

IMT2243 – Systemutvikling

Aladdin AS

04HBINFA
Anja Bekkelien
Kyle Arumugam

1. Forprosjekt	4
1.1. Mål og rammer	4
1.1.1. Bakgrunn	4
1.1.2. Prosjekt mål	5
1.1.3. Rammer	5
1.2. Omfang	6
1.2.1. Oppgavebeskrivelse	6
1.3. Prosjektorganisering	8
1.3.1. Ansvarsforhold	8
1.3.2. Øvrige roller og bemanning	8
1.4. Planlegging, oppfølging og rapportering	8
1.4.1. Valg av utviklingsmodell	8
1.4.2. Krav til statusmøter og beslutningspunkt	9
1.5. Risikoanalyse	9
1.5.1. Kritiske suksessfaktorer	9
1.5.2. Risikoevaluering	9
1.6. Kvalitetssikring	10
1.6.1. Organisering av kvalitetssikring	10
1.6.2. Kvalitetssikring av kritiske suksessfaktorer	10
1.7. Gjennomføring / GANTT-Skjema	11
2. Kravspesifikasjon	12
2.1. Bakgrunn	12
2.2. Kort om krav til systemet	12
2.3. Brukerbeskrivelse	13
2.3.1. Omgivelser	13
2.3.2. Systemets brukere	13
2.3.3. Aspekter omkring livssyklus	13
2.4. Detaljert Kravspesifikasjon	14
2.4.1. Use Case Diagram	14
2.4.2. Overordnet Use Case beskrivelse	15
2.4.3. Detaljert Use Case beskrivelse	19
2.4.4. Sekvensdiagram	22
2.5. Software Design Begrensninger	24
2.5.1. Software standarder og språk	24
2.5.2. Software kommunikasjonsstandarder og grensesnitt	24
2.5.3. Database	24
2.5.4. Operativsystem	24
2.5.5. Hardware Design Begrensninger	24

2.6. Aspekter omkring livssyklus.....	25
2.6.1. Dokumentasjon	25
2.6.2. Krav til support, service og vedlikehold.....	25
2.6.3. Krav til utvidelse	25
2.7. Aspekter rundt installasjon	25
2.7.1. Hardware Installasjon	25
2.7.2. Opplæring	25
3. Design.....	26
3.1. Arktekturvudering	26
3.1.1. Systemorganisering	26
3.1.2. Kontrollmekanismer	28
3.1.3. Designskisse.....	29
3.2. Objekt Orientert Design	30
3.2.1. Overordnet systemmodell	30
4. Test Plan	32

1. Forprosjekt

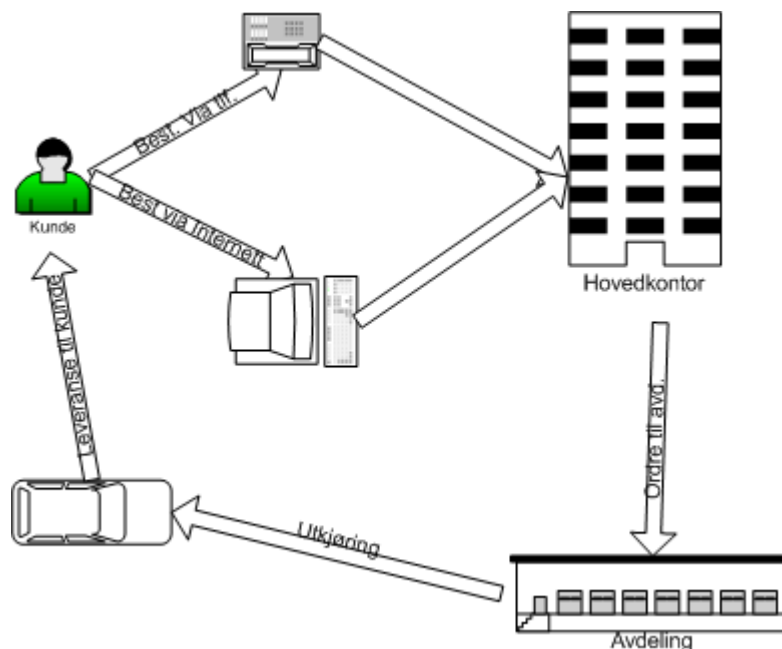
1.1. Mål og rammer

1.1.1. Bakgrunn

Aladdin AS er en fast food kjede med fem avdelinger, Oslo, Gjøvik, Lillehammer, Hamar og Ålesund. Kjeden ble etablert for fire og et halvt år siden og etter at de har hatt suksess i en lang periode, har de merket en relativt stor nedgang i salget.

De har funnet ut at det at de ikke tilbyr bringing av mat eller at de ikke har nettsider hvor kunder enkelt kan finne oppdaterte menyer, gjør at de taper en god del kunder til fordel for deres konkurrenter. Dette ønsker de å gjøre noe med.

I august har kjeden fem års jubileum og i den sammenhengen ønsker de å lansere hjemkjøring og et nettsted med bestillingsmuligheter.



Slik ser de for seg at systemet skal fungere.

1.1.2. Prosjektmål

Som et ledd i prosessen med å endre firmaets profil utad ønsker Aladdin AS å innføre et online bestillingssystem med tilsvarende levering hjemme hos kunden. Dette vil effektiviserer driften betraktelig, siden mye arbeid som før ble gjort manuelt nå automatiseres. Antall bestillinger skal øke med rundt 30 %, og på sikt vil deres markedsandel øke med 25%. Man ønsker også å satse på en bredere kundegruppe, blant annet bedriftskunder som ønsker å bestille mat for et større antall mennesker. Systemet skal også ha mulighet for at store bestillinger kan bestilles lang tid i forveien.

Resultatet av prosjektet vil være et komplett bestillingssystem med tilhørende kundedatabase og fakturasystem. Systemet vil også ta seg av all administrasjon av bestillinger.

1.1.3. Rammer

Prosjektet har en absolutt tidsfrist 18.08.2006 som ikke kan overskrides da dette er datoen for 5-års jubileet. Av kostnadmessige årsaker vil man benytte et allerede eksisterende kunderegister tidligere utviklet av vår bedrift. Da dette avviker noe fra kundens krav må det brukes tid på å tilpasse dette til ordresystemet. Fakturasystemet skal også kjøpes inn.

Prosjektet har en kostnadsramme på 250 000 NOK.

Det er i tillegg viktig systemet innehar en høy grad av brukervennlighet. Det skal ikke kreves noen form for data-kompetanse utover det helt grunnleggende, og systemet skal kunne brukes etter en times opplæring.

1.2. Omfang

1.2.1. Oppgavebeskrivelse

Prosjektet skal ende opp i et nettbasert bestillingssystem bestående av de fire hovedmodulene hjemmeside, ordresystem, kunderegister og fakturasystem.

Kunden skal via nettsidene enkelt kunne bestille varer fra Aladdin AS via ordresystemet som integreres i nettsidene. Ved førstegangs bestilling må kunden gjennom en registreringsprosess hvor navn, adresse og kontaktinformasjon lagres i kunderegisteret.

Deretter vil han kunne sette sammen sin bestilling ved å velge fra en meny. Nettsidene skal også inneha funksjonalitet for å gjøre større bestillinger på en enkel måte. Ordresystemet vil automatisk sortere alle bestillinger basert på adressen kunden oppgir, og sender bestillingene videre til avdelingen som er ansvarlig for det aktuelle området bestillingen kommer fra. Systemet skal også ta seg av bestillinger/kjøp som kommer via telefon/kassen. Ved telefonbestillinger må de ansatte manuelt registrere ordren i et grensesnitt i systemet, mens ved kjøp i kassen registreres informasjon automatisk.

Hver avdelings kjøkken vil ha en skjerm som viser bestillingene som kommer inn. Bestillingene vil også overføres til en PDA som sjåføren har med seg i bilen som viser all nødvendig informasjon for levering av varen, i tillegg til å regne ut den mest lønnsomme reiseruten mellom de forskjellige leveringsstedene.

Ordresystemet skal også kommunisere med avdelingenes allerede eksisterende varelagersystem og kassasystem.

