytret ønske om å kunne administrere sin egen brukerprofil fra en stasjonær enhet med større skjerm og vanlig tastatur.

Det ble spurt om testpersonene ville fortsette å bruke et slikt system etter at de var blitt seniorstudenter. På det svarte fem at de ville gjort det. De var da mest interessert i å bruke de andre tjenestene enn de som gav statistisk stedsinformasjon og kart, og ville spesielt benyttet tjenester som gav oppdatert informasjon om arrangementer, eventuelle hendelser og nyheter. I tillegg ville de benyttet buddy

tjenesten. To av testpersonene bemerket at systemet burde være enklere å bruke og finne fram i enn det ville vært å lete opp informasjonen andre steder, for eksempel på web, hvis de skulle fortsatt å bruke det.

Metoden med å vise små symboler når bestemte typer informasjon er tilgjengelig ble sett på som mer positivt enn å oversvømme brukeren med all informasjonen på en gang. På denne måten kunne brukeren selv velge om han var interessert i å benytte seg av det som var tilgjengelig. Det ble dog bemerket at slike symboler må være forståelige og gi en mening for brukeren, hvis ikke kan de bli forvirrende.

### 8.6.1 Oppsummering

Vi ser ut i fra denne testingen at det er dynamisk informasjon som er mest ettertraktet av brukerne generelt og av eldre studenter spesielt. Statisk informasjon er kun interessant dersom man ikke er kjent og stedet er stort og uoversiktlig. Sistnevnte vil være tilfellet for nye studenter mens det første vil være tilfellet for seniorstudenter. I prototypen er det mest fokus på statisk informasjon og noe mindre på dynamisk. Systemet har likevel mulighet til å oppdatere informasjonen som ligger. Men det vil være behov for å innarbeide flere dynamiske "elementer" av samme karakter som event/arrangementer for å tilby dynamiske lokasjonsbaserte informasjonstjenester.

Vi kan også trekke ut at det er behov for en mer utførlig kart-tjeneste med interaktive funksjoner. Slike funksjoner kan bygges og utvides basert på datagrunnlaget i dataformatet slik det er foreslått. Når det gjelder informasjon i systemet er det behov for mer metadata om lokasjonene som kan støtte opp om utvidede søkefunksjoner. Nøkkelord for lokasjoner er en delvis løsning på dette, noe det er tatt høyde for i dataformatet. Men enda flere er sannsynligvis nødvendig, så som tilgjengelig utstyr, hvor man finner personer, timeplaner og åpningstider for lokasjoner og tilknytning til fag, kurs og forelesninger.

Og her kommer vi inn på enda et viktig punkt som er identifisert: Tid. Ved flere anledninger er tid oppgitt som et viktig kriterium for brukerne, og det henger ofte nøye sammen med sted. Tidsaspektet ble planlagt og tatt høyde for i dataformatet gjennom flere mekanismer, men ble oversett til fordel for andre elementer når det kom til implementering og testing. Dette bør tas til etterretning og være en mer viktig del av fremtidige arbeider med lokasjonsbaserte tjenester.

Det ble også avdekket et ønske om flere utvidede tjenester basert på lokasjon og sporing, hvorav samtlige kan imøtekommes gjennom et fullstendig system slik det er tilrettelagt for i denne oppgaven. Brukerne uttrykker et behov for å kunne akseptere mest mulig informasjon i oversiktlige former gjennom et felles system eller anvendelsespunkt, og da gjerne et slikt system som baserer seg på organisering etter lokasjon. Vi kan også slå fast at et fullstendig implementert system ville



vært et nyttig og ikke minst velkomment verktøy for både nye studenter såvel som senior-studenter.

## 9 Konklusjon

I løpet av denne avhandlingen har vi sett på en rekke eksisterende og mulige løsninger på lokasjonsbaserte systemer og tjenester, og hvordan disse har blitt gjennomført og benyttet i praksis. Det har blitt identifisert viktige aspekter ved slike systemer og tjenester og den informasjonen disse skal håndtere. Det har blitt sett på metoder og teknologier for å posisjonere brukere av lokasjonsbaserte systemer, og teori i rundt lokasjonsavhengig informasjon har blitt gjennomgått. En lokasjonsmodell/dataformat har blitt foreslått for å representere den nødvendige informasjonen i en hensiktsmessig struktur basert på gazetteers. Videre har et lokasjonsbasert system for studenter ved NTNU Gløshaugen blitt planlagt, designet, og en prototype blitt implementert og testet på grunnlag av dette. Hovedfokus har vært på håndtering og strukturering av informasjon for å støtte lokasjonsbaserte tjenester.

GPS posisjonering er ikke implementert i prototypen da det ikke var tilgang på GPS-utstyr hverken integrert i klient eller eksternt tilkoblet, og det fantes ikke budsjett til å anskaffe slikt utstyr. Dette har begrenset posisjoneringen til innendørs bruk gjennom WiFi Fingerprinting.

Underveis i dette arbeidet har det blitt avdekket et behov for en standard lokasjonsmodell/dataformat for å fremme enklere utvikling, implementering og deling av lokasjons- og kontekstbaserte tjenester og informasjon. De systemene som har blitt studert i denne oppgaven utviklet og implementerte i stor grad sine egne proprietære formater for å gjennomføre systemene.

I tillegg er det identifisert et behov for lokasjonsbaserte systemer til å støtte flere posisjoneringsteknologier da det i dag finnes flere gode teknologier og metoder for dette. De færreste av slike posisjoneringsteknologier og metoder er brukbare til både innendørs og utendørs bruk, i både stor og liten skala og med innhold av høy og lav lokasjonsgranularitet. ADL Gazetteer Content Standard har tatt forbehold om svært mange måter å merke opp et område med koordinater, men har ingen støtte for posisjoneringsteknologier utenom dette.

Granularitet på lokasjonsutslag i seg selv er også en stor utfordring når man skal utvikle lokasjonsbaserte tjenester som benytter seg av informasjonspush. Når man kommer ned på et nivå som bare strekker seg over noen få meter er det viktig at informasjonen som pushes er riktig og nøyaktig, hvis ikke blir tjenesten sett på som forvirrende og upålitelig. Det kan i slike tilfeller være lurt å kunne bedømme denne usikkerheten og tilby et alternativ heller enn å la tjenesten "ta en sjanse" på en usikker levering. Bedømming av slik usikkerhet i lokasjonsbaserte tjenester er også nevnt i litteraturen. Man kan ta høyde for dette ved å legge inn feilmarginer i systemene. For GPS og koordinater kan man gjøre dette ved å benytte feilmargin i meter: Hvis flere lokasjoner er innenfor en viss feilmarginsradius av

posisjoneringen skal alternativ informasjon pushes, eventuelt kan brukeren gis et valg mellom hvilken lokasjon han ønsker informasjon om. For WiFi Fingerprinting kan denne feilmarginen angis i form av poeng: Hvis en lokasjon ikke er mer enn et gitt antall poeng unna posisjonert lokasjon kan brukeren gis alternativ.

Vi har sett at et modulært design kan være en fordel for utvikling av et lokasjonsbasert system da man kan koble til tilleggsmoduler for hver enkelt tjeneste man ønsker å ha støtte for, siden et slikt system vil være tjenesteorientert. Dette vil også være tilfellet for lokasjonsmodellen/dataformatet da data og informasjon kan kobles rett til lokasjoner og være uavhengig av øvrig struktur og data.

Push, pull og interact har blitt identifisert som aktuelle paradigmer for tjenestelevering. Push er svært aktuell for lokasjons- og kontekstbaserte tjenester da tjenestene kan leveres basert på visse kriterier heller enn at brukeren spesifikt ber om det. Det er også viktig å merke seg at brukerne gjerne ønsker å kunne interagere med systemet heller enn å være passive mottakere, og at man derfor bør kunne tilby interaksjonstjenester. Push og pull er også på vei til å møtes og smelte sammen i en felles paradigme gjennom behov og ønsker fra brukerne. Brukerne vet at informasjonen ligger klar (push) og titter på den når de føler for det og forventer at den er der (pull). Det har også kommet frem et ønske fra brukerne om å heller bli gitt beskjed om at aktuell informasjon er tilgjengelig (push), uten å føle seg tvunget til å benytte den. De ønsker heller å kunne gå dypere hvis de selv mener de har behov for det (pull). Dette stiller igjen visse krav til systemets pålitelighet.

Tid er kommet frem som et svært viktig element i forhold til å tilby lokasjonsbaserte tjenester. Akkurat som lokasjoner, er informasjon om lokasjoner spatio-temporære av natur, det vil si at det avhenger av både tid og sted. Og for oss mennesker er både tid og sted to svært viktige aspekter som ofte går hånd i hånd. Dette bør det derfor tas hensyn til i utvikling av lokasjonsbaserte tjenester.

Brukernes krav til nytt og oppdatert innhold og dynamisk informasjon i tjenestene har kommet frem. Dette stiller i sin tur større krav til håndtering av denne typen informasjon, og tilrettelegging for slike tjenester.

For utvikling av lokasjonsmodell/dataformat ble det tatt utgangspunkt i en gazetteer for å strukturere informasjonen, da denne allerede søker å knytte informasjon til steder. Dette gav gode resultater da det viste seg at gazetteers i sin enkleste form støtter opp om flere av funksjonene som et slikt system krever. Det er også enkelt å knytte ny informasjon opp i mot denne strukturen, og det tillater at lokasjoner kan defineres så løst eller fast som man har behov for og knytte dem sammen i logiske hierarki slik vi mennesker ofte ser det for oss. Dette gjør dataene og informasjonen både enkel å forstå og håndtere.

## 10 Videre arbeid

Det som nå kan sees på som viktig for videre arbeid i henhold til denne oppgavens fokus er bedre tilrettelegging for dynamisk informasjon i systemet. Det kan argumenteres at det som nå sees på som statisk informasjon i systemet enkelt kan holdes oppdatert ved for eksempel å gi de forskjellige fakultetene, instituttene og ellers de som hører til de forskjellige stedene ansvar for å oppdatere informasjonen for sine lokasjoner. Men informasjonens natur vil fortsatt være statisk eller semi-statisk. Det er derfor behov for å utvide lokasjonsmodellen/dataformatet til å inkludere elementer som bedre fanger opp dynamisk informasjon og kan tilrettelegge for bedre dynamiske informasjonstjenester, slik event-tjenesten gjør.

Nært knyttet til overstående problem ligger tidsaspektet. Dette bør implementeres i tjenestene som har behov for det og testes ut både med tanke på praktisk gjennomføring og i forhold til brukerne. Dette vil sannsynligvis forbedre kvaliteten og brukbarheten til tjenestene som allerede er implementert, i tillegg til å være en effektiv støtte til å håndtere dynamisk informasjon i større grad.

Her passer det også å nevne at det finnes ingen formater eller standarder for å støtte lokasjonsbaserte tjenester, og dette er en jobb som vil måtte gjøres innenfor dette feltet. I så måte er det foreslåtte formatet CULBIS et lite skritt på veien til å gjøre dette. Videre arbeid med dette bør fokusere på å støtte flest mulig tjenester og flest mulig posisjoneringsteknologier for å kunne være anvendbart for flere applikasjonsområder som søker å støtte enten bare et fåtall eller flere typer lokasjonsbaserte tjenester i samme system. Det er derfor også viktig at formatet er åpent i den forstand at det kan konfigureres slik det er behov for, slik CULBIS formatet har gjort gjennom skjemaer.

