



BACHELOROPPGAVE:

*BOLIGHANDEL 1*

FORFATTERE:

THOMAS A. ALMENNINGEN

LARS ERIK STRAND

AMUND SØRUMSHAGEN

Dato:

23.05.2012

## SAMMENDRAG

|  |  |                                 |
|--|--|---------------------------------|
| Tittel:  | BoligHandel1   | Dato : 23.05.12                 |
| Deltaker(e)/   | Thomas A. Almenningen<br><hr/> Lars Erik Strand<br><hr/> Amund Sørumshagen |                                 |
| Veileder(e):   | Øivind Kolloen   |                                 |
| Evt. oppdragsgiver:  | Bolighandel1   |                                 |
| Stikkord/nøkkel ord  | Auksjon bolighandel  |                                 |
| (3-5 stk)  |  |                                 |
| Antall sider/ord: 128 sider, 21 849 ord  | Antall vedlegg: 4  | Publiseringsavtale inngått: nei |
| <p>Kort beskrivelse av bacheloroppgaven:</p> <p>BoligHandel1 sin virksomhet handler om å selge boliger uten bruk av eiendomsmegler.</p> <p>De forskjellige produktene strekker seg fra enkel hjelp med kontrakt og oppgjør, til full pakke som inkluderer annonsering, visning, salg og alt kjøp/salg av bolig vil inneholde.</p> <p>BoligHandel1 bruker i dag en blanding av sin egen hjemmeside, finn.no og QXL.no i sin virksomhet. Hvor QXL.no brukes til auksjonene.</p> <p>Oppgaven går ut på å utvikle en solig og teknisk nyvinnende løsning for budgivning på eiendom.</p> <p>Løsningen vil innebære en webside som skal kunne fungere tilsvarende som den de benytter seg av i dag, men i tillegg skal det være mulighet for kjøp og salg av eiendom. Kjernen i oppgaven blir å implementere selve presentasjonen og auksjonen rundt eiendommen i denne løsningen, slik at bruk av QXL.no er unødvendig.</p> |  |                                 |

## ABSTRACT

|   |  |                    |
|---|--|--------------------|
| Title:  | BoligHandel1   | Date : 23.05.12    |
| Participants/   | Thomas A. Almenningen<br><hr/> Lars Erik Strand<br><hr/> Amund Sørumshagen |                    |
| Supervisor(s)   | Øivind Kolloen   |                    |
| Employer:   | BoligHandel1   |                    |
| Keywords  | Auction realestate   |                    |
| (3-5)   |  |                    |
| Number of pages/words: 128 pages, 21 849 words  | Number of appendix: 4  | Availability: open |
| <p>Short description of the bachelor thesis:</p> <p>BoligHandel1's goal is to sell real estate without the use of a real estate broker. Their products covers aspects ranging from help with contracts and final payment to full professional help with everything from showing the object to the final dealings.</p> <p>Today BoligHandel1 uses a mix of their own website, finn.no and qxl.no for their business.</p> <p>Where qxl.no is the site where auctions are held.</p> <p>Our assignment is to develop a solid and innovative solution for buying and selling real estates.</p> <p>The solution will involve a website which will work in the same fashion as the one they are using today, but with the possibility for dealing in real estates. The main part of our assignment will be to implement the presentation and the auction part for real estate in this solution, so that BoligHandel1 no longer will have to use QXL.no for the auctions.</p> |  |                    |

## FORORD

Oppgaven vi har valgt er laget av Asgeir og Gaute Strandlie som driver firmaet BoligHandel1 og nettsiden [www.bolighandel1.no](http://www.bolighandel1.no). BoligHandel1 sitt konsept går ut på at du som privatperson skal kunne selge bolig uten å gå veien gjennom et meglerfirma.

Pr. i dag er de den eneste aktøren som tilbyr elektronisk budgivning med verifisering gjennom BankID gjennom den kjente nettauksjonssiden QXL.no.

Oppgaven vår blir å designe en løsning på web som skal kunne stå på egne ben uten bruk av ekstern aktør som QXL.no. Finn.no skal inntil videre benyttes grunnet deres reklameverdi i markedet. Dermed blir kjernen i oppgaven å designe en nettside hvor man kan gjennomføre kjøp og salg av eiendom, som skal realiseres gjennom en auksjon/budrunde på samme måte som de benytter QXL.no til i dag. For en ferdig løsning til det norske markedet vil det være nødvendig med en godkjent autentiseringsløsning som BankID eller Buypass. Dette vil ikke implementeres på grunn av at dette er et null-budsjetts-prosjekt, men vi skal likevel redegjøre for hvilke muligheter som finnes i Norge, samt et anslag på hvor mye dette vil koste.