Varelageret har en elektronisk oversikt over varebeholdningen som automatisk oppdateres når det foretas bestillinger av råvarer. De ansatte gjør med jevne mellomrom opptelling i varelageret og justerer oversikten. Ordresystemet skal utformes slik at når nye bestillinger kommer inn, enten via nettsiden, i kassen eller via telefon, vil råvarene som trengs trekkes fra varelagerets beholdning. Dersom varelageret skulle gå tomt for en råvare vil dette gjenspeiles på nettsidene ved at rettene som inneholder denne råvaren blir umulig å bestille.

I tillegg skal systemets administrator ha mulighet for å oppdatere websiden for å endre priser, meny osv. Han skal også kunne hente fram statistikk om salg og inntekter.

1.3. Prosjektorganisering

1.3.1. Ansvarsforhold

Prosjektleder – Erik Mykland

Arbeidsgiver og kontaktperson – Ali Bardia

1.3.2. Øvrige roller og bemanning

Programmering: George Best, Harry C. Cobolt og Lothar Matthäus

Database: Ian Rush

Design: Shakira Nilsson

Test- og driftansvarlig: Roy Keane

I tillegg til de overnevnte, skal det også være en testgruppe basert på aktuell kundegruppe. For applikasjoner som brukes innad i Aladdin As, vil deres ansatte være med i testgruppen. Nettstedets funksjonaliteter og brukervennlighet vil bli testet av en ekstern kundegruppe.

1.4. Planlegging, oppfølging og rapportering

1.4.1. Valg av utviklingsmodell

Vi har valgt å bruke en inkrementell utviklingsmodell ettersom prosjektet enkelt kan deles inn i atskilte inkremitter. RUP ble også vurdert som en aktuell modell ettersom den vil gi prosjektet både en ryddig struktur og en grundig dokumentasjon.

Noen av medlemmene i prosjektgruppen har også tidligere erfaring fra prosjekter hvor RUP har blitt benyttet. Imidlertid anser vi metoden for å bli for ressurskrevende for et prosjekt av denne størrelsen, og den inkrementelle metoden gir oss tilstrekkelig fleksibilitet med hensyn til endringer underveis.

Vi vil innenfor to av inkrementene benytte oss av gjenbruksmodellen, siden vi allerede har et godt kunderegister som tidligere har blitt utviklet hos oss, og det finnes også gode fakturasystem i markedet. Ved å bruke gjenbruksmodellen, vil vi spare tid og ressurser.

XP som utviklingsmodell har vi valgt bort i dette prosjektet, da dette vil kreve mer ressurser og dermed øke kostnadene. Dessuten er rammene rundt systemet ganske klare og det vil nødvendigvis ikke bli mye prøving og feiling.

Men det er heller ikke aktuelt med fossefallsmodellen, da kunden har tydelig understreket at de ønsker å se og prøve systemet underveis og komme med innspill.

1.4.2. Krav til statusmøter og beslutningspunkt

Statusmøte før hvert inkrement og da skal også forrige inkrement oppsummeres. Da skal gruppen også evaluere om prosjektet holder den tidsrammen som er beskrevet, og at systemet overholder de krav som er satt.

1.5. Risikoanalyse

1.5.1. Kritiske suksessfaktorer

For at prosjektet skal ha suksess må følgende krav være oppfylt:

- Levering til rett tid
- Systemet må være lett å lære

1.5.2. Risikoevaluering

Følgende faktorer kan føre til at prosjektet ikke blir fullført:

- Bedriften har ikke penger nok til å fortsette utviklingen

Her må prosjektgruppen ta en grundig evaluering av bedriften før prosjektstart, for å unngå at prosjektet ikke fullføres pga. økonomiske grunner

Følgende faktorer kan bidra til at prosjektet blir forsinket:

- Tap av nøkkelpersonell i prosjektgruppen.
- Problemer med eksterne kilder under installasjon
- Læring av eksterne systemer

Prosjektgruppen må sørge for at alle kontrakter med eksterne kilder overholdes. Eksterne systemer skal være lette å bruke, og kunden skal ha en viss kjennskap til de forskjellige systemene.

Tap av personell er en lav risiko, men kan føre til en liten forsinkelse når nytt personell skal hentes inn. For å minimisere den risikoen ytterligere, skal prosjektgruppen ha en tilstrekkelig og god dokumentasjon over alt som har blitt utført under utviklingsperioden.

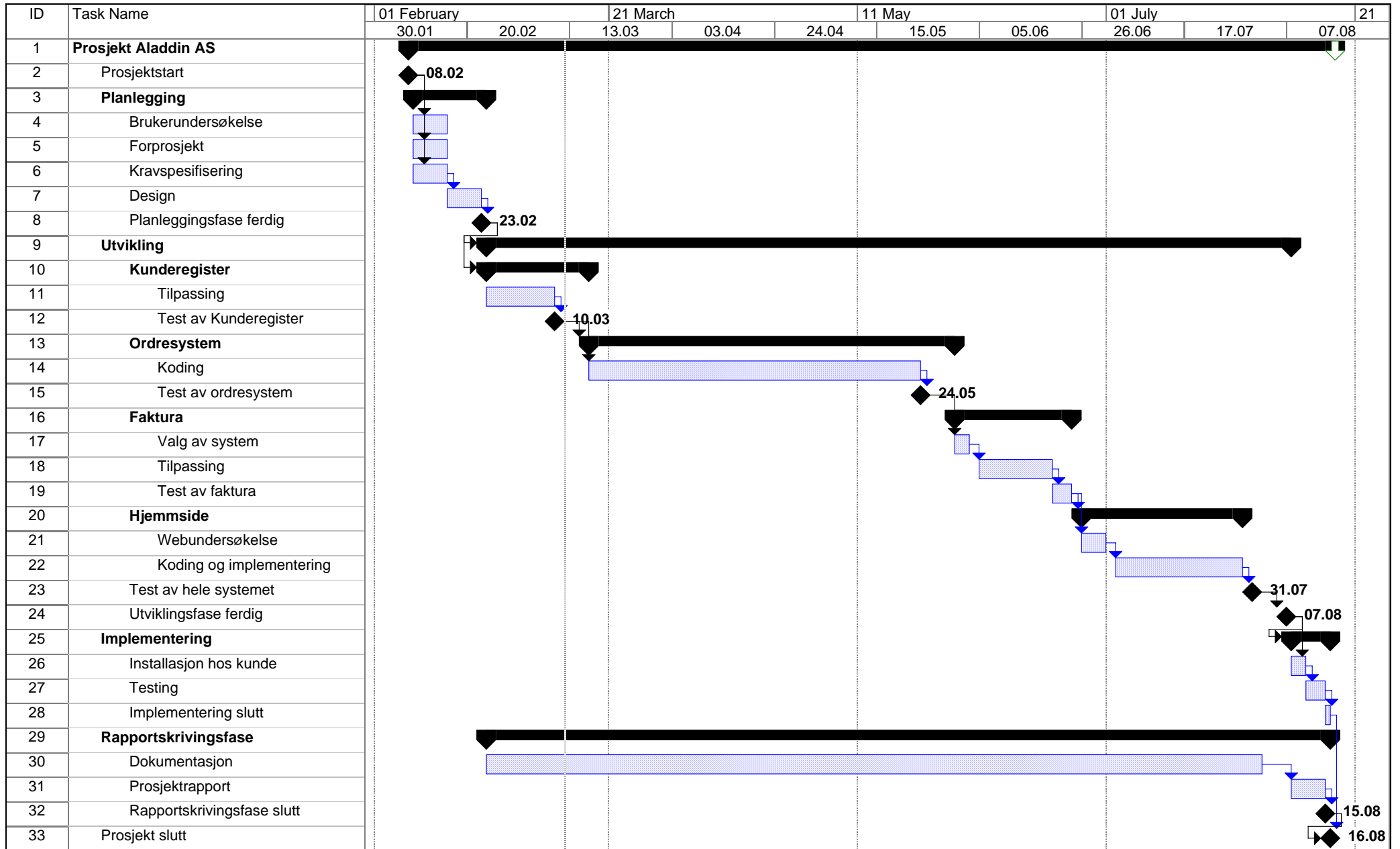
1.6. Kvalitetssikring

1.6.1. Organisering av kvalitetssikring

Alle dokumenterer underveis og dette blir levert til prosjektleder ved hvert statusmøte.