Hvis prototypen utviklet i denne oppgaven skulle videreutvikles ville følgende punkter være blant de mest aktuelle, i tillegg til nevnte problemstillinger:

- Implementere og teste systemet med bruk av koordinater og GPS
- Overføre logikken fra lokasjonsmodellen/dataformatet til en semantisk struktur som kan benyttes av en semantisk motor, likt det som gjøres i myCampus systemet, for å legge til rette for enklere utvikling av nye og forbedrede tjenester
- Lage objekter av lokasjoner i systemet, og benytte arv for typer steder til å lagre informasjon om stedene. Dette vil forenkle håndteringen av lokasjoner og deres tilhørende informasjon under utvikling av nye tjenester
- Utvikle en egen kart-modul som kan håndtere mer komplekse funksjoner som å tegne inn steder og brukere basert på stedsdefinisjonene og tilby stor grad av interaktivitet

- 
- Videreutvikle lokasjonmodellen til et XML format for å forbedre støtte for utveksling av den lokasjonsbaserte informasjonen
  - La brukergrensesnittet være web-basert. Dette vil føre til enklere utvikling av nye tjenester og er en metafor brukere er godt kjent med
  - Overføre systemets funksjoner og tjenester til et språk som lar seg kjøre på mindre mobile enheter som PDA. Et forslag er .NET Compact Framework
  - Se på metoder for å indeksere lokasjoner og informasjon for gjøre raskere oppslag
  - Se på metoder for å forutse brukerens posisjon litt frem i tid for å forbedre kontekstbevisstheten

## Referanser

- [1] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: a mobile context-aware tour guide. *Wirel. Netw.*, 3(5):421–433, 1997. ISSN 1022-0038. doi: <http://dx.doi.org/10.1023/A:1019194325861>.
- [2] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 304–307, London, UK, 1999. Springer-Verlag. ISBN 3-540-66550-1.
- [3] I.K. Adusei, K. Kyamakya, and F Erbas. Location-based services: Advances and challenges. In *Electrical and Computer Engineering, 2004. Canadian Conference on*, pages 1–7, Inst. of Commun. Eng., Hannover Univ., Germany, 2004. IEEE. ISBN 0-7803-8253-6. doi: <http://ieeexplore.ieee.org/iel5/9317/29618/01344944.pdf?isnumber=29618&prod=CNF&arnumber=1344944&arSt=+1&ared=+7+Vol.1&arAuthor=+Adusei%2C+I.K.%3B++Kyamakya%2C+K.%3B++Erbas%2C+F>.
- [4] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000. URL [citeseer.ist.psu.edu/bahl00radar.html](http://citeseer.ist.psu.edu/bahl00radar.html).
- [5] Paul Castro. The nibble location system website, 2001. URL <http://mms1.cs.ucla.edu/nibble/>. Sist besøkt: 12.12.05.
- [6] Keith Cheverst, Nigel Davies, Keith Mitchell, and Adrian Friday. Experiences of developing and deploying a context-aware tourist guide: the guide project. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 20–31, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-197-6. doi: <http://doi.acm.org/10.1145/345910.345916>.
- [7] Keith Cheverst, Keith Mitchell, and Nigel Davies. The role of adaptive hypermedia in a context-aware tourist guide. *Commun. ACM*, 45(5):47–51, 2002. ISSN 0001-0782. doi: <http://doi.acm.org/10.1145/506218.506244>.
- [8] Radionor Communications. Radioeye, 2006. URL <http://radionor.com/>. Sist besøkt: 07.06.06.
- [9] Alexandria Digital Library Gazetteer. Adl gazetteer content standard, 1999-. URL <http://www.ubiq.com/hypertext/weiser/UbiHome.html>. Sist besøkt: 03.05.06.
- [10] Geir Marius Gjøl. Lokasjons -og kontekstbaserte tjenester fra et informasjonsforvaltningsperspektiv. Master's thesis, NTNU Norges Teknisk-Naturvitenskapelige Universitet, 2006.

- [11] Anastasios Ioannidis, Manos Spanoudakis, Panos Sianas, Ioannis Priggouris, Stathes Hadjiefthymiades, and Lazaros Merakos. Using xml and related standards to support location based services. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 1629–1633, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-812-1. doi: <http://doi.acm.org/10.1145/967900.968226>.
- [12] Rui Jos&#233;, Adriano Moreira, Helena Rodrigues, and Nigel Davies. The around architecture for dynamic location-based services. *Mob. Netw. Appl.*, 8(4):377–387, 2003. ISSN 1383-469X. doi: <http://dx.doi.org/10.1023/A:1024531629631>.
- [13] Eija Kaasinen. User needs for location-aware mobile services. *Personal Ubiquitous Comput.*, 7(1):70–79, 2003. ISSN 1617-4909. doi: <http://dx.doi.org/10.1007/s00779-002-0214-7>.
- [14] B. Köbben, A.H. van Bunningen, and K. Muthukrishnan. Wireless Campus LBS - Building campus-wide Location Based Services based on WiFi technolog. In E. Stefanakis and M. Peterson, editors, *Proceedings of 1st International Workshop on Geographic Hypermedia.*, Denver (USA), 2005. International Cartographic Association, commission on Maps and the Internet. URL <http://db.ewi.utwente.nl/Publications/PaperStore/db-utwente-0000003681.pdf>.
- [15] Axel Küpper. *Location-based services : fundamentals and operation*. Chichester : Wiley, 2005. ISBN 0-470-09231-9.
- [16] Ray R. Larson. Geographic information retrieval and spatial browsing, 2000. URL [http://sherlock.berkeley.edu/geo\\_ir/PART1.html](http://sherlock.berkeley.edu/geo_ir/PART1.html). Sist besøkt: 23.05.06.
- [17] Sue Long, Rob Kooper, Gregory D. Abowd, and Christopher G. Atkeson. Rapid prototyping of mobile context-aware applications: the cyberguide case study. In *MobiCom '96: Proceedings of the 2nd annual international conference on Mobile computing and networking*, pages 97–107, New York, NY, USA, 1996. ACM Press. ISBN 0-89791-872-X. doi: <http://doi.acm.org/10.1145/236387.236412>.
- [18] MapBlast/Windows Live Local. Mapblast/windows live local, 2006. URL <http://www.mapblast.com>. Sist besøkt: 26.05.06.
- [19] Natalia Marmasse. commotion: a context-aware communication system. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, pages 320–321, New York, NY, USA, 1999. ACM Press. ISBN 1-58113-158-5. doi: <http://doi.acm.org/10.1145/632716.632910>.
- [20] Microsoft. .net framework 1.1, 2004. URL <http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=262D25E3-F589-4842-8157-034D1E7CF3A3>. Sist besøkt: 31.05.06.

- [21] Marius Milner. Netstumbler website, 2004. URL <http://www.stumbler.net/>. Sist besøkt: 13.12.05.
- [22] Richard Muntz, Dinesh Ajmera, Yun Chi, Murali Mani, Paul Castro, and Ted Kremenek. Muse website, 2002. URL <http://mmsl.cs.ucla.edu/muse/>. Sist besøkt: 12.12.05.
- [23] Michèle Nardelli. International it congress showcases smart planning in the palm of your hand, 2002. URL <http://www.unisa.edu.au/news/media2002/250202.htm>. Sist besøkt: 21.11.05.
- [24] Marit Olsen. Integrasjon og bruk av gazetteers og tesauri i digitale bibliotek. Master's thesis, NTNU Norges Teknisk-Naturvitenskapelige Universitet, 2004.
- [25] Stephen Parsli. Positioning mobile phones. Master's thesis, UiO Universitetet i Oslo, 2004.
- [26] Jason Pascoe. The stick-e note architecture: extending the interface beyond the user. In *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 261–264, New York, NY, USA, 1997. ACM Press. ISBN 0-89791-839-8. doi: <http://doi.acm.org/10.1145/238218.238344>.
- [27] James Reid. geoxwalk - a gazetteer server and service for uk academia. In Traugott Koch and Ingeborg Sølvberg, editors, *ECDL*, volume 2769 of *Lecture Notes in Computer Science*, pages 387–392. Springer, 2003. ISBN 3-540-40726-X.
- [28] Norman M. Sadeh, Ting-Chak Chan, Linh Van, OhByung Kwon, and Kazuaki Takizawa. A semantic web environment for context-aware m-commerce. In *EC '03: Proceedings of the 4th ACM conference on Electronic commerce*, pages 268–269, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-679-X. doi: <http://doi.acm.org/10.1145/779928.779992>.
- [29] Sumit Sen and Smita Sengupta. Open standards in location based services, 2002. URL <http://www.gisdevelopment.net/technology/lbs/techlbs002.htm>. Sist besøkt: 21.04.06.
- [30] Todd Simcock, Stephen Peter Hillenbrand, and Bruce H. Thomas. Developing a location based tourist guide application. In *CRPITS '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003*, pages 177–183, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc. ISBN 1-920682-00-7.
- [31] thefreedictionary.com. Definition - ubiquitous computing, 2005. URL <http://computing-dictionary.thefreedictionary.com/Ubiquitous+computing>. Sist besøkt: 03.05.06.
- [32] Mark Weiser. Ubiquitous computing, 1996. URL [http://www.alexandria.ucsb.edu/gazetteer/gaz\\_content\\_standard.html](http://www.alexandria.ucsb.edu/gazetteer/gaz_content_standard.html). Sist besøkt: 06.05.06.



## Appendix

### A - Brukermanual

#### A.1 Installering

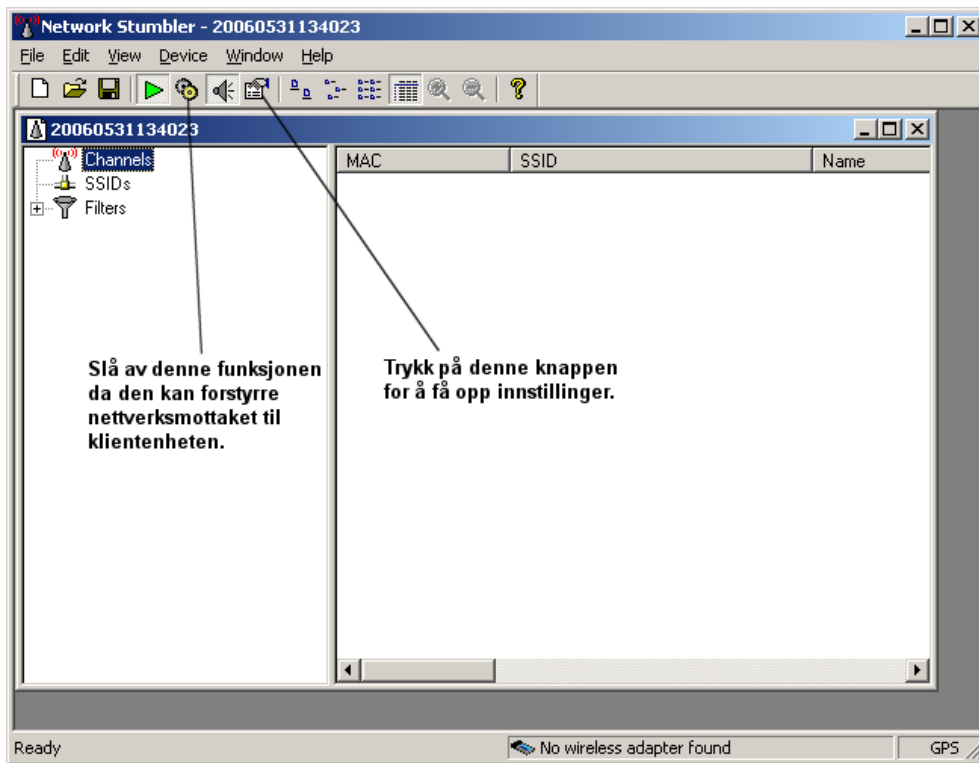
For å kunne kjøre applikasjonen trengs det enkelte tilleggskomponenter. Følgende må være installert:

1. Microsoft .NET Framework 1.1 [20]
2. Selve LOST (LOcation SysTem) applikasjonen
3. NetStumbler [21], med Visual Basic scriptet "fumble\_to\_db.vbs" kjørende

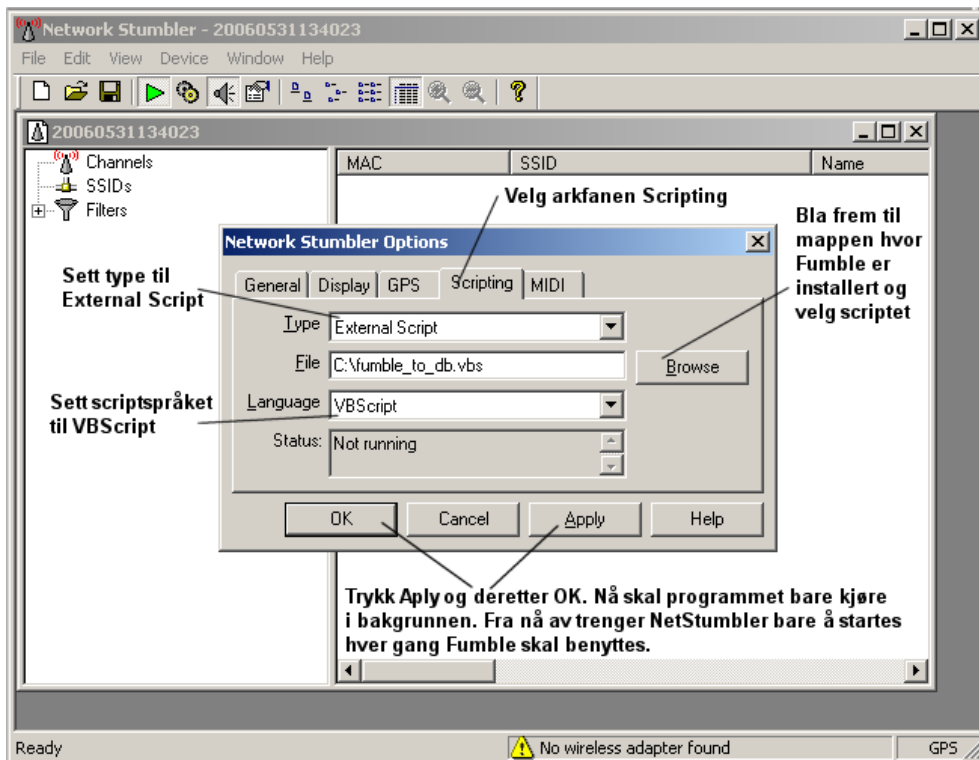
1. .NET Framework kan lastes ned fra Microsoft sine hjemmesider, og er også inkludert på vedlagt CD. Følg den automatiske instruksjonsveiledningen.