Vi vil få rette en takk til både Asgeir og Gaute for svært lærende og produktivt samarbeid med to meget hyggelige karer.

Kan også nevne veileder, Øivind Kolloen, som alltid har døra på gløtt og som sjekker mail hvert 5.min slik at hjelpen aldri er langt unna.

Gjøvik, 23.mai 2012

Thomas A. Almenningen

Lars Erik Strand

---

---

Amund Sørumshagen

---

## INNHALDSFORTEGNELSE

|  |    |
|--|----|
| 1. Innledning.....   | 9  |
| 1.1 Organisering av Rapporten – Sammenheng mellom kapitler ..... | 9  |
| 1.2 Definerings av oppgaven .....                                | 9  |
| 1.2.1 Bakgrunn og oppdragsgivers visjon.....                     | 9  |
| 1.2.2 Målsetning og hvorfor dette emnet?.....                    | 10 |
| 1.2.3 Definerings av oppgaven .....                              | 11 |
| 1.2.4 Avgrensninger .....  | 12 |
| 1.3 Målgruppe .....  | 12 |
| 1.4 Vår faglige bakgrunn .....                                   | 12 |
| 1.5 Valgte arbeidsformer .....                                   | 13 |
| 1.6 Øvrige roller.....   | 14 |
| 2. Kravspesifikasjon .....                                       | 15 |
| 2.1 Systemets omgivelser.....                                    | 15 |
| 2.2 Systemets aktører .....                                      | 15 |
| 2.3 Krav til løsningen .....                                     | 15 |
| 2.4 Overordnede krav til systemet.....                           | 17 |
| 2.5 Kvalitetsmessige krav .....                                  | 23 |

|   |    |
|---|----|
| 2.6 Detaljerte use case beskrivelser .....                                  | 24 |
| 2.7 Supplementære krav .....  | 28 |
| 3. Design .....   | 30 |
| 3.1 Komponent design .....  | 30 |
| 3.2 Data view .....   | 31 |
| 4. Realisering .....  | 33 |
| 4.1 CMS.....  | 33 |
| 4.1.1 Valg av CMS .....   | 34 |
| 4.2 Hvordan fungerer Joomla.....  | 35 |
| 4.2.1 Introduksjon – hva er Joomla! ? .....                                 | 35 |
| 4.2.2 Historie .....  | 36 |
| 4.2.3 Kort forklaring av de tekniske løsningene og begrepene i Joomla ..... | 37 |
| 4.2.4 Joomla i praksis .....  | 43 |
| 4.3 Komponentutvikling .....  | 51 |
| 4.3.1 Komponentens oppbygning .....   | 51 |
| 4.3.2 Brukerinteraksjon i komponenter .....                                 | 57 |
| 4.3.3 JDatabase .....   | 59 |
| 4.4 Komponentutviklingsprosessen .....                                      | 60 |
| 4.4.1 Auksjonsliste-visning .....   | 61 |
| 4.4.2 Kartvisning av auksjoner .....  | 62 |
| 4.4.3 Utvikling av auksjonssiden .....                                      | 64 |
| 4.4.4 Nytt bud funksjonen.....  | 66 |

|   |    |
|---|----|
| 4.4.5 Utvikling av Min Side .....                               | 69 |
| 4.4.6 Utvikling av auksjonsinformasjonside .....                | 70 |
| 4.4.7 Administrator side .....                                  | 71 |
| 4.5 Plugins/APIer benyttet til utvikling av komponenten: .....  | 72 |
| 4.6 Joomla Extensions brukt i utvikling av siden .....          | 74 |
| 4.6.1 Chronoforms [25] .....                                    | 74 |
| 4.6.2 Oversikt over former vi har opprettet i Chronoforms ..... | 78 |
| 4.6.3 Andre utvidelser .....                                    | 83 |
| 4.7 Andre Beslutninger .....                                    | 83 |
| 4.7.1 Valg av karttjeneste: .....                               | 83 |
| 4.7.2 Valg av SMS-Tjeneste .....                                | 84 |
| 4.7.3 Hosting .....   | 85 |
| 4.7.4 Login .....   | 86 |
| 4.7.5 Cron-jobs .....   | 86 |
| 4.8 Autentiseringsteknologier .....                             | 87 |
| 4.8.1 Hvilke teknologier for autentisering finnes? .....        | 88 |
| 4.8.2 Fordeler og ulemper ved de aktuelle teknologiene .....    | 88 |
| 4.8.3 Valg av autentiseringsløsning .....                       | 90 |
| 4.8.4 Veien videre .....  | 90 |
| 5. Testing .....  | 93 |
| <b>5.1 Utvikling</b> .....                                      | 93 |
| <b>5.2 Eksternt</b> .....                                       | 93 |