1.6.2. Kvalitetssikring av kritiske suksessfaktorer

For å unngå at prosjektet ikke blir fullført, må vi sikre oss om at systemet ikke blir for tungt å bruke eller lære. Systemet skal kunne implementeres/installeres uten problemer for å unngå forsinkelser.



Project: Aladdin AS
Date: Sun 12.03.06

Task

Split

Progress



Milestone

Summary

Project Summary



External Tasks

External Milestone

Deadline



2. Kravspesifikasjon

2.1. *Bakgrunn*

Bakgrunn for kravspesifikasjon og prosjekt er å finne i forprosjektet under punkt 1.1 mål og rammer.

2.2. *Kort om krav til systemet*

Prosjektet skal ende opp i et ordresystemet for bestilling av mat og det viktige i hele systemet er automatisering. Systemet skal ha disse funksjonene og modulene:

- Automatisert bestillingssystem
- Sentralisert ordremottak
- Sentralisert kunderegister
- Kommunisere med eksisterende kasse-, varelager- og regnskapssystem
- Brukervennlig administrasjonsdel
- Interaksjon med brukervennlig webside
- Kommunisere med nytt sjåførsystem

Sjåførene skal i bilen ha tilgjengelig en PDA som brukes for å lette navigasjonen ved at den regner ut de mest lønnsomme reiserutene. Det skal kjøpes inn et sjåførsystem som skal ta seg av dette. Aladdinsystemet skal kunne sende og motta data fra dette systemet.

2.3. Brukerbeskrivelse

2.3.1. Omgivelser

Systemet skal tilknyttes kassasystemet, varelagersystemet, skjerm for visning av bestillinger og sjåførenes pda-er ved hver avdeling. Disse skal kommunisere med hverandre over en trådløs nettforbindelse, i tillegg til å kommunisere med hovedkontoret hvor hoveddelen av systemet skal installeres.

2.3.2. Systemets brukere

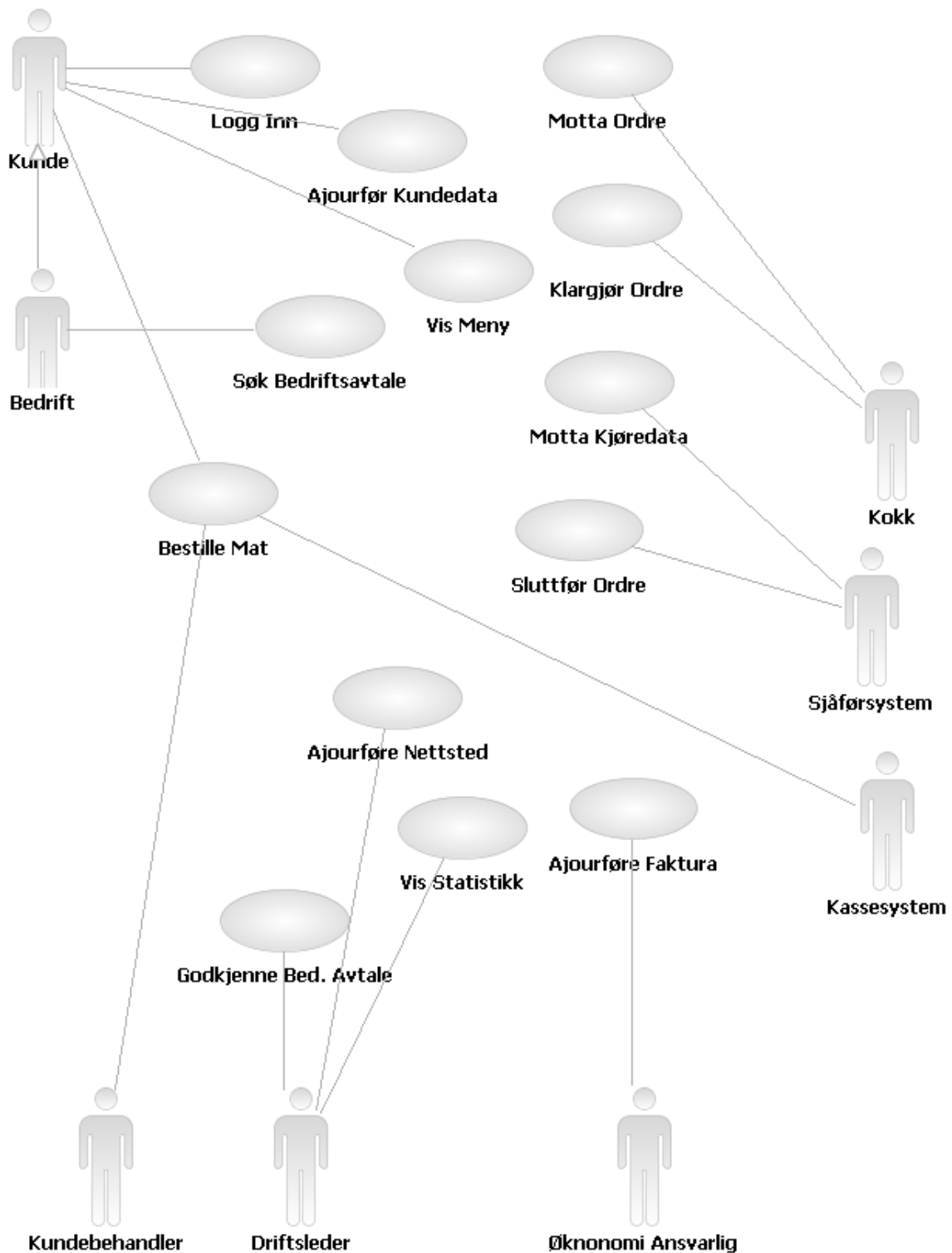
Brukerne av systemet vil ha variert bakgrunn og kompetanse, og det vil bli behov for noe opplæring. Vi vil stå for opplæring av personell og IT-ansvarlig ved hovedkontoret, og disse vil i sin tur lære opp de lokale avdelingene. For brukerne av systemets nettsider vil det foreligge en enkel dokumentasjon som beskriver bestillingsprosessen.

2.3.3. Aspekter omkring livssyklus

Vedlikehold og reparasjon av systemet hos de forskjellige avdelingene vil utføres av våre lokale samarbeidspartnere siden dette vil gi en så rask service som mulig. Vi vil selv utføre vedlikehold og eventuelle oppdateringer ved hovedkontoret. Systemet skal lages slik at det enkelt kan flyttes dersom en avdeling bytter lokaler.

2.4. Detaljert Kravspesifikasjon

2.4.1. Use Case Diagram



Kasse- og sjøførsystem er eksterne system som Aladdinsystemet skal samarbeide med.

Kunde

En kunde kan være privatperson eller bedriftskunde.

Kokk

Kokken får opp nye ordrer via casen Motta Ordre og når ordren er ferdig, sender han ordren videre til ved velge Klargjør ordre. Ordrene blir automatisk pushed ned til kokkens skjerm.

Sjåførsystem

Sjåførsystem er et ekstern system som Aladdinsystemet pusher data til. Dette systemet har ansvaret for å kommunisere med PDA'ene og til slutt sende tilbake en informasjon om ordrens status.

2.4.2. Overordnet Use Case beskrivelse

Use case:	Bestille mat*
Aktør:	Kunde
Mål:	Aktør skal legge igjen en ordre
Normal Hendelsesflyt:	Systemet tar i mot ordren og viser aktøren en kvittering
Variasjoner:	

Use case:	Motta Ordre*
Aktør:	Kokk
Mål:	Aktør mottar aktuelle ordrer forløpende som de kommer inn
Normal Hendelsesflyt:	Systemet sjekker og sender ordren til aktuell avdeling.
Variasjoner:	

* Begge casene er beskrevet detaljert under punkt 2.5.3

Use case:	Logge inn
Aktør:	Kunde
Mål:	Logge inn på systemet
Normal Hendelsesflyt:	<ol style="list-style-type: none"> 1. Aktøren skriver inn brukernavn/passord og trykker på 'Logg inn' 2. Sjekker om kunden er registrert 3. Systemet logger aktøren inn.
Variasjoner:	<p>Aktør er ikke registrert</p> <p>Server er nede</p>

Use case:	Motta kjøredata
Aktør:	Sjåførsystem
Mål:	Overføre informasjon om levering av en ordre til sjåførsystemet.
Normal Hendelsesflyt:	<p>Når en ordre er klar for utkjøring sender ordresystemet en forespørsel til sjåførsystemet om hvilke sjåførere som er ledige.</p> <p>Sjåførsystemet sender tilbake en oversikt over disse, og ordresystemet overfører så ordren til en av dem.</p>
Variasjoner:	Ingen ledige sjåførere eller ingen kontakt med kjøresystem.