2. LOST applikasjonen installeres ganske enkelt ved å kopiere mappen inn på harddisken. Det er en fordel å la den ligge rett på C:\disken, som C:\Lost, da mellomrom i mappenavn kan skape problemer i enkelte tilfeller.

3. NetStumbler kan lastes ned fra hjemmesiden til programmet. Følg den automatiske installasjonsveiledningen. Når programmet er installert må scriptet "fumble\_to\_db.vbs" kjøres. Når dette er lagt inn må NetStumbler alltid kjøre i bakgrunnen når LOST skal brukes. For å kjøre scriptet gjøres følgende:



Figur 54: Installering av NetStumbler med script, steg 1.



Figur 55: Installering av NetStumbler med script, steg 2.

## A.2 Bruk

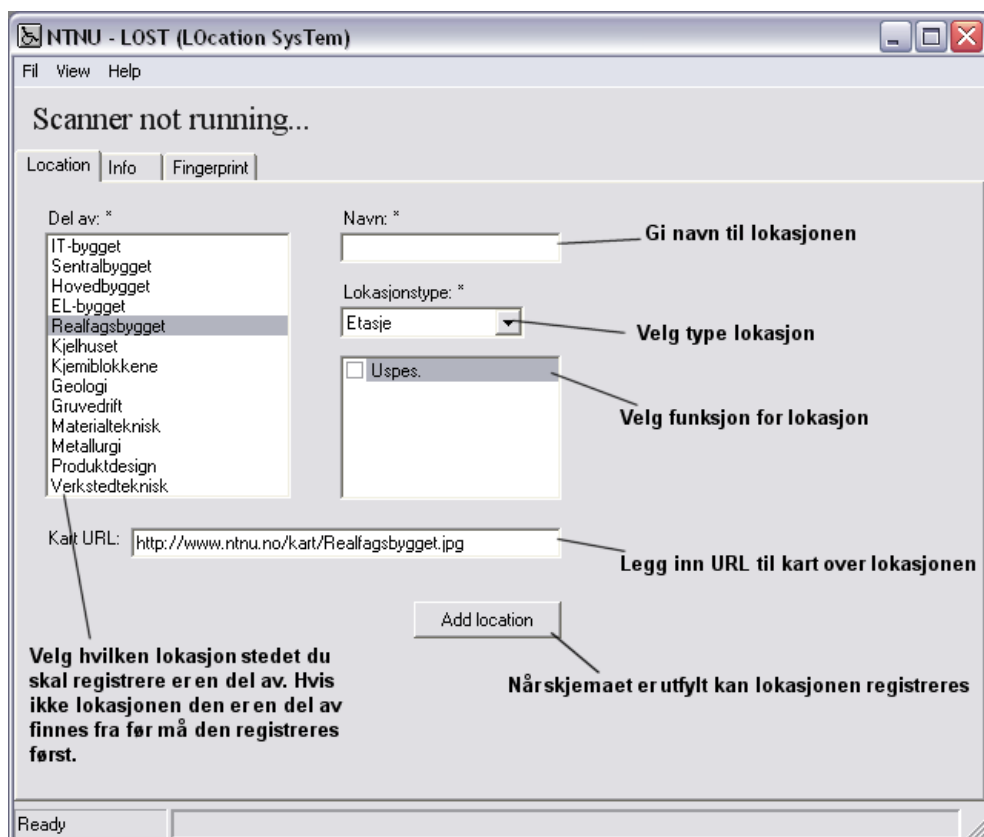
For å kunne bruke både administratoren og klienten må man ha NetStumbler kjørende i bakgrunnen. For klienten må man være pålogget VPN-nettverket med en VPN-klient, eller noe litt enklere, være autentisert med Web-Auth. I sistnevnte tilfelle starter man bare en nettleser og logger seg på nettverket med brukernavn og passord. Hvis man ikke har brukernavn eller passord kan man velge å logge seg inn som gjest.

### A.2.1 Administrator

For å kjøre LOST administratoren startes LostAdmin.exe. Administratoren gir adgang til å legge inn lokasjoner, informasjon om lokasjoner og WiFi Fingerprints i systemet, samt å kunne teste posisjoneringen.

### Lokasjoner

Fra arkfanen Location kan man legge inn lokasjoner.



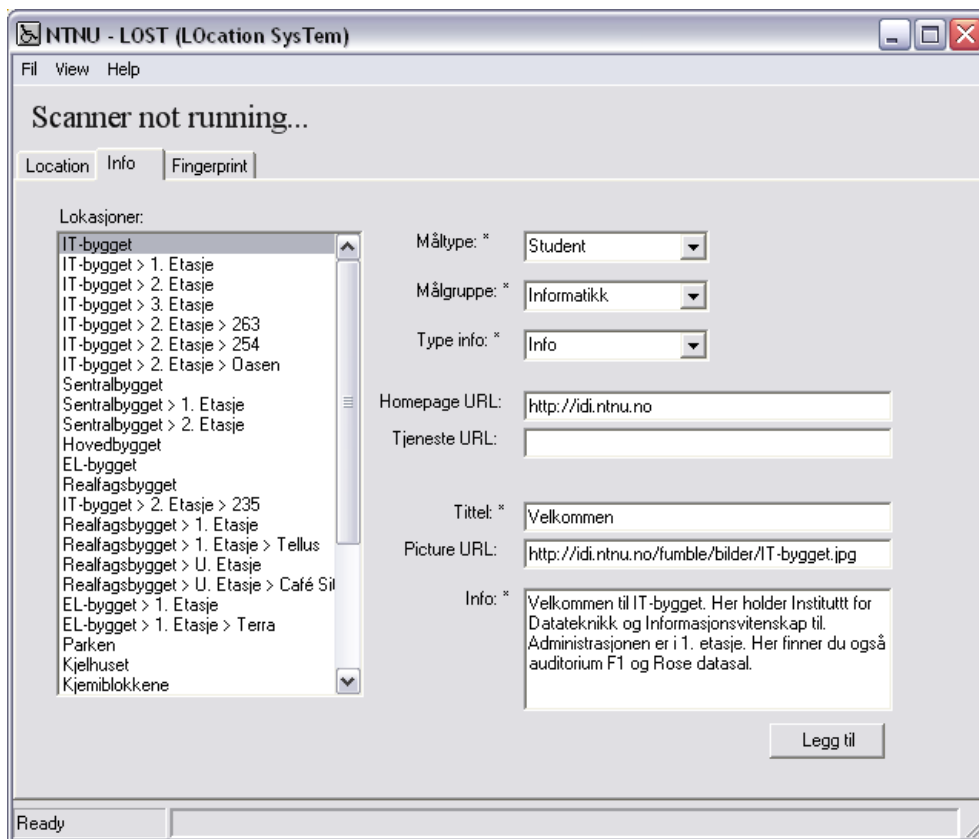
Figur 56: Innlegging av lokasjon i systemet via LostAdmin.

Lokasjonene gis et navn via navnefeltet. Hva slags type lokasjon som skal registreres velges fra listen over typer. Når aktuelle type er valgt vil listen til venstre i bildet vise hvilke lokasjoner som finnes i systemet som den nye lokasjonen kan

være en del av. Rom kan kunne være en del av en bestemt etasje, en etasje kan bare være en del av en bygning osv. Når type lokasjon er valgt vil også listen over funksjoner under type oppdateres til å gjenspeile funksjoner for f.eks. bygning eller rom. Her velges funksjon ved å hake av aktuelle funksjon. Hvis for eksempel man har valgt lokasjon av type "Rom" kan funksjoner være "arbeidsrom", "kopirom", "kontor", "kantine" og lignende. Hvis lokasjon ikke har funksjon som stemmer overens med noen av valgene kan man velge "Uspes." for uspesifisert funksjon. For å legge inn kart skrives URL for kart inn i boksen nederst. For å registrere innlagt lokasjon trykker man til slutt på knappen "Add location".

## Info

Fra arkfanen Info kan man legge inn informasjon om lokasjoner som allerede er



Figur 57: Innlegging av informasjon i systemet via LostAdmin.

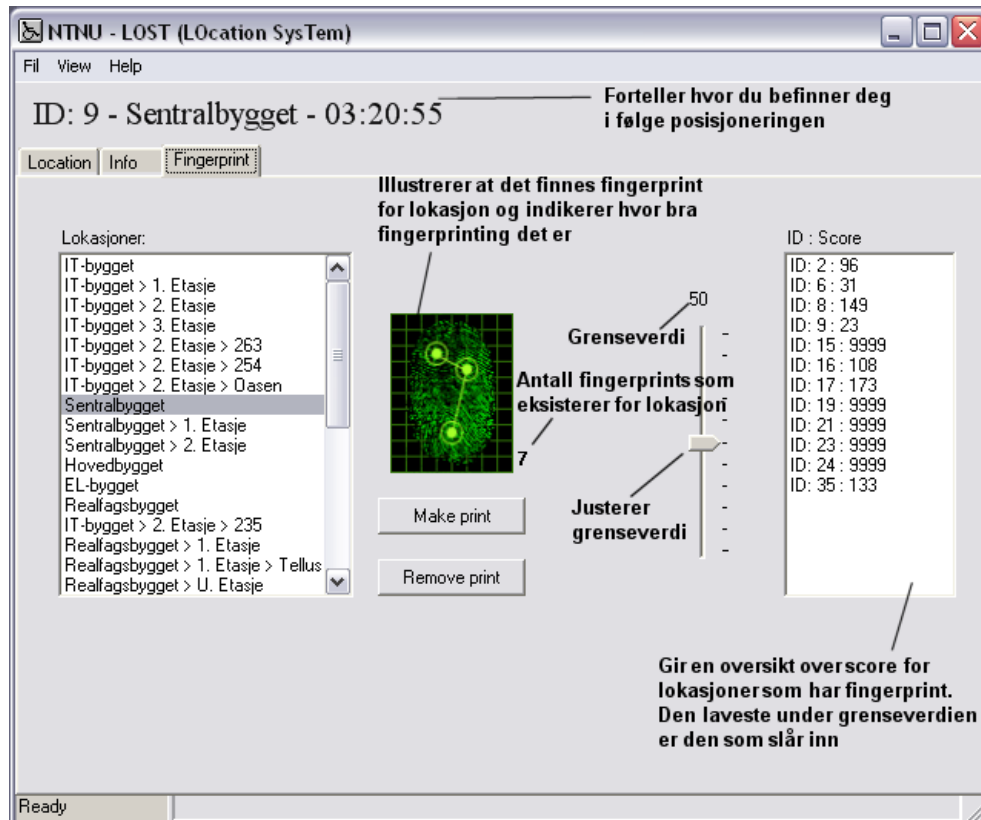
registrert i systemet.

Man velger da lokasjonen man vil legge inn informasjon for i listen til venstre i bildet. Deretter fyller man ut skjemaet til høyre. Brukertype og brukergruppe spesifiserer hvem informasjonen er rettet mot. Hvis det er studenter på informatikk det er rettet mot velges dette her. Type info velges, f.eks. generell info, reklame, veibeskrivelse o.l. Hvis lokasjonen har en hjemmeside legges URL for denne inn. Det samme gjelder for eventuelle interaktive webtjenester lokasjonen måtte ha, som for eksempel reservasjon av rom e.l. i feltet tjeneste URL. De tre nederste

feltene i skjemaet legger inn informasjonskapitler. Disse består av en tittel på kapitlet, et bilde som illustrasjon og en brødtekst. Hvis flere kapitler vil legges inn fylles dette inn på nytt og registreres. Informasjonen legges inn ved å trykke på ”Legg til” knappen.