|  |     |
|--|-----|
| 5.3 Auksjoner.....                           | 93  |
| 5.5 Fokus.....                               | 93  |
| 6. Avslutning.....                           | 94  |
| 6.1 Diskusjon av resultater .....            | 94  |
| 6.2 Forbedringspotensial.....                | 96  |
| 6.3 Videre arbeid .....                      | 97  |
| 6.4 Evaluering av arbeid .....               | 98  |
| 6.5 Erfaringer .....                         | 99  |
| 6.6 Konklusjon .....                         | 100 |
| 7. Referanser .....                          | 102 |
| Vedlegg A – Forprosjekt .....                | 104 |
| Vedlegg B – Timelogger og møtereferater..... | 116 |
| Vedlegg C – Statusrapporter .....            | 124 |
| vedlegg D – Prosjektavtale .....             | 127 |



## 1. INNLEDNING

### 1.1 ORGANISERING AV RAPPORTEN – SAMMENHENG MELLOM KAPITLER

Rapporten er delt inn i åtte hovedkapitler:

- Innledning: definerer oppgaven, og forklarer den faglige bakgrunnen i prosjektgruppa.
- Kravspesifikasjon: viser krav og retningslinjer til systemet.
- Design: viser hovedtrekkene av hvordan oppgaven skal løses.
- Realisering: beskriver utarbeidelsen av hele prosjektet samt hvilke valg som er gjort.
- Testing
- Avslutning
- Vedlegg

### 1.2 DEFINERING AV OPPGAVEN

---

#### 1.2.1 BAKGRUNN OG OPPDRAGSGIVERS VISJON

##### BAKGRUNN FOR OPPGAVEN

Utviklingen i markedet går mot at flere og flere velger å selge sin eiendom uten å benytte eiendomsmegler. Derfor vil det være et enda større behov for tilpassede og effektive verktøy for nettopp dette i fremtiden. Den store utfordringen er å designe dette på en måte som gjør det enkelt å forstå, men også gjør mulighetene for menneskelige feil minimale.

BoligHandel1 har erfart at boligeiere ofte kan gjøre den samme jobben (og kanskje bedre?) som en megler kan gjøre. Derfor tilbyr BoligHandel1 deg de verktøy du trenger for å gjennomføre den viktigste salgsjobben selv på en profesjonell og trygg måte.

## VISJON

BoligHandel1 sin visjon er at de i størst mulig grad skal erstatte en eiendomsmeglers rolle ovenfor boligeier/selger. Dette kan BoligHandel1 gjøre ved å tilby en fullstendig nettbasert løsning ved eiendomshandel, med elektronisk budgivning som garanterer en trygg og sikker budrunde.

---

### 1.2.2 MÅLSETNING OG HVORFOR DETTE EMNET?

Vår målsetning i dette prosjektet vil være todelt.

#### TEKNISK

Vi skal høste erfaringer rundt det å jobbe med et "Content Management System" (Utviklingsverktøy). Ingen av oss har gjort dette før, og derfor vil det i seg selv være en oppgave å sette seg grundig inn i systemet, samt hvilke muligheter og begrensninger det har.

Vi har også mål om å utvikle en webside som inneholder auksjonsfunksjonaliteten så man kan gjennomføre auksjoner slik oppdragsgiver ønsker. Det følger også mulighet for å registrere seg som bruker, og legge ut en salgspresentasjon av sin eiendom. Vår ferdige løsning vil ikke kunne brukes i markedet fordi den ikke vil inneholde en autentiseringsløsning. Vi skal utrede litt om de forskjellige mulighetene for autentisering og et røft anslag om hvor mye dette vil koste.

#### KOMMUNIKASJON MED KUNDE

Vår oppdragsgiver, eller kunde, har brukt et CMS til å sette opp sin egen nettside. Allikevel har de ingen erfaring med de tekniske løsningene som ligger under "panseret", det vil si programmering, databaser eller selvdefinerte muligheter ved det CMS de har brukt.

Målsetningen vår med tanke på dette punktet vil være å tilfredsstillende kundens krav og ønsker og oversette disse fra munn og skrift til design og funksjonalitet på nettsiden. Det å diskutere ønsker og krav opp imot mulighetene og begrensningene som ligger både i vår kompetanse og generelle tekniske muligheter vil være en meget viktig erfaring å ta med seg videre.