Use case:	Ajourføre nettsider
Aktør:	Driftsleder
Mål:	Opprette/oppdatere innholdet på bedriftens nettsider
Normal Hendelsesflyt:	<p>Det er kun driftslederen ved hovedkontoret som har anledning til å oppdatere/endre nettsidene.</p> <p>Han/hun logger seg først på websystemet, og får så tilgang til et grensesnitt hvor han/hun kan gjøre endringer i menyer, priser og lignende.</p> <p>Dette skal skje uten å måtte benytte HTML tager. Før endringene lagres permanent skal det være mulig å forhåndsvise nettsiden.</p>
Variasjoner:	Aktør får feil meld pga. mangelfull utfylling av skjemaer eller det kan være driftsproblemer.

Use case:	Ajourføre faktura
Aktør:	Økonomiansvarlig
Mål:	Sjekk en fakturas status, i tillegg til å sende ut fakturaer og purringer.
Normal Hendelsesflyt:	<p>Økonomiansvarlig ber fakturasystemet om å vise nye fakturaer. Han velger hvilke som skal sendes ut, og ber systemet skrive ut disse.</p> <p>Han kan også be om å få se fakturaer som ikke er betalt innen fristen, og sende ut purringer på disse.</p>
Variasjoner:	Det finnes ingen nye eller ubetalte faktura.

Use case:	Søk bedriftsavtale
Aktør:	Bedriftskunde
Mål:	Søke om bedriftsavtale for å kunne få rabatt
Normal Hendelsesflyt:	<p>Bedriften fyller ut og sender inn et søknadsskjema på Aladdins nettsider.</p> <p>Skjemaet overføres til websystemet, som sjekker bedriften opp mot Brønnøysundregistrene og et eksternt inkassosystem, for å unngå falske søknader eller søknader fra svartelistede bedrifter.</p> <p>Dersom søknaden vurderes som reell, overføres den til administrasjonsdelen av systemet hvor driftslederen senere vil ta hånd om den.</p>
Variasjoner:	

Use case:	Godkjenne bedriftsavtale
Aktør:	Driftsleder
Mål:	Godkjenne søknader om å inngå bedriftsavtale
Normal Hendelsesflyt:	<p>Når nye søknader kommer inn skal systemet automatisk varsle driftslederen om dette. Han/hun kan da velge å få dem vist.</p> <p>Dersom søknaden godkjennes får driftslederen autogenerert en e-post til søkeren, som forteller at søknaden er godkjent. Bedriften og dens rabattordning lagres i kundedatabasen.</p>
Variasjoner:	Søknad om bedriftsavtale blir avslått.

2.4.3. Detaljert Use Case beskrivelse

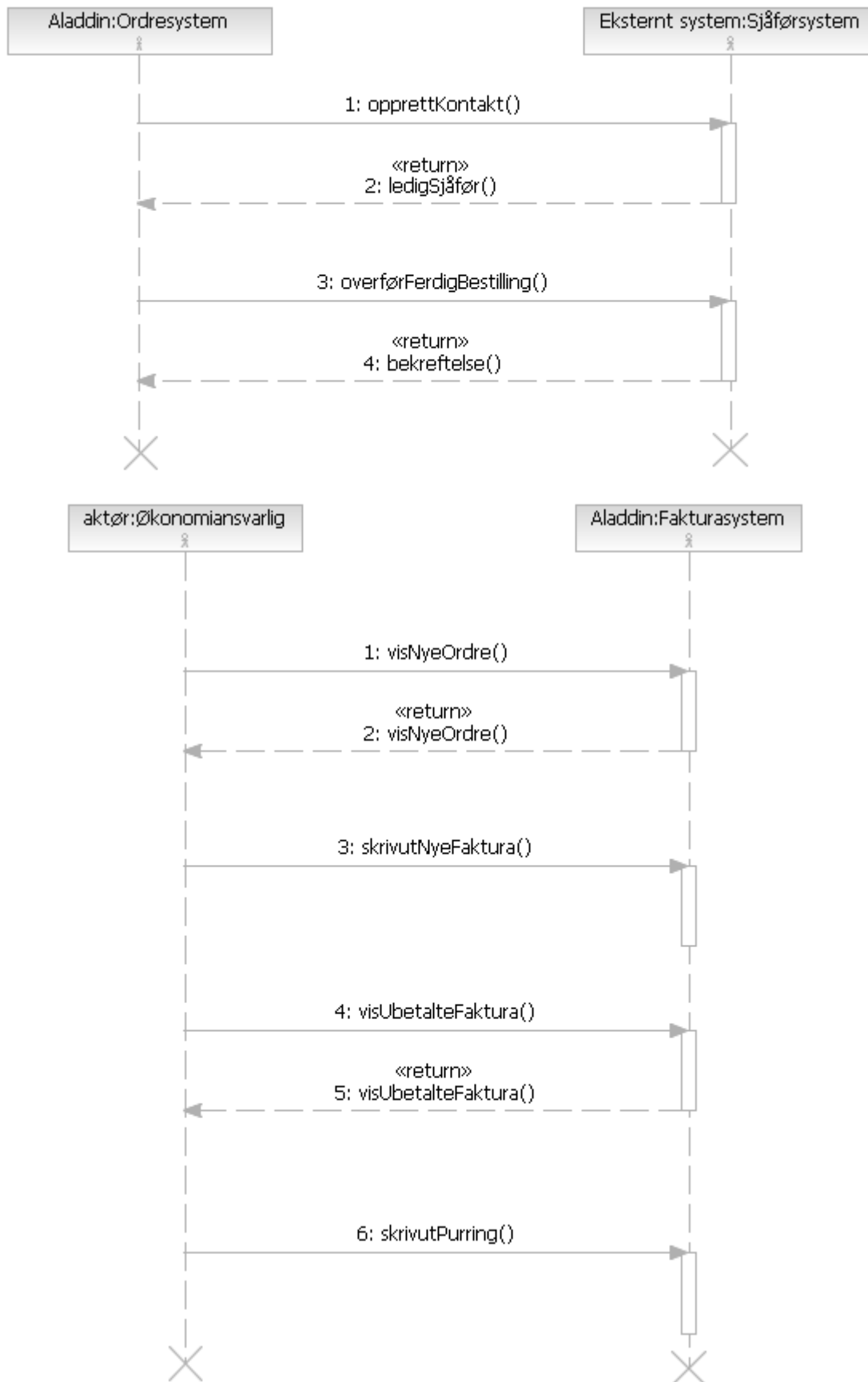
Under detaljert Use Case beskrivelse har vi valgt å skrive litt mer om casene bestille mat og motta ordre. Dette er det som kommer til å være det viktigste og det mest kritiske ved Aladdin systemet for at en ordre skal bli laget og sendt til rett avdeling slik at kunden får sine varer uten problemer.