## Fingerprint

Fra arkfanen Fingerprint kan man skape lokasjon fingerprints av WiFi signalene



Figur 58: Innlegging av fingerprint i systemet via LostAdmin.

som systemet vil bruke for å posisjonere brukerne.

Man plasserer seg på den aktuelle lokasjonen hvor man ønsker at posisjoneringen skal slå inn og lager så fingeravtrykk av signalene. Man velger først aktuelle lokasjon fra listen til venstre. Hvis det finnes fingeravtrykk fra før vil man kunne se indikasjon på dette i ruten rett til venstre for listen som da vil inneholde et bilde av et fingeravtrykk. Antallet fingeravtrykk for denne lokasjonen er angitt av tallet ved siden av bildet. Man kan lage så mange fingeravtrykk man ønsker for en lokasjon, jo flere avtrykk desto bedre vil presisjonen være. For å laget et nytt avtrykk trykker man ganske enkelt på ”Make print” knappen. Hvis man av en eller annen grunn ønsker å slette avtrykkene velger man aktuelle lokasjon og trykker på knappen merket ”Remove print”.

Listen til høyre er der for å kunne teste at lokasjonen man legger inn avtrykk for fungerer slik man ønsker. Hvis man bruker menyen kan man starte posisjoneringen i programmet ved å velge ”Fil” og ”Scan on”. Da vil man se hvor man

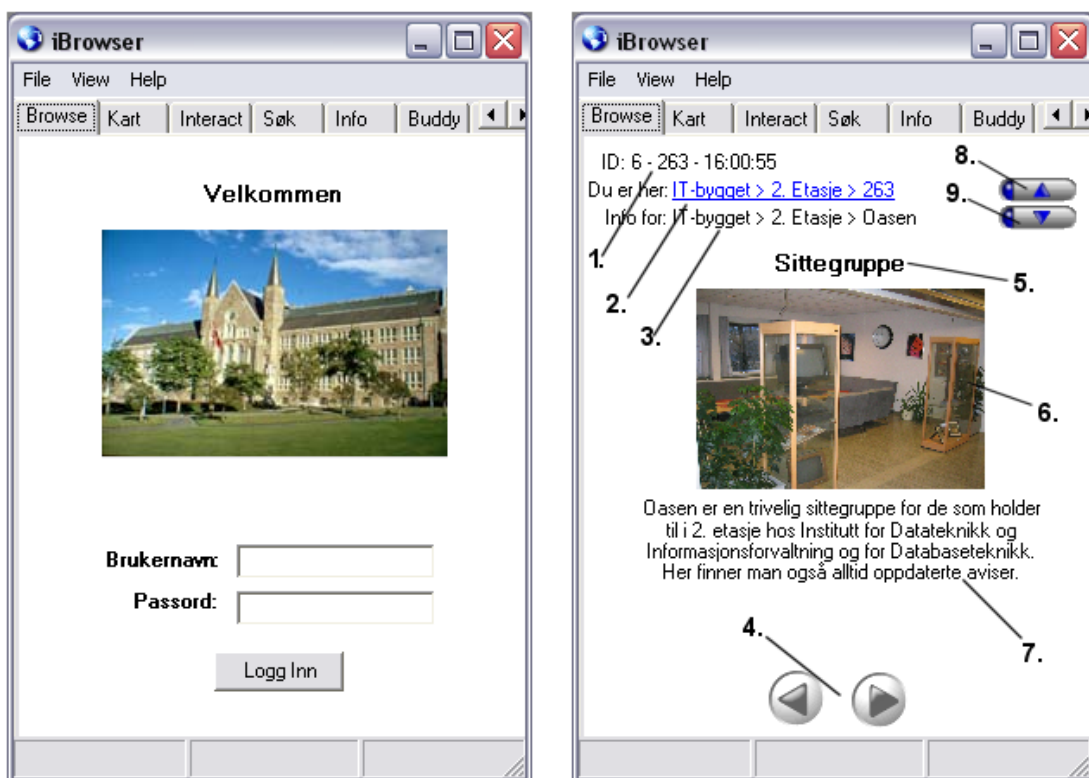
blir posisjonert på meldingen øverst på skjermbildet, og utreknet score for hver lokasjon med fingerprint vil listes. Hvis avtrykket man laget er for dårlig til å gi det utslaget man ønsker kan man lage flere avtrykk for å forbedre det. Hvis man ønsker å flytte grenseverdien for lokasjonsutslag (hvor god score et fingerprint må ha for å posisjoneres) kan man gjøre dette via kontrollen rett til venstre for listen.

## A.2.2 Brukerklient

For å kjøre brukerklienten kjøres LostClient.exe. Klienten gir adgang til å de lokasjonsbaserte tjenestene og informasjonen.

Det første man blir møtt av er en innloggingsskjerm. Brukeren logger seg inn ved å oppgi personlig brukernavn og passord som må fås av en administrator. Gjestekonto har brukernavn *gjest* og passord *gjest*. Deretter starter systemet posisjoneringen og man står fri til å benytte tjenestene.

### Browse



Figur 59: Innloggingsskjerm og infoskjerm for Lost-klienten.

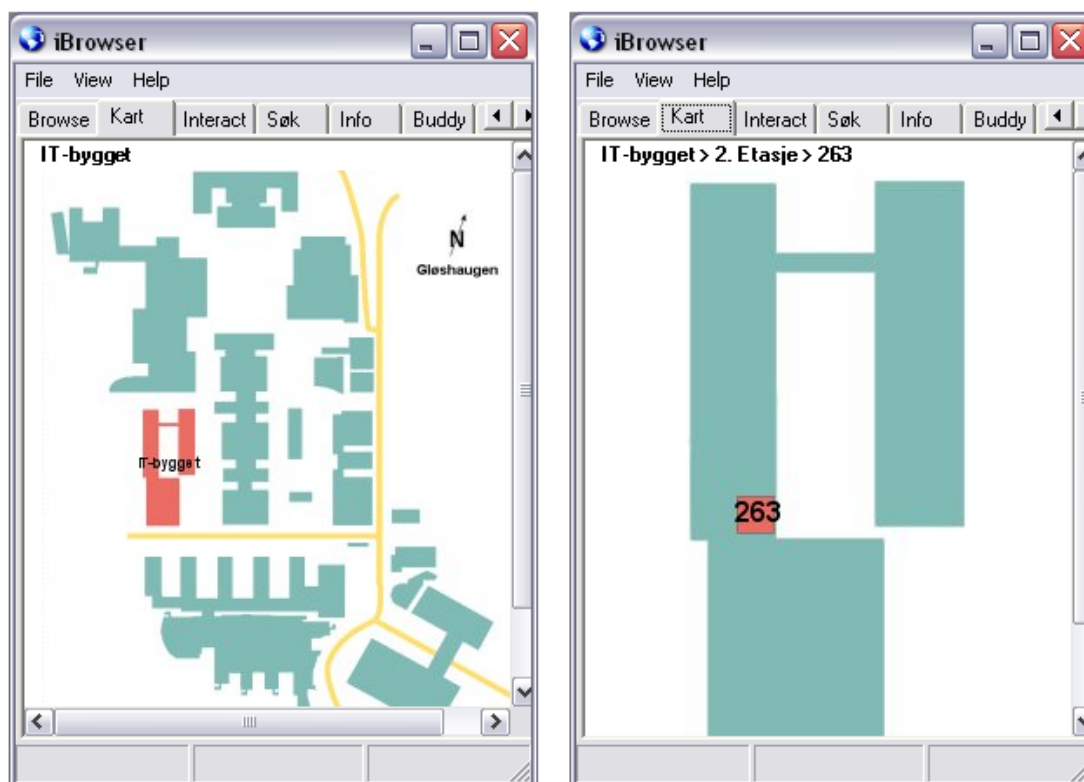
1. Denne strengen angir til enhver tid hvor systemet har posisjonert brukeren og når siste posisjonering ble gjort.
2. Denne linken viser hvor man sist ble posisjonert og fikk pushet info for. Den angir også sporingsveien i form av en sti på formen "bygning > etasje > rom". Hvis man trykker på denne linken vil man komme tilbake igjen til informasjonen om lokasjonen man befinner seg på uansett hvilken annen info man har navigert til.
3. Denne tekststrengen angir posisjon på samme måte som i punkt 2., men angir hvilken lokasjon man for øyeblikket ser informasjon om, uavhengig av hvor systemet har sporet brukeren. For eksempel hvis man har søkt opp informasjon om en lokasjon og vil se på denne.
4. Dette er frem og tilbake knapper som lar brukeren bla gjennom informasjons-

kapitler for hver lokasjon hvis det finnes flere. Hvis det ikke finnes flere vil disse knappene ikke være aktive.

5. Dette er en overskrift for hvert informasjonskapittel.
6. Bilde som vil bli vist for hvert informasjonskapittel.
7. Dette er brødteksten for hvert informasjonskapittel som skal fortelle brukeren noe om lokasjonen.
8. Denne knappen vil ta brukeren opp et nivå i lokasjonshierarkiet, f.eks. hvis man har info på skjermen for "Sentralbygget >1. Etasje >Hangaren" vil et trykk på denne skifte infoen til info for "Sentralbygget >1. Etasje" og nok et trykk helt ut til "Sentralbygget".
9. Denne knappen gjør det motsatte av forrige knapp, og lar brukeren gå ned igjen i hierarkiet.

## Kart

Hvis man skifter arkfane til "Kart" vil man få vist kart over hvor man befinner seg, eventuelt for lokasjonen man ser info for, siden kartet hele tiden gjenspeiler lokasjonen man har valgt å se info for. Granulariteten på kartet varierer etter hvor i lokasjonshierarkiet man er.



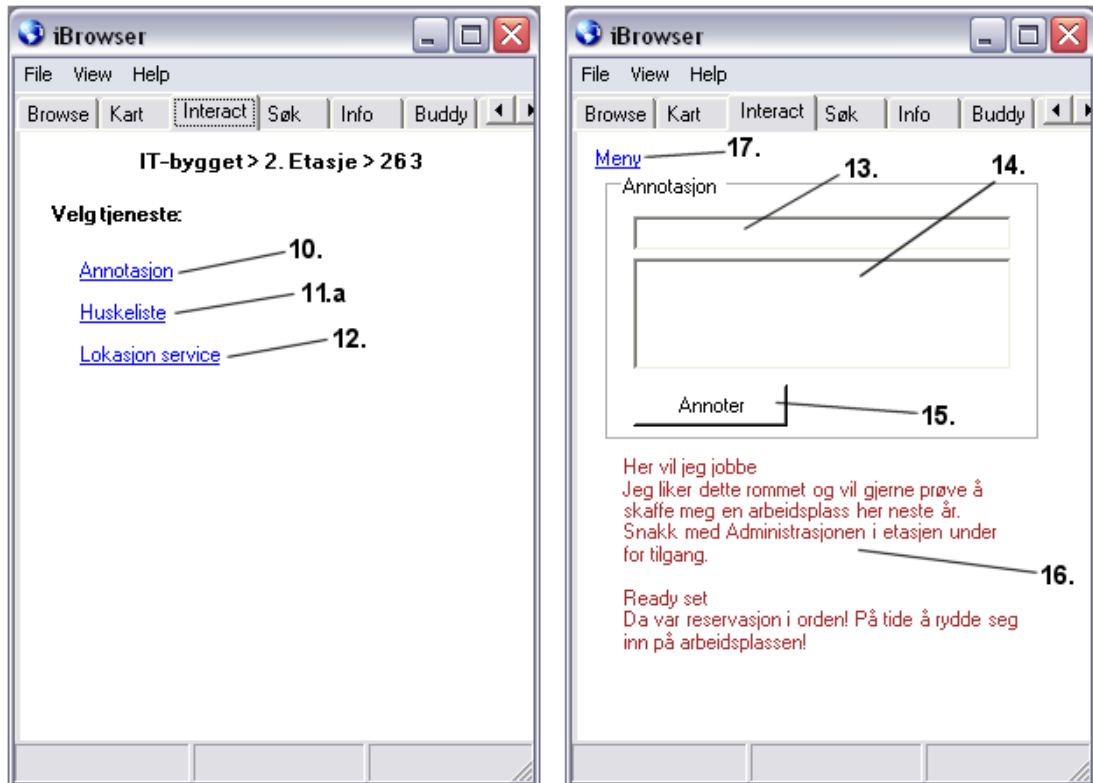
Figur 60: Kartvisning for Lost-klienten.