## HVORFOR DETTE EMNET?

Når vi så over og leste gjennom oppgavene, hadde vi ingen preferanser på type oppgave vi ønsket. Etter å ha sett gjennom stakk denne oppgaven seg ut.

Denne oppgaven innebærer å lage en funksjonalitet på web. Det å utvikle applikasjoner for web og hele nettløsninger vil det bli mer og mer av i fremtiden. Vi i gruppa er alle tre meget interesserte i å skaffe oss erfaringer rundt nettopp det å utvikle for web, og denne oppgaven var da svært interessant for oss.

---

### 1.2.3 DEFINERING AV OPPGAVEN

BoligHandel1 benytter seg i dag av nettauksjonssiden QXL.no. Vår oppgave går ut på å utvikle et utkast til en nettside for BoligHandel1 hvor denne auksjonsfunksjonaliteten er innebygd. For at en slik auksjon skal kunne gjennomføres, vil det også måtte legges til mer funksjonalitet.

Det må være en mulighet for en person til å opprette en brukerkonto. En brukerkonto må man ha hvis man skal kunne selge en eiendom, eller by på en eiendom.

For en bruker må det være mulig å opprette en auksjon med tilhørende tekst, som igjen trenger funksjonalitet for å laste opp filer i form av bilder og dokumenter som har med denne eiendommen å gjøre.

Det skal også utvikles et grensesnitt for administratorer, i dette tilfellet ansatte i BoligHandel1, hvor de enkelt kan legge inn nye og administrere produkter, godkjenne auksjoner opprettet av bruker eller hvor de selv kan legge til auksjoner.

Forsiden på web skal også fungere som en reklame-side for hva BoligHandel1 tilbyr. I vår løsning begrenser dette seg til å vise frem de ulike produktene de tilbyr, samt at forsiden skal vise frem et utvalg av eiendommer.

Gjennom grensesnittet for administratorer vil man kunne velge de produktene og de eiendommene som skal vises på forsiden.

Vi skal også redegjøre for hvilke autentiseringsløsninger som kan være aktuelle for bruk av denne nettauksjonsløsningen i Norge.

#### 1.2.4 AVGRENSNINGER

Vår ferdige løsning, vil kunne gjennomføre en auksjon. Men uten noen form for autentisering, vil det ikke kunne gjøres bindende i henhold til Norsk lov. Derfor vil man i en ferdig løsning for bruk i markedet måtte bruke en eller annen form for slik autentisering. Da eksempelvis BankID, som QXL.no benytter, eller BuyPass som begge vil kunne brukes til kjøp og salg av eiendom. Implementering av dette i vår løsning vil påføre prosjektet en kostnad, og siden verken vi som studenter eller oppdragsgiver i denne omgang er villige til et slikt utlegg, vil vi bare redegjøre for hvilke muligheter man har og hvilke kostnader det vil påføre BoligHandel1 skulle de velge å ta dette videre.

I selve oppgaveskrivet i fra BoligHandel1 var det også et ønske om internasjonalisering av løsningen. Dette punktet ville ført med seg utrolig mye jobb spesielt med å sette seg inn i lover og regler rundt kjøp og salg av eiendom og autentiseringsløsninger i de respektive landene. Det vil med andre ord strekke seg ut over både tidsbruk og teknisk overgå vår kompetanse pr. dags dato.

#### 1.3 MÅLGRUPPE

For å forstå det tekniske innholdet av denne rapporten er det viktigste at man forstår programmering, og hvordan samspillet mellom kode og en relasjonsdatabase fungerer. All kommunikasjon med relasjonsdatabasen i vårt prosjekt går gjennom skriptspråket php, og derfor vil forståelse og litt erfaring med php være vesentlig.

Målgruppen vår blir da studenter og sensor med samme eller liknende bakgrunn som listet over. Hva som ligger i "samme bakgrunn" vil komme klarere frem i punktet nedenfor.

#### 1.4 VÅR FAGLIGE BAKGRUNN

##### KOMPETANSE

Vår faglige bakgrunn som gruppe kan oppsummeres som følger:

- Grunnleggende programmeringsfag: Assembly, C/C++, Java
- Grunnleggende databasekompetanse i fra MySQL (Generelt relasjonsdatabaser)
- Grunnleggende skripting som PHP, JavaScript (med bruk av JQuery og AJAX), PowerShell og Bash.
- Innføring i algoritmiske metoder (C++).
- Systemutvikling med innføring i systemutviklingsmodeller, planlegging, gjennomføring og rapportskrivning.