Use case:	Bestille mat
Aktør:	Kunde
Formål:	La kunden bestille mat.
Sammendrag:	<p>Aktøren kommer inn på nettstedet og velger de rettene han/ho ønsker og legger det inn i handlekurven. For hver rett/vare som blir lagt i handlekurven, blir den sjekket mot varelageret for å se om det er mulig å levere retten/varen.</p> <p>Før ordren kan sendes, må kunden være innlogget. Når dette er gjort og ordren er sendt, får kunden opp en melding fra systemet og det vil enten være en kvittering eller en melding om hvorfor ordren ble stoppet og hva aktøren kan foreta seg.</p> <p>Kunde har også mulighet til å ringe inn sin ordre og da kommer han/ho til en kundebehandler ved hovedkontoret. Kundebehandleren legger ordren inn i systemet og kunden får oppgitt et ordrenummer.</p>
Type:	Viktig
Pre:	Aktør som ønsker å bestille og webside/tlf sentral er operativ.
Post:	Aktør har fått inn en ordre og mottatt en kvittering.
Hendelsesflyt	
Aktør	Systemrespons
1. Bruker ankommer nettsted og velger avdeling 2. Aktør legger inn en rett/vare.	
	3. Systemet sjekker om varen er gyldig. 4. Sjekk mot varelager for å se om det er mulig å levere retten/varen.
5. Aktør fjerner vare	
	6. Systemet fjerner vare fra handlekurven
7. Aktør logger inn	
	8. Systemet validerer bruker

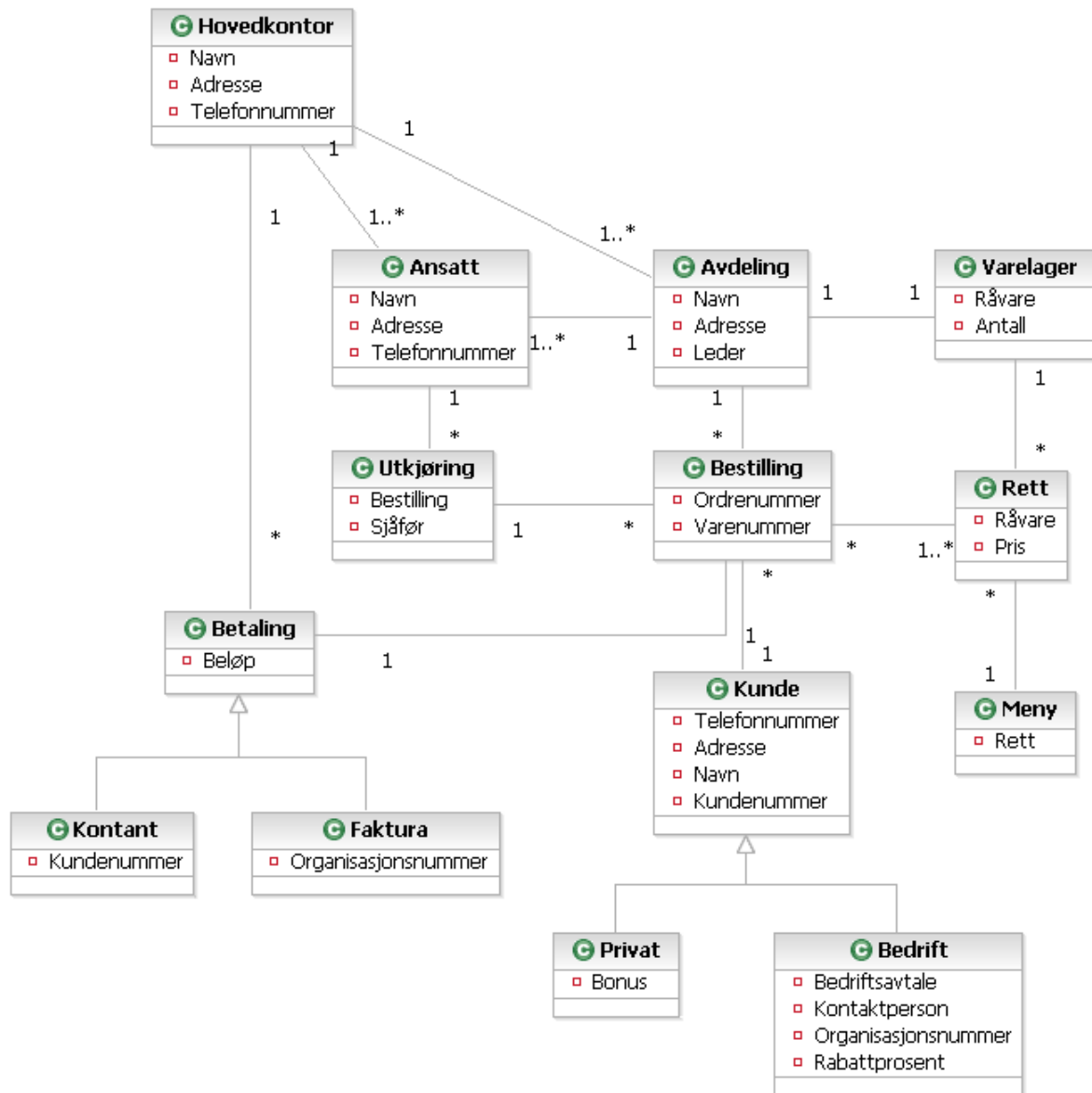
9. Aktør sender inn ordre	
	10. Systemet tar imot ordren og sender ordren til aktuell avdeling. 11. Systemet returnerer en bekreftelse
12. Aktør mottar kvittering	
Alternativ Hendelsesflyt	
<p>Webside er ute av drift og kunden har mulighet til å prøve seinere eller ringe inn bestillingen.</p> <p>Punkt 1 og 2: Her kan det også være snakk om en bestilling via telefon hvor en kundebehandler tar imot ordren og legger den inn manuelt.</p> <p>Punkt 3 og 4: Noen retter/varer kan ikke leveres da dette ikke er på lager og kunden får da beskjed om dette og kan velge andre retter/varer.</p> <p>Punkt 7 og 8: Bruker kunne ikke logge inn på grunn av feil brukernavn eller passord. Bruker kan prøve på nytt eller be om å få tilsendt nytt passord på tidligere oppgitt epost. Eller brukeren ikke er registrert og får da beskjed om dette. Bruker blir sendt til en registreringsside. Når bruker har fylt ut nødvendig informasjon, blir brukers adresse og telefonnummer sjekket. Dette for å sikre at det finnes en slik adresse og tlf.nr. Bruker får tilsendt en verifiseringskode enten pr. epost eller mobil og før dette er blitt skrevet inn av kunden kan ikke bestillingen fortsette.</p> <p>Punkt 10 og 11: Feil mellom hovedkontor og avdeling kan føre til at systemet ikke får sendt ordren og da får brukeren beskjed om dette og kan velge å prøve igjen.</p> <p>Under de forskjellige handlingene kan brukeren, så lenge ordren ikke er sendt, avslutte handelen.</p>	

Use case:	Motta Ordre	
Aktør:	Kokk	
Formål:	Vise aktør fortløpende nye ordrer	
Sammendrag:	Modulen behandler ordrene som blir lagt inn i ordresystemet og sjekker i forhold til kundens adresse hvilken avdeling ordren skal sendes til. Så bli ordren sendt til den aktuelle avdeling og dette vil vises fortløpende på en touchscreen ved avdelingen.	
Type:	Viktig	
Pre:	Ordrer er inn i ordresystemet	
Post:	Ordrer er sendt til de aktuelle avdelingene	
Hendelsesflyt		
Aktør	Systemrespons	
1. Aktør har en oversikt over innkommende ordrer på sin touchscreen		
	2. Use caset starter ved at det finnes en ny ordre 3. Systemet sjekker hvilken avdeling ordren skal sendes til. 4. Ordre blir sendt til aktuell avdeling automatisk 5. Systemet sender tilbake en melding om at avdeling har mottatt ordren	
6. Aktør får se den nye ordren på sin skjerm		
Alternativ Hendelsesflyt		
3) Om det ikke var mulig å finne rett avdeling, blir hovedkontoret varslet. 4) Dersom sending til avdeling var mislykket, vil systemet varsle hovedkontoret. 5) Dersom det ikke var mulig å sende en bekreftelse, vil systemet forsøke å sende på nytt og samtidig varsle de ansatte på avdelingen. Dette vil være i form av en beskjed på en skjerm.		
Ved feil mellom lokalsystem og kokken sin terminal, vil det også komme opp et varsel.		

2.4.4. Sekvensdiagram



2.4.5. Konseptuelt Klassediagram



2.5. Software Design Begrensninger

2.5.1. Software standarder og språk

Systemet utvikles hovedsaklig under C++. For å kunne kommunisere med PDA til sjåføreren benytter vi .Net teknologien til Microsoft.