## Interact

På arkfane "Interact" finner man tjenester hvor brukeren kan interagere med lokasjonene og informasjon for lokasjonene. Teksten øverst angir hvilken lokasjon

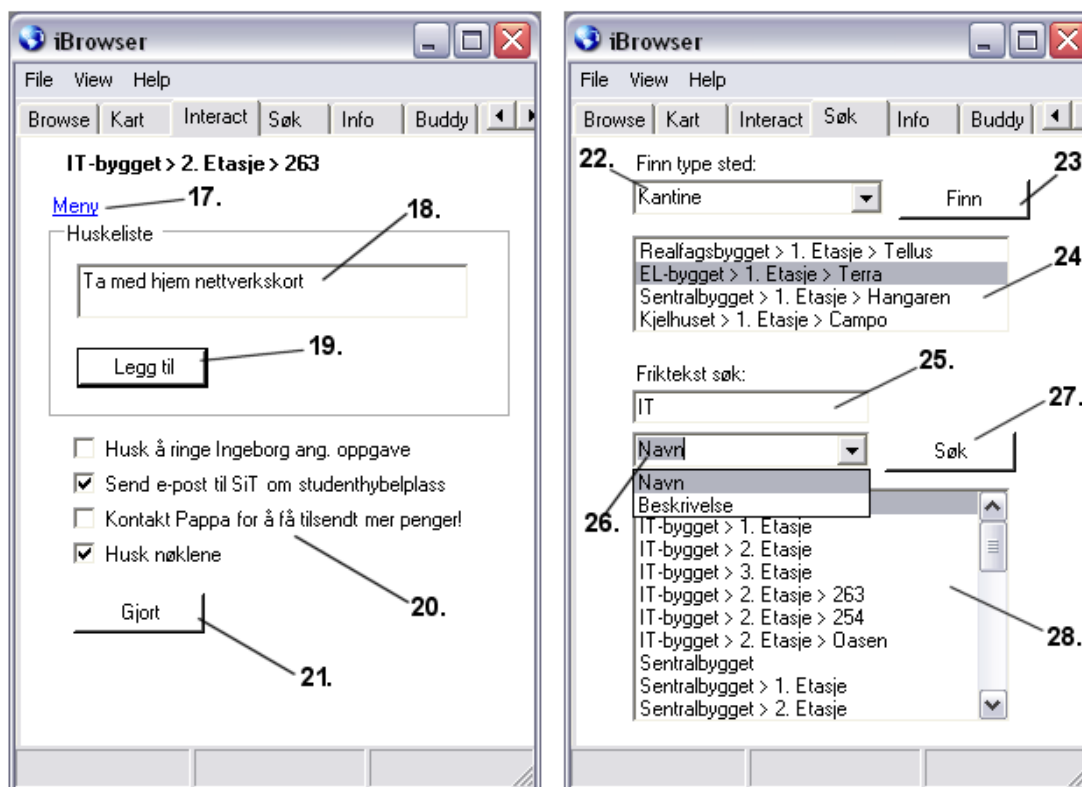


man bruker tjenester for, og gjenspeiler lokasjonen man sist ble posisjonert eller eventuelt har valgt sist i brukergrensesnittet.



Figur 61: Interaksjonsskjerm og annotering i Lost-klienten.

- 10. Annotasjon tar brukeren til skjermbildet for annoteringer på lokasjoner.
- 11.a Huskeliste tar brukeren til skjermbildet for huskeliste for lokasjoner.
- 11.b Systemet gir automatisk beskjed hvis du har oppføringer i huskelisten for lokasjonen du befinner deg på eller browser til. Dette ikonet varslers at brukeren har huskeliste for aktuelle lokasjon, og viser skjermbildet for huskeliste hvis brukeren klikker på ikonet (se figur 63).
- 12. Lokasjon service åpner lokasjonens webtjeneste i klientenhetens hovednettleser. Dette kan være f.eks. reservasjon av rom e.l.
- 13. På skjermen for annotasjon angir man en tittel på annotasjonen man vil legge inn.
- 14. Selve brødteksten på annotasjonen skrives inn i dette feltet.
- 15. Når man har fylt inn annotasjonen trykker man på "Annoter" knappen for legge den til lokasjonen.
- 16. Her vises annotasjonene som brukeren allerede har lagt inn for aktuelle lokasjon.
- 17. "Meny" linken tar brukeren tilbake til menyen for valg av interaksjonstjenester.
- 18. For å legge til nytt punkt i huskeliste for lokasjon skrives hva man skal huske i dette feltet.
- 19. Når man har skrevet inn punkt legges det til liste ved å trykke på "Legg til"



Figur 62: Huskeliste og Søk i Lost-klienten.

knappen.

**20.** Punkter man legger til vil legge seg i listen og vises for brukeren når han er på aktuelle lokasjon eller browser innom lokasjonen.

**21.** Når man har utført et punkt eller flere på huskelisten og ønsker å fjerne dem kan man hake de av og trykke på "Gjort" knappen og punktene vil bli slettet.

### Søk

På arkfanen "Søk" kan man søke etter lokasjoner og info. Her finner vi to forskjellige typer søk, en som leter opp lokasjoner basert på type, f.eks. printerrom, kantine, bibliotek osv., og en som søker i fritekst etter enten navn eller beskrivelse av lokasjon.

**22.** Her kan man spesifisere hva slags type sted man leter etter, f.eks. printerrom, kantine, bibliotek osv.

**23.** "Finn" knapper setter i gang søket etter type sted.

**24.** Her kommer resultatet av søk etter type lokasjon opp.

**25.** er kan man skrive inn et søkeord for å lete etter lokasjoner og informasjon.

**26.** I denne boksen angir man om man vil søke på navn eller i beskrivelser for lokasjon.

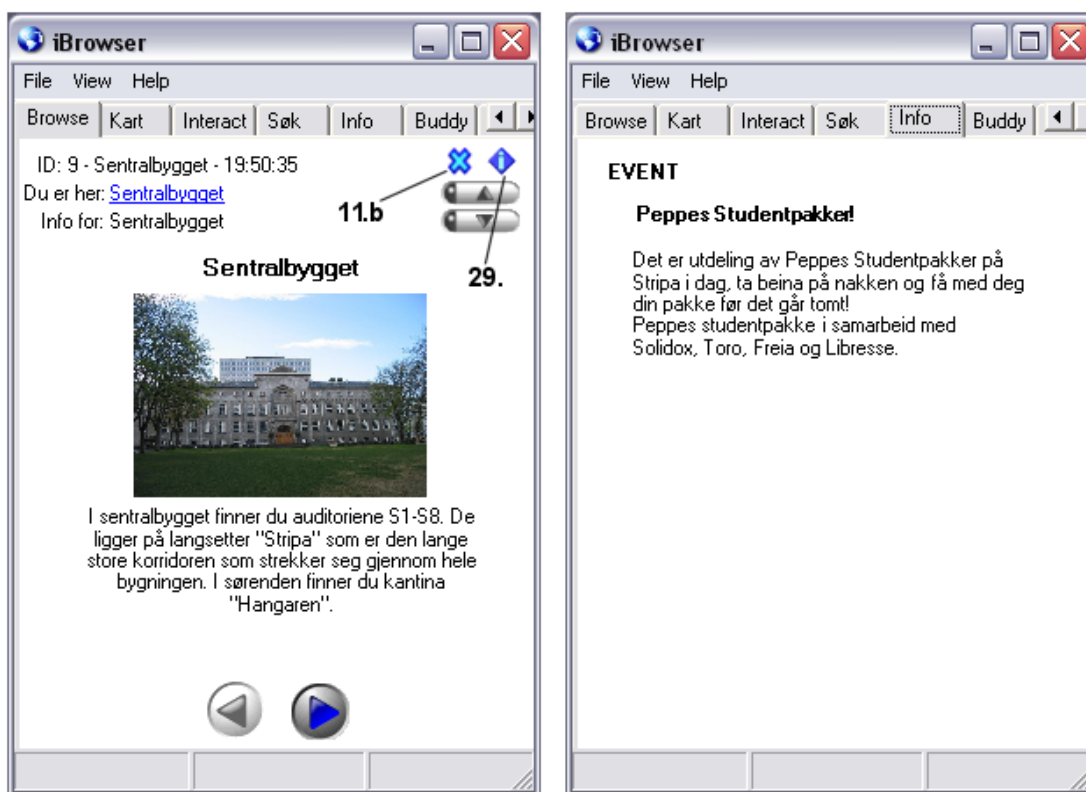
**27.** "Søk" knappen starter søket etter søkeordet (Tips: La søkeboksen stå tom og søk etter navn vil gi resultat på alle registrerte lokasjoner).

**28.** Her vises resultatet av søket. For å få vist informasjon om oppsøkte lokasjoner kan man dobbeltklikke på lokasjonen i listen.

### Info/event

Arkfanen "Info" Benyttes for å vise dynamisk informasjon om en lokasjon, f.eks. event informasjon om en hendelse ved en lokasjon, så som et arrangement som finner sted eller kommer til å finne sted ved lokasjonen.

**29.** Dette ikonet spretter fram for å gjøre brukeren oppmerksom på dynamiske informasjonsmeldinger som er tilgjengelig for lokasjonen. Hvis brukeren trykker på denne vil han få vist informasjonen i info arkfanen.



Figur 63: Event -og 2do ikon, og event info i Lost-klienten.

### Buddy

På arkfanen "Buddy" kan man benytte Buddy-tjenesten som lar brukeren spore kontakter i kontaktlisten sin for å finne ut hvor de befinner seg.

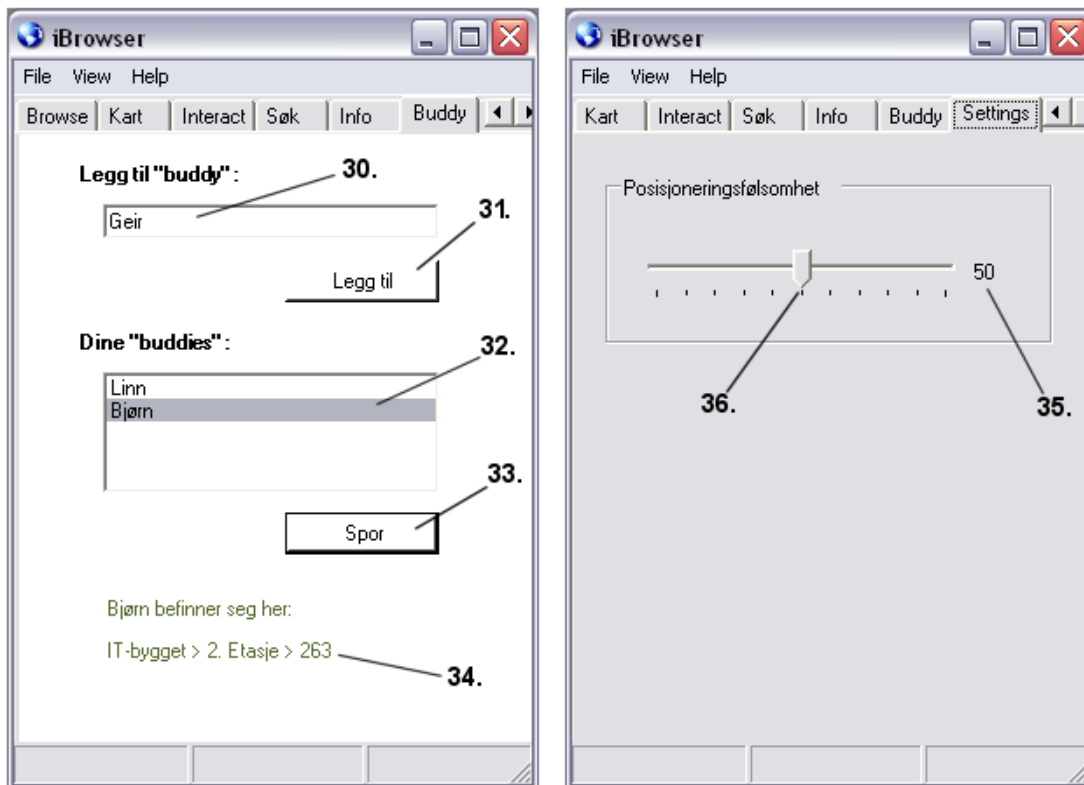
**30.** For å legge til en buddy (kontakt) kan du skrive inn navnet på kontakten i denne boksen.

**31.** Trykk deretter på "Legg til" knappen for å legge buddyen til listen.

**32.** For å spore en buddy, velg navnet fra listen.

33. Trykk deretter på "Spor" knappen for å finne kontakten.

34. Systemet forteller brukeren hvor buddyen er gjennom en sporingsti.



Figur 64: Buddy funksjon og innstillinger i Lost-klienten.

### Settings

på arkfanen "Settings" kan man justere posisjoneringsfølsomheten til systemet. Hvis man har problemer med å få inn sporing av sin egen lokasjon kan man justere opp følsomheten. Presisjonen blir da dårligere, men man har bedre sjanse for å få utslag.

35. Dette er grenseverdien for lokasjonsutslag. Når denne er høyere har systemet lettere for å få utslag på hvor man befinner seg, men presisjonen for å bestemme riktig lokasjon går ned.

36. Denne kontrollen justerer grenseverdien for lokasjonsutslag opp eller ned.

## B - Use Cases

### Use Case 1 - Informasjonsoppdagelse

---

**Primær aktør:** Student

**Interessenter og interesser:**

- Skolen (NTNU og alle fakultetene) ønsker å gi studenten relevant informasjon mens han/hun beveger seg rundt på Gløshaugen, slik at studenten raskt lærer det han/hun trenger å vite for å komme i gang med studiene og bli en effektiv student.

**Forutsetninger:** Studenten har en mobil enhet klargjort for tjenesten.

**Suksessgarantier:** Studenten oppdager og lærer nye ting mens han/hun beveger seg rundt på Gløshaugen.

**Hovedsuksess-scenario:**

1. Studenten velger å bruke tjenesten
2. Systemet ber om innloggingsinformasjon
3. Studenten logger inn
4. Systemet sporer studenten
5. Systemet gir relevant informasjon i forhold til studentens posisjon
6. Studenten mottar informasjonen og lærer av den
7. Systemet husker at denne informasjonen nå har blitt levert
8. Studenten logger av tjenesten

**Varianter:**

3a Studenten oppgir gal innloggingsinformasjon

1. Systemet opplyser at informasjonen var gal, og viser ny innloggingsskjerm samtidig som det spør om studenten ønsker å registrere seg som bruker
2. Studenten logger inn

4a Systemet greier ikke å spore studenten

1. Systemet opplyser om feil, og gir informasjon om support
2. Systemet gir tilgang til manuell søking vha. søkeord og posisjoner

**Spesielle krav:**

- Systemet skal være såpass enkelt og intuitivt at studentene ikke behøver bruke mer enn maks 5 minutter på å sette seg inn i hvordan man skal komme i gang med tjenesten.

**Teknologi og datavarianter:**

- Bør støtte forskjellige enheter så lenge de benytter samme teknologi

**Frekvens:** Skal kunne håndtere X studenter

**Åpne spørsmål:**

- Hva er datatilsynets krav?

---

## Use Case 2 - Finn nærmeste datasal

---

**Primær aktør:** Student

**Interessenter og interesser:**

- Skolen (NTNU og alle fakultetene) ønsker å tilby studenten den informasjonen han trenger.

**Forutsetninger:** Studenten har en mobil enhet klargjort for tjenesten.

**Suksessgarantier:** Studenten finner nærmeste datasal hvor han har tilgang, og eventuell aktuell informasjon om denne.

**Hovedsuksess-scenario:**

1. Studenten forespør informasjon om nærmeste fasilitet, datasal
2. Systemet sporer studenten
3. Systemet gir relevant informasjon i forhold til studentens posisjon og profil
4. Studenten mottar informasjonen og lærer av den
5. Studenten logger av tjenesten/fortsetter med nye forespørsler

**Varianter:**

2a Systemet greier ikke å spore studenten

1. Systemet opplyser om feil, og gir informasjon om support
2. Systemet gir tilgang til manuell søking vha. søkeord og posisjoner/lokasjoner

3a Systemet finner ingen datasal som samsvarer med profil

1. Systemet opplyser at ingen datasal er tilgjengelig for bruker
2. Systemet viser nærmeste datasal basert kun på lokasjon

**Spesielle krav:**

- Systemet skal være såpass enkelt og intuitivt at studentene ikke behøver bruke mer enn maks 5 minutter på å sette seg inn i hvordan man skal komme i gang med tjenesten.

**Teknologi og datavarianter:**

- Bør støtte forskjellige enheter så lenge de benytter samme teknologi

**Frekvens:** Skal kunne håndtere X studenter

**Åpne spørsmål:**

- Hva er datatilsynets krav?

---

### Use Case 3 - Hva er her?

---

**Primær aktør:** Student

**Interessenter og interesser:**

- Skolen (NTNU og alle fakultetene) ønsker å tilby studenten den informasjonen han trenger.
- SiT har interesse av å formidle sine tilbud/tjenester til studentene.
- Andre mulige foretak på Gløshaugen har interesse av å gi seg til kjenne.

**Forutsetninger:** Studenten har en mobil enhet klargjort for tjenesten og er logget inn i systemet.

**Suksessgarantier:** Studenten får den informasjonen han ønsker om det stedet han befinner seg på.

**Hovedsuksess-scenario:**

1. Studenten forespør informasjon om lokasjonen han befinner seg på
2. Systemet sporer studenten
3. Systemet gir relevant informasjon i forhold til studentens posisjon og profil
4. Studenten mottar informasjonen og lærer av den
5. Studenten logger av tjenesten/fortsetter med nye forespørsler

**Varianter:**

2a Systemet greier ikke å spore studenten

1. Systemet opplyser om feil, og gir informasjon om support
2. Systemet gir tilgang til manuell søking vha. søkeord og posisjoner/lokasjoner

3a Systemet finner ingen info om lokasjonen

1. Systemet finner info om nærliggende lokasjoner
2. Systemet lar brukeren velge om han vil se info om hver enkelt av disse

**Spesielle krav:**

- Systemet skal være såpass enkelt og intuitivt at studentene ikke behøver bruke mer enn maks 5 minutter på å sette seg inn i hvordan man skal komme i gang med tjenesten.

**Teknologi og datavarianter:**

- Bør støtte forskjellige enheter så lenge de benytter samme teknologi

**Frekvens:** Skal kunne håndtere X studenter

**Åpne spørsmål:**

- Hva er datatilsynets krav?
-

## Use Case 4 - Event message

---

**Primær aktør:** Student

**Interessenter og interesser:**

- Skolen (NTNU og alle fakultetene) ønsker å tilby studenten informasjon om arrangementer og andre organiserte tilstelninger. - Andre mulige foretak har interesse av å formidle sine aktiviteter på Gløshaugen.

**Forutsetninger:** Studenten har en mobil enhet klargjort for tjenesten og er logget inn i systemet.

**Suksessgarantier:** Studenten får informasjon om arrangementet til passende tid og sted.

**Hovedsuksess-scenario:**

1. Systemet sporer studenten
2. Systemet matcher sted, tid og profil for aktuelt arrangement
3. Systemet gir informasjon om aktuelt arrangement som foregår i nærheten av studentens lokasjon, til nåværende tid, og som er relevant for studenten
4. Studenten mottar informasjonen og avgjør om han ønsker å stikke innom arrangementet
5. Studenten logger av tjenesten/gjør forespørsler/fortsetter å bruke tjenesten

**Varianter:**

**Spesielle krav:**

- Systemet skal være såpass enkelt og intuitivt at studentene ikke behøver bruke mer enn maks 5 minutter på å sette seg inn i hvordan man skal komme i gang med tjenesten.

**Teknologi og datavarianter:**

- Bør støtte forskjellige enheter så lenge de benytter samme teknologi

**Frekvens:** Skal kunne håndtere X studenter

**Åpne spørsmål:**

- Hva er datatilsynets krav? - Hvor lenge i forveien bør arrangement utlyses? (Kan la forfatter av infoen avgjøre?)

---



---

## Use Case 5 - Brannmelding og sikkerhet

---

**Primær aktør:** Student

**Interessenter og interesser:**

- Skolen (NTNU og alle fakultetene) ønsker å tilby studenten den informasjonen han trenger. - Brannvesenet ønsker best mulig sikring mot brann og skader ved brann.

**Forutsetninger:** Studenten har en mobil enhet klargjort for tjenesten.

**Suksessgarantier:** Studenten finner raskest mulig en sikker vei ut, og evt. hjelpemidler han trenger for å sikre sin egen og andres sikkerhet.

**Hovedsuksess-scenario:**

1. Brannalarm går, og systemet iverksetter brannmeldingstjeneste
2. Systemet sporer studenter i aktuelle bygning/område
3. Systemet gir informasjon om nærmeste rømningsvei samt andre alternative rømningsveier
4. Systemet gir informasjon om nærliggende slukningsutstyr (gjerne sammen med pkt. 3)
5. Studenten finner raskest mulig rømningsvei og nødvendig slukningsutstyr
6. Studenten logger av/fortsetter å bruke systemet

**Varianter:**

5a Studenten kommer ikke ut

1. Systemet tillater sporing av studenten for å berge ham ut
2. Redningsarbeidere finner studenten og berger ham ut

**Spesielle krav:**

- Systemet skal være såpass enkelt og intuitivt at studentene ikke behøver bruke mer enn maks 5 minutter på å sette seg inn i hvordan man skal komme i gang med tjenesten. - Systemet må være koblet til brannalarmsystemet, evt. må en ansvarlig iverksette brannmeldingen.

**Teknologi og datavarianter:**

- Bør støtte forskjellige enheter så lenge de benytter samme teknologi

**Frekvens:** Skal kunne håndtere X studenter

**Åpne spørsmål:**

- Hva er datatilsynets krav?

---

## Use Case 6 - Student tracker

---

**Primær aktør:** Student

**Interessenter og interesser:**

- Skolen (NTNU og alle fakultetene) ønsker å gi studenten relevant informasjon mens han/hun beveger seg rundt på Gløshaugen, slik at studenten raskt lærer det han/hun trenger å vite for å komme i gang med studiene og bli en effektiv student.

**Forutsetninger:** Studenten har en mobil enhet klargjort for tjenesten.

**Suksessgarantier:** Studenten finner medstudenter og andre relevante personer på Gløshaugen.

**Hovedsuksess-scenario:**

1. Studenten forespør informasjon om hvor studenter som deltar på en event, studium eller fag befinner seg
2. Systemet sporer studentene
3. Systemet viser hvor studentene befinner seg i forhold til brukeren
4. Studenten finner sine medstudenter
5. Studenten logger av tjenesten/fortsetter med nye forespørsler

**Varianter:**

2a Systemet kan ikke finne studentene

1. Systemet opplyser at ingen studenter ble funnet
2. Systemet lar studenten gjøre ny forespørsel eller fortsette med andre tjenester

**Spesielle krav:**

- Systemet skal være såpass enkelt og intuitivt at studentene ikke behøver bruker mer enn maks 5 minutter på å sette seg inn i hvordan man skal komme i gang med tjenesten. - Studenter må tillate sporing (informasjon om hvem man er vil ikke være tilgjengelig ved sporing, kun at man er en deltaker)

**Teknologi og datavarianter:**

- Bør støtte forskjellige enheter så lenge de benytter samme teknologi

**Frekvens:** Skal kunne håndtere X studenter

**Åpne spørsmål:**

- Hva er datatilsynets krav?

---

## Use Case 7 - Romreservering

---

**Primær aktør:** Student

**Interessenter og interesser:**

- Skolen (NTNU og alle fakultetene) ønsker å tilby studenten relevante tjenester mens han/hun beveger seg rundt på Gløshaugen, slik at studenten raskt finner seg til rette for å komme i gang med studiene og bli en effektiv student.