2.5.2. Software kommunikasjonsstandarder og grensesnitt

Kommunikasjon innad i de enkelte avdelinger går over ethernet, mens kommunikasjon mellom avdelinger og hovedkontor bruker TCP/IP (Internet).

2.5.3. Database

Vi benytter oss av MySQL på webserveren og resterende system skal bruke MSSQL.

2.5.4. Operativsystem

Operativsystem som server og klient program skal kjøre på er Windows XP og Windows Server 2003. På webserveren velger vi Apache og Linux.

2.5.5. Hardware Design Begrensninger

Serveren må stå i ett godt ventilert rom med lav temperatur. Klient Pc-ene bør ikke stå inne på kjøkkenet da dette kan føre til varmeproblemer.

PDA må kunne tåle store temperatursvingninger og de må i tillegg være vanntette og støtsikre. Trykkskjermene må kunne tåle varme og fuktig miljø.

2.6. Aspekter omkring livssyklus

2.6.1. Dokumentasjon

Brukeren av nettsidene (kundene) vil få en enkel dokumentasjon om hvordan nettsiden skal brukes for å bestille mat. Vi vil lage en opplæringsmanual for de ansatte ved avdelingene for bruk av systemet generelt.

2.6.2. Krav til support, service og vedlikehold

Backup og vedlikehold utføres etter at avdelingene har stengt. Ved problemer med hardware/software vil hovedkontoret få hjelp av oss, mens hver avdeling vil vi kontakte lokale samarbeidspartnere for utføring av vedlikehold

2.6.3. Krav til utvidelse

Systemet skal kunne installeres hos nye avdelinger uten at dette går utover de eksisterende avdelingene, hovedkontoret og brukerne (kundene) av systemet. Systemet skal kunne være oppgraderbar slik at dette lett kan gjøres senere ved behov.

2.7. Aspekter rundt installasjon

2.7.1. Hardware Installasjon

Systemet skal installeres både ved hovedkontoret og alle avdelingene. Eksterne aktører skal ta seg av det elektriske anlegget og kobling av bredbånd. Vi skal installere hos hovedkontoret, mens lokale samarbeidspartnere skal ta seg av installasjon hos de ulike avdelingene.

2.7.2. Opplæring

Hovedkontoret, inkludert it-ansvarlig, vil få opplæring av oss. De har ansvaret for å lære opp ansatte ved de forskjellige avdelingene.

Hovedkontoret skal også gi support til avdelingene, mens vi gir support til hovedkontoret.

3. Design

3.1. Arktekturvudering

3.1.1. Systemorganisering

Vi har diskutert flere måter å organisere Aladdin systemet på. Ved å bruke repository modellen ville man ha fått sentralisert og kostnadsbesparende løsning ved at alle delte data var lokalisert ved hovedkontoret og at avdelingen *pusher* ny data oppover og hentet ned data når det er nødvendig. Backup og vedlikehold av systemet ville vært enkelt siden dette bare gjøres på en plass.

Men med en slik løsning blir avdelingen helt avhengig av hovedkontoret og om det skulle være brudd på forbindelsen, ville den eller de avdelingene være ute av drift.

For Aladdin systemet har vi valgt å gå for en distribuert systemmodell. Her vil hver avdeling ha en lokal server som tar vare på data om sin egen avdeling. Denne serveren vil også være bindeleddet med mellom hovedkontor og avdeling.

For å sende og motta data mellom avdeling og hovedkontor, kan vi bruke enten *push* eller *pull* metode.

Vi ønsker ikke å bruke *pull* for å distribuere data, siden det ikke er lett å vite når nye data vil være tilgjengelig og systemene ved avdelingene blir nødt til å kjøre sjekk mot hovedkontoret veldig ofte. Det vil være bedre bruke en *push* metode, da slipper avdelingene å lytte men ligge i dvale og vente på å bli kontaktet av hovedsystemet.

Det samme skjer når avdelingene har nye data som skal sendes til hovedkontoret. Når det er lagt inn nye ordrer lokalt over disk, vil disse bli pushet opp til hovedkontoret når dette er nødvendig

Dersom det skulle være problemer med forbindelsen mellom hovedkontor og avdeling, vil avdelingen fremdeles kunne være operativ ved hjelp av den lokale serveren og kunne ta imot bestillinger fra lokale kunder i avdelingen.

Den lokale serveren vil motta informasjon og endringer fra hovedkontoret fortløpende og slik vil den til enhver tid være oppdatert. De nye lokale ordrene vil bli lagret på serveren og all nye data vil med valgt intervall synkronisert med hovedkontoret.

Slik vil det fremdeles være lett å styre alle avdelingene fra hovedkontoret ved at nye priser, nye produkter, endringer i produktene og kampanjer lett kan bli pushet ned til avdelingene. Samtidig blir det enkelt å for hovedkontoret å sjekke statusen til de forskjellige avdelingene og siden all data blir samlet på en plass, trenger man bare å ta backup av de sentraliserte dataene. De lokale serverne bør også ha en backupløsning, men dette kan være av en enklere automatisert løsninger.

For at det skal minst mulig trafikk mellom avdelingene og hovedkontor, vil kun data sendes og mottas. Det som trenges av programvare vil ligge lokalt på klient maskinene. Dette for å unngå belastning på linjene og for at både kunder og ansatte skal få slippe å vente lenge på å få gjort det de ønsker å gjøre på systemet.

3.1.2. Kontrollmekanismer

Systemets kontrollmekanismer avgjør kontrollflyten og hvordan subsystemene kommuniserer med hverandre. Det finnes to generelle kontrollmodeller: sentralisert og hendelsesbasert. I den sentraliserte modellen fungerer ett av subsystemene som en kontroller med ansvar for å starte og stoppe de andre, ofte sekvensielt.

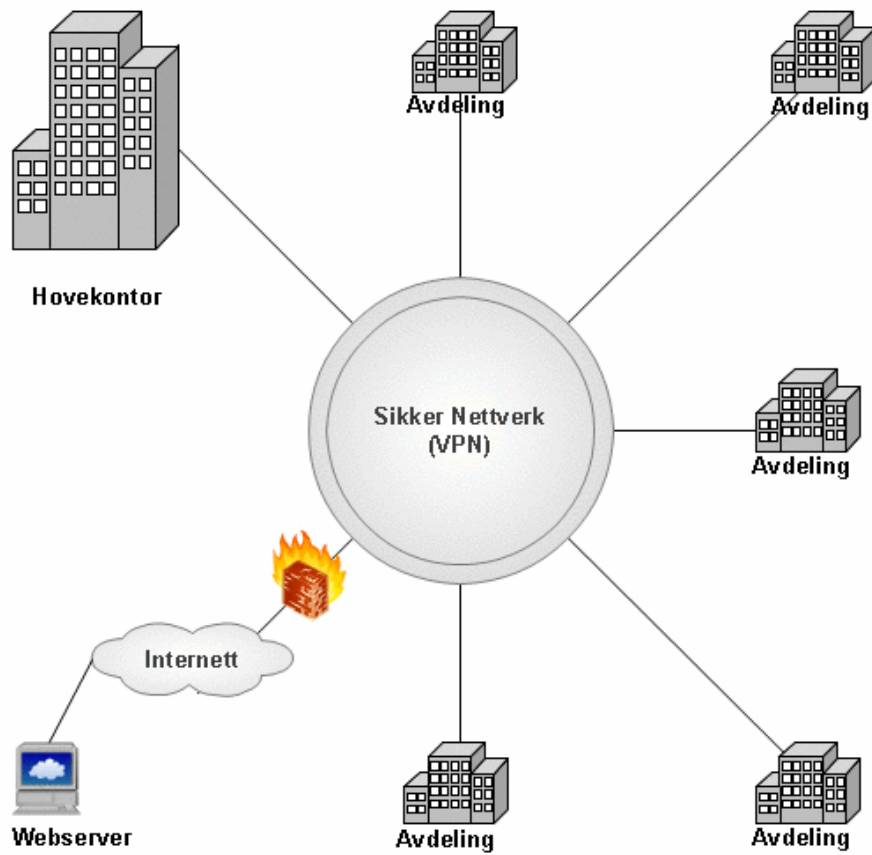
Vårt system skal imidlertid bygges opp som et hendelsesbasert kontrollsystem. Dette innebærer mye av det samme som den sentraliserte modellen, men i stedet for kontrolleren vil det være ytre hendelser som avgjør hvilket subsystem som skal kjøres.