**Forutsetninger:** Studenten har en mobil enhet klargjort for tjenesten.

**Suksessgarantier:** Studenten får reservert riktig rom til aktuell tid.

**Hovedsuksess-scenarior:**

1. Studenten har lokalisert aktuelle rom, og velger å bruke tjenesten for romreservering
2. Studenten velger tid for reservering
3. Systemet sjekker hvorvidt rommet er ledig ved aktuelle tidspunkt
4. Systemet reserverer rommet og gir studenten beskjed om reservasjonen
5. Systemet minner studenten på reservasjonen når tiden nærmer seg

**Varianter:**

3a Rommet er ikke ledig ved aktuelle tidspunkt

1. Systemet opplyser at rommet ikke er ledig
2. Systemet lar studenten velge alternativt tidspunkt

5a Studenten kan ikke lokaliseres på campus ved reservert tidspunkt

1. Systemet sletter reservasjonen

**Spesielle krav:**

- Systemet skal være såpass enkelt og intuitivt at studentene ikke behøver bruke mer enn maks 5 minutter på å sette seg inn i hvordan man skal komme i gang med tjenesten. - Studenter må tillate sporing (informasjon om hvem man er vil ikke være tilgjengelig ved sporing, kun at man er en deltaker)

**Teknologi og datavarianter:**

- Bør støtte forskjellige enheter så lenge de benytter samme teknologi

**Frekvens:** Skal kunne håndtere X studenter

**Åpne spørsmål:**

- Hva er datatilsynets krav?

---

## C - ADL GCS

Elementer som er markert med en asterisk(\*) og uthevet skrift er obligatoriske. Andre elementer er valgfrie. Når et obligatorisk element er et sub-element til et ikke-obligatorisk element, er dette obligatorisk kun såfremt det overordnede elementet er i bruk. Repeterbare elementer etterfølges av (R). Betegnelser for dataelementene er omsluttet av vinkelparenteser <...> Forklaringer for bruk og bruksområde er omsluttet av hakeparenteser [...] Standardverdier er omsluttet av vanlige parenteser (...) Tillatte verdier er omsluttet av klammeparenteser {...}

1. **Geographic Feature ID \*** <feature\_id>
2. **Geographic Name \***
  1. **Name \*** <primary\_name>[the primary name for feature in a particular gazetteer application]
  2. Name Source <name\_source>
  3. Etymology <etymology>
  4. Language <language\_used>(default is English)
  5. Pronunciation <pronunciation>
  6. Transliteration Scheme Used <trans\_schema>
  7. Character Set (default is ASCII) <character\_set>
  8. **Current / Historical Note \*** (default is Current) <is\_current>boolean
  9. Beginning Date <beginning\_date>
  10. Ending Date <ending\_date>
  11. Time Period Note <time\_period\_note>
  12. Source Mnemonic <source\_mnemonic>
  13. Entry Date <entry\_date>
3. Variant Geographic Name (R)
  1. **Variant Name \*** <feature\_name >
  2. Name Source <name\_source >
  3. Etymology <etymology >
  4. Language <language\_used >(default is English)
  5. Pronunciation <pronunciation >
  6. Transliteration Scheme Used <trans\_schema >
  7. Character Set (default is ASCII) <character\_set >
  8. **Current / Historical Note \*** (default is Current) <is\_current >boolean
  9. Beginning Date <beginning\_date >
  10. Ending Date <ending\_date >
  11. Time Period Note <time\_period\_note >
  12. Source Mnemonic <source\_mnemonic >
  13. Entry Date <entry\_date >
4. **Type of Geographic Feature \***
  1. **Feature Type Schema \*** <schema\_name>[e.g., ADL Feature Type Thesaurus]
  2. **Feature Type \*** (R) <type >
5. Other Classification Terms (R)
  1. Classification Schema <classification\_name >

2. Classification Term (R) <term >
6. Geographic Feature Code (R)
  1. **Feature Code Schema** \* <code\_schema >
  2. **Feature Code** \* <code >
  3. Source Mnemonic <source\_mnemonic >
  4. Entry Date <entry\_date >
7. **Spatial Location** \*
  1. **Planetary Body** \* <planet\_body >(default of Earth)
  2. Spatial Representation \* (R)
    1. **Bounding Box Spatial Geometry Representation** \*
      - <primary\_bounding\_box>
      - 1. **West Bounding Coordinate** \* <west\_bounding\_coor >
      - 2. **East Bounding Coordinate** \* <east\_bounding\_coor >
      - 3. **South Bounding Coordinate** \* <south\_bounding\_coor >
      - 4. **North Bounding Coordinate** \* <north\_bounding\_coor >
      - 5. **Current / Historical Note** \* (default is Current)
        - <is\_current >{boolean}
      - 6. Beginning Date <beginning\_date >
      - 7. Ending Date <ending\_date >
      - 8. Time Period Note <time\_period\_note >
      - 9. Measurement Date, Beginning Date <measurement\_begin\_date >
      - 10. Measurement Date, Ending Date <measurement\_end\_date >
      - 11. Method of Measurement <measurement\_method >
      - 12. Accuracy of Measurement <measurement\_accuracy >
      - 13. Source Mnemonic <source\_mnemonic >
      - 14. Entry Date <entry\_date >
    2. Detailed Spatial Geometry Representation (R) <g\_points >[set of points; dependent on system capabilities and requirements; can represent set of non-contiguous areas]
      1. **Detailed Spatial Geometry Representation** \*
        - <geometry\_type>{point, bounding box, linear, complex object}
      2. **Number of Points** \* <num\_points >
      3. **Points Order** \* <points\_order >
      4. **(Longitude, Latitude)** \* (R) <long\_lat\_pairs >
      5. **Current / Historical Note** \* (default is Current) <is\_current >boolean
      6. Beginning Date <beginning\_date >
      7. Ending Date <ending\_date >
      8. Time Period Note <time\_period\_note >
      9. Measurement Date, Beginning Date <measurement\_begin\_date >
      10. Measurement Date, Ending Date <measurement\_end\_date >
      11. Method of Measurement <measurement\_method >
      12. Accuracy of Measurement <measurement\_accuracy >
      13. Source Mnemonic <source\_mnemonic >
      14. Entry Date <entry\_date >
8. Street Address (Physical Address)
  1. **Address** \* <street\_address >
  2. **City** \* <city >
  3. **State or Province** \* <state\_province >
  4. Postal Code <postal\_code >
  5. Country <country >
  6. Source Mnemonic <source\_mnemonic >

7. Entry Date <entry\_date >
9. Related Feature (R)
  1. **Type of Relationship** \* <relationship\_type >in-state-of, in-province-of, in-county-of, in-country-of, in-region-of, part-of, formerly-known-as
  2. **Geographic Name of Related Feature** \* <related\_name >
  3. Related ADL Feature ID <related\_feature\_id >
  4. **Current / Historical Note for Relationship** \* <is\_current >
  5. Beginning Date of Relationship <begin\_date\_rel >
  6. Ending Date of Relationship <end\_date\_rel >
  7. Time Period Note <time\_period\_note >
  8. Source Mnemonic <source\_mnemonic >
  9. Entry Date <entry\_date >
10. Description
  1. **Short Description** \* <short\_description >
  2. Source Mnemonic <source\_mnemonic >
  3. Entry Date <entry\_date >
11. Geographic Feature Data (R)
  1. **Type of Geographic Feature Data** \* <label\_name >
  2. **Geographic Feature Data Value**\* <data\_value >
  3. **Geographic Feature Data Value Unit of Measure** \* <measurement\_unit >
  4. Explanatory Note <explanatory\_note >
  5. Beginning Date <beginning\_date >
  6. Ending Date <ending\_date >
  7. Time Period Note <time\_period\_note >
  8. Source Mnemonic <source\_mnemonic >
  9. Entry Date <entry\_date >
12. Link to Related Source of Information (R)
  1. **Description of Linked Item** \* <link\_description >
  2. **URL** \* <link\_url >
  3. Source Mnemonic <source\_mnemonic >
  4. Entry Date <entry\_date >
13. Supplemental Note
  1. **Note** \* <note >
  2. Source Mnemonic <source\_mnemonic >
  3. Entry Date <entry\_date >
14. **Metadata Information** \*
  1. Entry Note <entry\_note>
  2. **Entry Date** \* <entry\_date >
  3. **Modification Date** \* <mod\_date >

I tillegg til selve gazetteer-posten er det en nødvendig kildedel som må være med siden denne informasjonen skal kunne deles mellom flere organisasjoner, og det derfor er hensiktsmessig å holde rede på hvor informasjonen kommer fra.

**Source Information** \* (R)

1. **Source Mnemonic** \* <source\_mnemonic >
2. **Contributor Organization** \* <organization\_name >
3. Contributor Web Site
  1. **Web Site Title** \* <web\_site\_title >
  2. **URL** \* <web\_site\_url >
4. Contact Person <contact\_person >
5. Email <email >
6. Telephone Number <telephone >
7. **Contributor Address** \*
  1. **Address** \* <street\_address >
  2. **City** \* <city >
  3. **State or Province** \* <state\_province >
  4. **Postal Code** \* <postal\_code >
  5. **Country** \* <country >
8. **Source Information** \* (R)
  1. Author Statement <author >
  2. **Title** \* <title >
  3. Edition <edition >
  4. Series Name <series\_name >
  5. Series Issue <series\_issue >
  6. **Publisher** \* <publisher >
  7. **Publication Date** \* <publication\_date >
  8. Publication Date Note <pub\_date\_note >
  9. Pages <pages >
  10. Source Identifier <source\_identifier > [e.g., ISBN]
  11. Source\_URL <source\_url >

## D - Skjema for lokasjonsmodellen

### LokTypeFunk

lokTypeNavn	funkNavn
Bygning	Bibliotek
Bygning	Kantine
Bygning	Kiosk
Bygning	Uspes.
Etasje	Uspes.
Plass	Parkeringsplass
Plass	Uspes.
Rom	Arbeidskontor
Rom	Arbeidsrom
Rom	Auditorium
Rom	Bibliotek
Rom	Datasal
Rom	Grupperom
Rom	Instituttkontor
Rom	Kantine
Rom	Kiosk
Rom	Kopiroom
Rom	Lesesal
Rom	Printerrom
Rom	Toalett
Rom	Uspes.
Vei	Uspes.

### Brukergruppe/Brukertype

brukergroupe_ID	brukertype_ID
Alle	Alle
Data	Student
Elektro	Student
IDI	Ansatt
Informatikk	Student
Kybernetikk	Student
Vaktmester	Ansatt

### InfoArt

infoArtNavn
Info
Kunngjøring
Meny
Reklame

### Kontekst

kontekst_ID
Arbeid
Fritid
Skole

### RelType

relType_ID
del_av
sees_fra

### LokTypeRel

lokTypeNavnOver	lokTypeNavnUnder
Bygning	Etasje
Campus	Bygning
Campus	Plass
Campus	Vei
Etasje	Rom

Figur 65: Data i skjema for prototypen.

Disse skjemaene legger til rette for deler av logikken i CULBIS formatet. De aktuelle tabellene spesifiseres ut i fra hva man ønsker å oppnå med implementasjonen. Vi ser at tabellen LokTypeRel holder rede på en enkel hierarkisk struktur over typer lokasjoner, hvor bygninger ligger under campus, etasjer ligger under bygninger og rom ligger under etasjer. Hvis det hadde vært behov for hadde man kunnet spesifisere at en avdeling eller fløy i en bygning inngikk som et ledd mellom etasje og rom, eller på annen måte man hadde behov for.



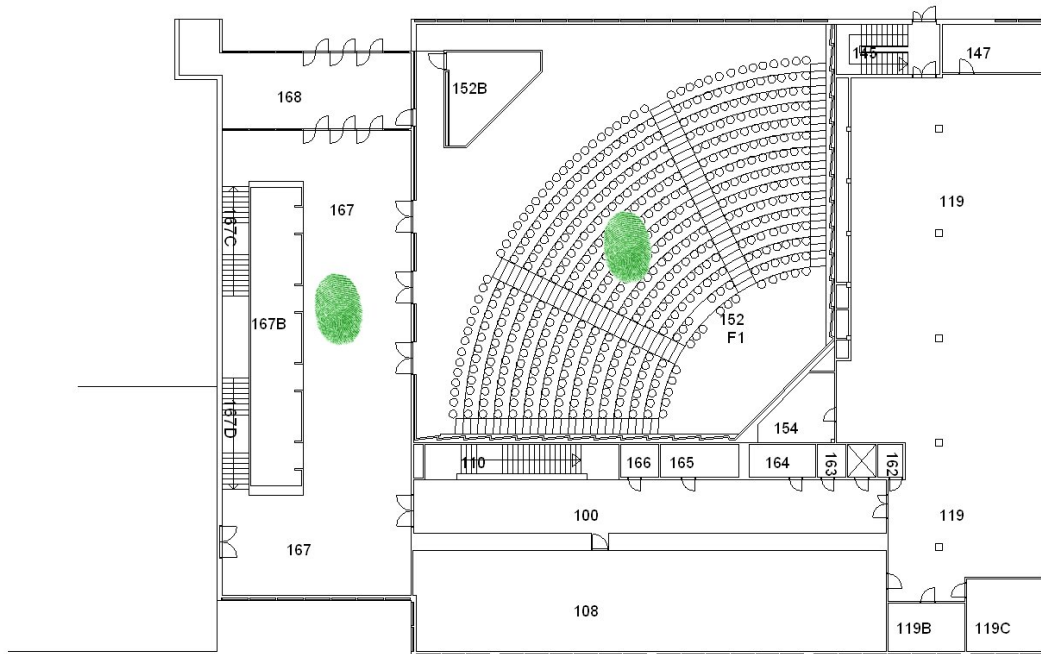
I tabellen LokTypeFunk ser vi en enkel fordeling av funksjoner på de forskjellige typene lokasjoner. For eksempel kan rom ha funksjon som kontor, grupperom, printerrom osv. Bygninger kan være for eksempel bibliotek, kantine eller lignende. Enkelte funksjoner kan også innehas av forskjellige typer lokasjoner. Hvis man ikke har noen funksjon å tildele en lokasjon kan man bruke "Uspes." for uspesifisert funksjon.