Dette vil være den mest hensiktsmessige løsningen, ettersom kundenes bestillinger "genererer" hendelser. Ordresystemet som er installert hos hovedkontoret vil fungere som en hendelseskontroller, som sender hendelsene videre til "systemene", avdelingskontorene. Dette vil foregå etter prinsippene i broadcastmodellen, hvor hendelseskontrolleren bruker et register over subsystemer og hendelser som er av interesse for disse, for å distribuere hendelsene. Kriteriet som avgjør et subsystems interesse er bestillingens adresse.

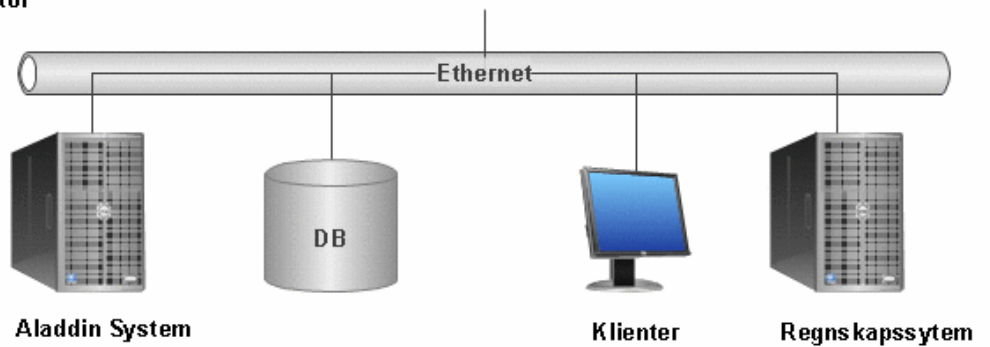
Bruk av denne modellen vil gjøre det enkelt å legge til eventuelle nye avdelinger til systemet ettersom dette i hovedsak bare vil innebære å registrere dennes hendelser som er av interesse hos hendelseskontrolleren.

Figur på neste side viser hvordan hele nettverket vil se ut.