Brukertyper og brukergrupper er definert etter behov. For prototypen ble det lagt inn noen enkle brukergrupper og brukertyper for å demonstrere dette. Brukertypen student har for eksempel brukergruppene informatikk, kybernetikk, data og elektro underlagt seg. Det vil si at hvis man er student klassifiseres man etter hvilken linje man går på. Dette kan også tilpasses til det behovet man har. Hvis man for eksempel skulle implementere et system for kommunalt ansatte kunne man spesifisert en inndeling etter avdeling og stilling, eller andre former grupperinger som kunne vært hensiktsmessig.

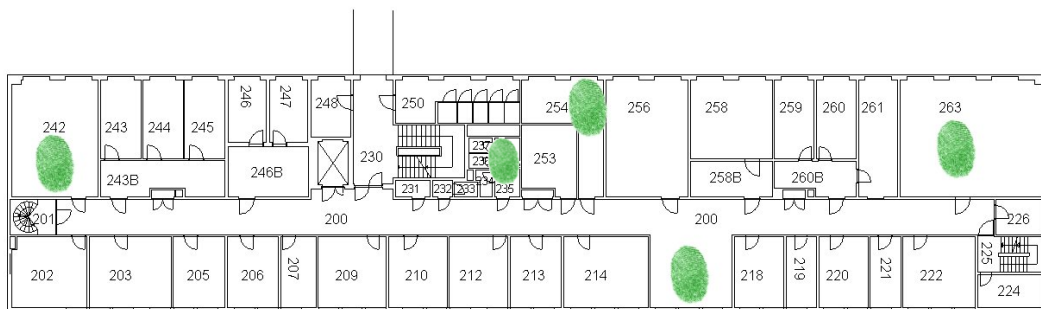
De tre tabellene InfoArt, Kontekst og RelType angir henholdsvis hvilken type informasjon man har lagt inn for en lokasjon, en type kontekst en bruker benytter systemet til, og hvilken type relasjon en lokasjon har til en annen. Disse kan redefineres fra en implementasjon til en annen for tilpasses det aktuelle formålet.

## E Fingerprint lokasjoner

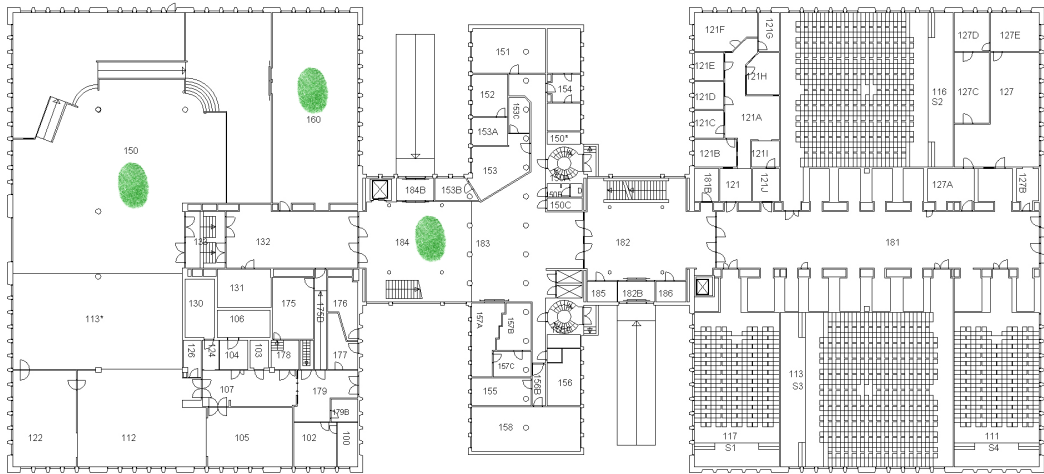
Planskissene viser en oversikt over fingerprints plassert i de tre bygningene som hovedsaklig ble benyttet til testing av systemet, Realfagbygget, IT-bygget og Sentralbygget.



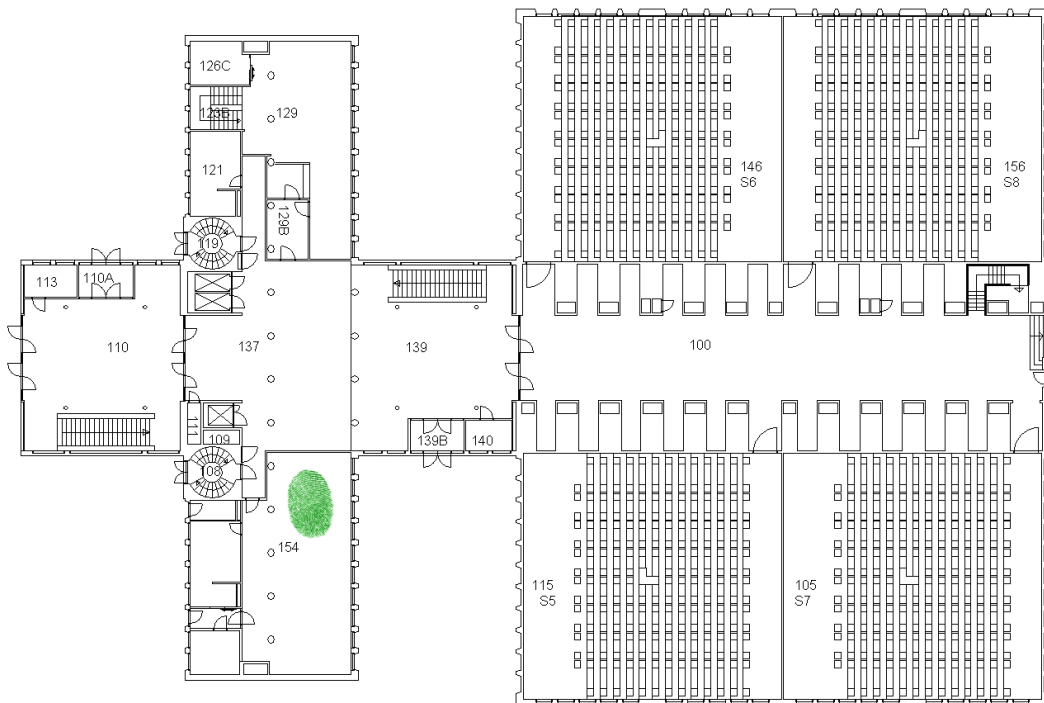
Figur 66: Fingerprint for posisjonering i IT-bygget, 1.etasje.



Figur 67: Fingerprint for posisjonering i IT-bygget, 2.etasje.



Figur 68: Fingerprint for posisjonering i Sentralbygget Syd, 1.etasje.



Figur 69: Fingerprint for posisjonering i Sentralbygget midtparti, 1.etasje.



## F - Spørreskjema

1.	Har du benyttet en lignende tjeneste eller system før?	Ja / nei							
2.	Hvilke tjenester (funksjoner) benyttet du?								
	<table border="1"> <tr> <td>Søk</td> <td>annotering</td> <td>webtjeneste</td> <td>huskeliste</td> <td>event</td> <td>buddy</td> <td>kart</td> </tr> </table>	Søk	annotering	webtjeneste	huskeliste	event	buddy	kart	
Søk	annotering	webtjeneste	huskeliste	event	buddy	kart			
3.	Hvilke tjenester (funksjoner) syntes du var nyttige?								
	<table border="1"> <tr> <td>Søk</td> <td>annotering</td> <td>webtjeneste</td> <td>huskeliste</td> <td>event</td> <td>buddy</td> <td>kart</td> </tr> </table>	Søk	annotering	webtjeneste	huskeliste	event	buddy	kart	
Søk	annotering	webtjeneste	huskeliste	event	buddy	kart			
4.	Har du noen kommentarer til sporingsfunksjonen?								
5.	Har du noen kommentarer til søk funksjonen?								
6.	Har du noen kommentarer til buddy funksjonen?								
7.	Har du noen kommentarer til huskeliste funksjonen?								
8.	Har du noen kommentarer til annoteringsfunksjonen?								
9.	Har du noen kommentarer til event funksjonen?								
10.	Har du noen kommentarer til kart funksjonen?								
11.	Har du noen kommentarer til webtjeneste funksjonen?								
12.	Var det noen av funksjonene du gjerne ville skulle fungert annerledes?	Ja / nei							

Figur 72: Spørreskjema for testing side 1

a.	Hvis ja, hvilken/hvilke, og hva burde vært annerledes?	
13.	Var det tjenester (funksjoner) du savnet som du kunne tenkt deg?	Ja / nei
a.	Hvis ja, hvilke?	
14.	Var det informasjon du savnet som du synes burde være tilgjengelig?	Ja / nei
a.	Hvis ja, hva?	
15.	Var det ting som var vanskelig å forstå?	Ja / nei
a.	Hvis ja, hva?	
b.	Hvis ja, hvordan kunne dette blitt gjort mer forståelig?	
16.	Var funksjonene enkle å bruke?	Ja / nei
a.	Hvis nei, hva var vanskelig å bruke, og hvorfor? (Søk, annotering, huskeliste, webtjeneste, event, buddy, kart)	
17.	Var det greit å navigere rundt i informasjonen?	Ja / nei
a.	Hvis nei, hvordan kunne det gjøres bedre?	
18.	Ville du lært deg å kjenne campus raskere med en slik tjeneste?	Ja / nei
19.	Er dette noe du ville brukt som ny student?	Ja / nei

Figur 73: Spørreskjema for testing side 2

a.	Hvorfor/hvorfor ikke?	
b.	Hvis nei, hva skulle til for at du ville brukt det?	
20.	Er dette noe du tror du ville fortsatt å bruke som senior student?	Ja / nei
a.	Hvorfor/hvorfor ikke?	
b.	Hvis nei, hva skulle til for at du ville brukt det?	
21.	Tror du de fleste nye studenter ville syntes dette systemet var nyttig og hjulpet dem å finne fram?	Ja / nei
a.	Hvorfor/hvorfor ikke?	
22.	Ser du nytte i dette systemet utover bare det å finne fram og få informasjon om bygninger?	Ja / nei
a.	Hvis ja, hva?	

Figur 74: Spørreskjema for testing side 3

23.	Ville du syntes en veiviserfunksjon var nyttig?	Ja / nei
24.	Ville du syntes en abonnementsfunksjon var nyttig?	Ja / nei
25.	Ville du syntes en funksjon som lot deg sende annotasjoner til andre var nyttig?	Ja / nei
26.	Ville du syntes en funksjon som sa fra om venn/kollega i nærheten var nyttig?	Ja / nei
27.	Ville du syntes en integrert chat funksjon var nyttig?	Ja / nei
a.	Er det andre kommunikasjonsformer du kunne tenkt deg?	
28.	Føler du det hadde vært behov for å kunne trykke på kartet og få info om stedene du trykket på?	Ja / nei
29.	Ville du syntes det var nyttig å kunne automatisk få opp forelesningsnotater, fagets hjemmeside, faglige meldinger og lignende når du kom inn i et auditorium hvor du hadde forelesning i faget?	Ja / nei
30.	Hvordan hadde du sett på det å få viktige meldinger sendt til deg gjennom et slikt system, likt It:s Learning?	

Figur 75: Spørreskjema for testing side 4