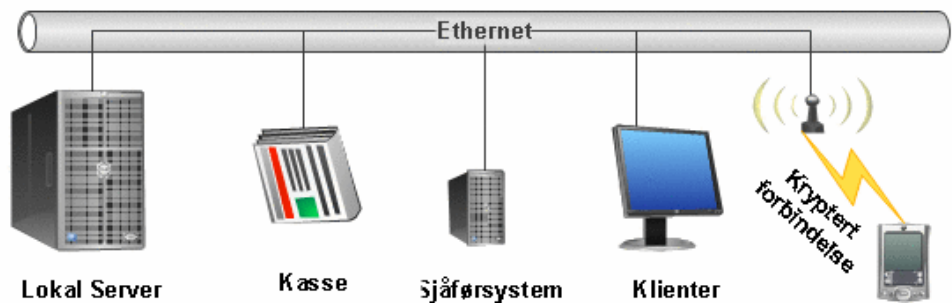
3.1.3. Designskisse



Hovedkontor

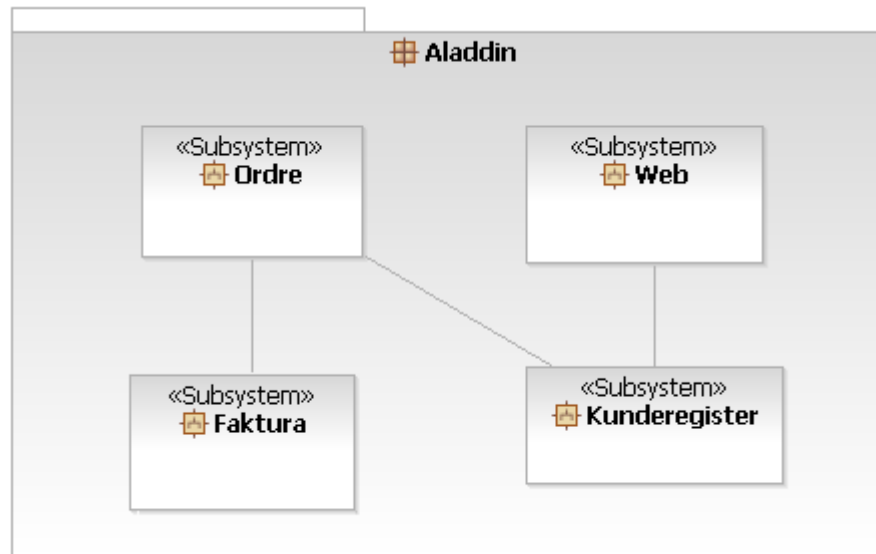


Avdeling



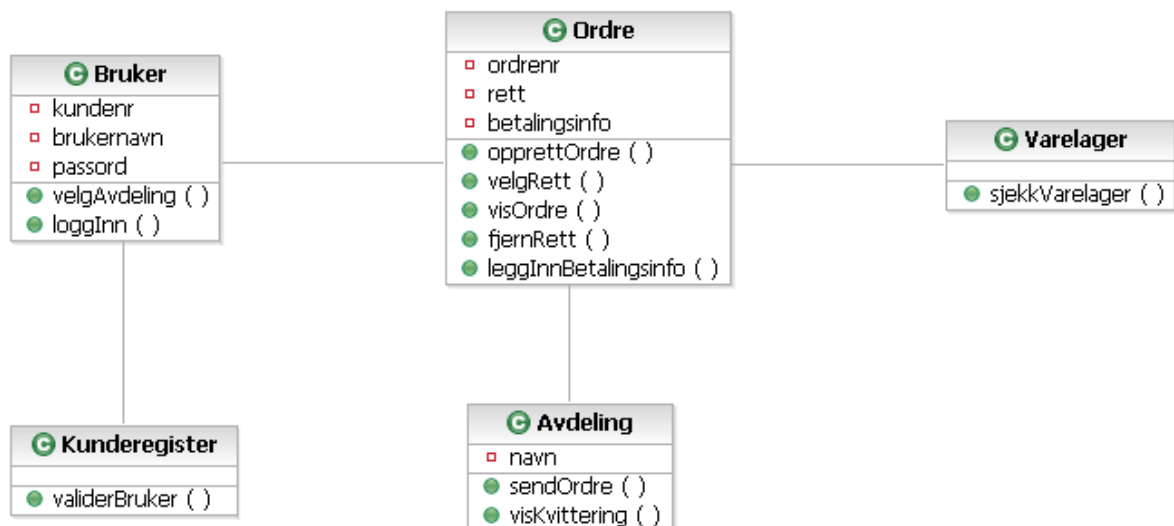
3.2. Objekt Orientert Design

3.2.1. Overordnet systemmodell

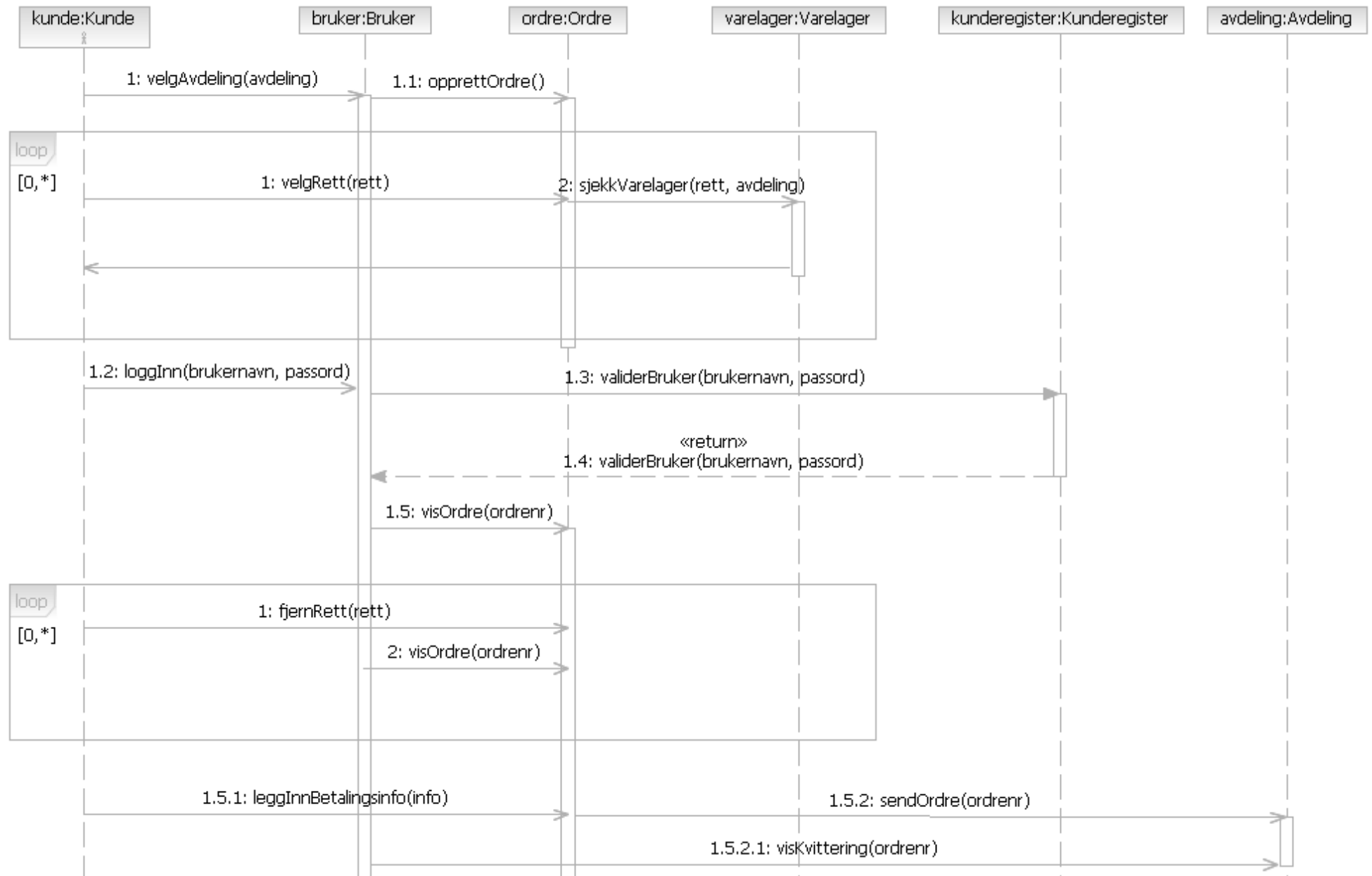


3.2.2. Detaljdesign på Use Caset Bestille

Vi har her valgt å lage et detaljert objektorientert design på UseCaset Bestille. Dette er det mest kritiske UseCaset ettersom det danner grunnlaget for hele systemet, og alle andre deler er avhengige av det. Kontakten med kundene vil også foregå her, og det er derfor viktig at bestillingsdelen av systemet fungerer optimalt for å skaffe seg en så stor kundegruppe som mulig. Vi vurderer i tillegg Bestille til å være tidsmessig krevende. Alt dette gjør at vi anser Bestille til å være det viktigste UseCaset, og har derfor gått dypere inn på dette.



3.2.3. Detaljert Sekvensdiagram for caset bestille



Inkludert i designklassediagrammet er klassen Varelager, som fungerer som et grensesnitt mot avdelingens varelagersystem. Dette systemet er imidlertid ikke en del av vårt system. Klassen Varelager brukes for å sjekke om en gitt avdeling har nok råvarer til å lage rettene kunden har bestilt.

4. Test Plan

Det vil bli avholdt møter etter fullføringen av hvert inkrement, hvor inspeksjon av koden foretas på de viktigste/mest kritiske delene av inkrementet. Dette gjøres for å sikre kvaliteten på koden, og for å spare ressurser senere i prosjektet, siden det vil være enklere å rette opp feilene som avdekkes nå enn senere. Vi vil også trenge mindre tid på testing av systemet etter at utviklingsfasen er fullført.

Inspeksjon av koden vil i særlig grad utføres på sikkerhetskritiske deler, som f. eks kundenes innlogging og systemets administrasjonsdel, for å minske risikoen for at noen kan bryte seg inn i systemet. Inspeksjonen skal ikke utføres av den samme som har skrevet koden. Dette fordi det vil være lettere for en som ikke har sett koden før å komme med nye innspill og oppdage feil, siden denne ikke har noen form for følelsesmessig tilknytning til koden.

Vi vil også foreta enhetstester av inkrementene etter hvert som de blir ferdige. Når et inkrement skal legges til systemet, skal det foretas en test av hele systemet for å sikre at det virker som det skal etter utvidelsen. Ved både enhetstesting og integrasjonstesting vil vår kontaktperson hos Aladdin AS bistå, i tillegg til en uavhengig testgruppe. I den gruppen vil ansatte ved Aladdin og personer fra Aladdins kundegruppe inngå.

De som har vært med på å lage de forskjellige delene i systemet skal også være med på testingen, men hovedtyngden av testingen vil

foregå av uavhengige når det gjelder deler av systemet som skal brukes av ansatte ved Aladdin og kunder. Når det gjelder programdeler som er automatiserte, vil det bli intern testing, hvor det vil bli rulling på testingen. Dette for at andre enn selve ansvarlige skal se og prøve koden.

Det som skal testes i de forskjellige inkrementene er:

Ordresystem:

- At ordre overføres på korrekt måte fra hovedkontor til avdeling
- At oppdateringen og sjekk av varelager fungerer som den skal
- At ordresystemet varsler ansatte ved feil
- At ferdigstilte ordrer overføres til sjåførsystemet
- At ordrer fra kassesystemet overføres til ordresystemet
- At kun registrerte/godkjente kunder og bedrifter kan bestille

Kunderegister:

- At data om kunder lagres og oppdateres uten feil i databasen

Fakturasystem:

- At systemet kan vise, lagre og skrive ut faktura
- At systemet bruker riktig bedriftsavtale
- Sjekke kommunikasjon mellom fakturasystem og regnskapssystem

Hjemmeside:

- At kunder kan navigere på nettstedet uten problemer

Hver test skal dokumenteres med hva som ble testet, resultatet av testen, dato og aktører.

Det er særdeles viktig at ordresystemet er grundig testet, da dette må fungere optimalt for at hele Aladdin systemet skal fungere som forventet. Ordresystemet har mange interaksjoner med eksterne systemer og dette bør kontrolleres og sjekkes nøye med tanke på at det kan komme inn uventede data fra disse systemene

Under testing av websystemet er det viktig at brukere med forskjellige typer bakgrunn og nivå bør teste dette. Dette er en del av systemet som må være meget brukervennlig og fungere uten feil. Dette for at kundene skal føle at de mestrer siden og at de sitter igjen med en positiv opplevelse og ønsker å komme tilbake til Aladdin ved en senere anledning.

Feilhåndtering er viktig å få testet ut, da det er en ting å luke ut feil i programvare, mens andre situasjoner som feil på linjer å sånn er det vanskelig å gjøre noe med. Når det oppstår brudd mellom hovedkontor og avdeling, bør ansatte få beskjed om dette. Det samme gjelder når ordresystemet ikke kan finne rett avdeling eller ordren ble sendt til avdeling og men en kvittering ble ikke returnert.

Dette for å sikre at ansatte/ansvarlige blir varslet ved slike feil og for å kunne instruere ansatte slik at de selv kan håndtere noen av slike feilsituasjoner.

Den totale testperioden vil ta 20% av prosjektiden. Fordelt tid på de forskjellige inkrementene er opplyst i Gantt-skjemaet under punkt 1.6. Innad i testperioden vil enhetstesting være den delen som tar mest tid ettersom vi pga den valgte utviklingsmodellen (inkrementell) vil bruke tid på å teste hvert inkrement, i tillegg til hele systemet. Av inkrementene er ordresystemet det tyngste, og vil også kreve mest testing.