### HVA MÅ LÆRES?

Ingen av oss har tidligere erfaringer med CMS. Og på det nivået denne oppgaven må gjøres, vil det å sette seg relativt grundig inn i hvordan et CMS fungerer være et meget viktig punkt. I tillegg vil vi måtte bygge inn funksjonalitet i dette utviklingsverktøyet da det helt sikkert ikke finnes ferdiglagede komponenter som gjør akkurat det vi skal gjøre.

Vil også måtte lære litt mer avansert bruk av JQuery-biblioteket, AJAX og JavaScript generelt.

## 1.5 VALGTE ARBEIDSFORMER

### KOMMUNIKASJON MED OPPDRAGSGIVER

Vi valgte å utvikle web-siden bit for bit (inkrementell modell). Grunnet muligheten for at oppdragsgiver hele tiden kunne komme med nye ønsker, innspill og forslag til endringer.

Vi har hele veien hatt jevnlig kommunikasjon pr. mail, pluss en del innlagte møter. Helt ifra første møtet hvor den grove kravspesifikasjonen ble utarbeidet, og videre statusmøter underveis. Oppdragsgiver har hele tiden hatt tilgang til siden, og det har åpnet for muligheten for å sende en mail med korte ønsker og forslag til endringer hele veien.

Denne måten å jobbe på, har også gjort det enklere å utarbeide alternative løsninger hvis ønsket/kravet i utgangspunktet vanskelig lot seg gjøre.

## LITTERATUR

Vi har brukt svært lite litteratur i bokform gjennom dette prosjektet.

Det meste av dokumentasjon som vi var avhengig av for gjennomføring, ligger på nett slik at måten å innhente dette på stort sett ble gjort gjennom nettsøk.

### 1.6 ØVRIGE ROLLER

BoligHandel1 skal hele tiden være med i prosessen gjennom fysiske møter og elektronisk kommunikasjon. Vi vil hele veien komme over problemer eller alternativer til løsninger der det er BoligHandel1 sitt valg om hvordan ting skal se ut eller fungere. De har hele tiden tilgang til løsningen (web-siden) slik at de løpende kan komme med innspill hva gjelder ny funksjonalitet eller endringer av design eller eksisterende funksjonalitet.

Asgeir Strandlie har eiendomsmeglerutdanning med tittelen Eiendomsmegler (som krever minst to års godkjent praksis og gjennomført utdanning.) Har jobbet 15år i eiendomsmeglerfirmaene Terra og Garanti. Har vært innom alle fagområder av eiendomsmegling, og benytter sin ekspertise og kompetanse til å utvikle og kvalitetssikre de produktene og tjenestene BoligHandel1 tilbyr. Asgeir har ansvar for at alt innhold, faglig/juridisk eller de faktiske markedsproduktene skal holde tilsvarende kvalitet som hos et meglerfirma. Kan påpeke at Asgeir ikke jobber som megler i BoligHandel1, men som konsulent for deg som vil selge privat.

Gaute Strandlie er ansvarlig for design og utvikling av alt markedsmateriell som tilbys boligselgerne. Har også ansvaret for webutviklingen, samt design, videreutvikling, vedlikehold og oppdatering av BoligHandel1 sine nettsider.

Vår veileder, Øivind Kolloen, underviser daglig i faget WWW Teknologi. Det gjør at han er en naturlig støttespiller rundt tekniske utfordringer og problemer i selve løsningen.

## 2. KRAVSPESIFIKASJON

Dette kapitlet omhandler krav til hva den endelige løsningen skal inneholde. Kravene i er utledet fra den originale oppgavebeskrivelsen, samt møter med oppdragsgiver.

### 2.1 SYSTEMETS OMGIVELSER

Systemet vil basere seg på Joomla! CMS. Nødvendig kunnskap om bruk av Joomla! må tilegnes for å kunne benytte seg av de mange klassene og funksjonene som Joomla! har. Joomla! genererer også en egen database for å ta vare på nødvendig informasjon. Vi vil benytte den samme databasen for å ta vare på relevant informasjon om eiendommer og auksjoner.

### 2.2 SYSTEMETS AKTØRER

Vi har primært to typer brukere i vår løsning.

**Bruker** – Brukeren skal kun ha tilgang til Joomla! front-end, dvs selve nettsiden.

**Administrator** – Skal i tillegg ha adgang til Joomla! back-end hvor man kan administrere innholdet på nettsiden samt en administrator del i front-end.

Brukeren kan eventuelt deles i to undergrupper: selger og budgiver. I selve løsningen er det i realiteten ingen forskjell mellom disse undergruppene. Vi har derfor valgt å omtale både selgere og budgivere som brukere.

### 2.3 KRAV TIL LØSNINGEN

Løsningen skal være automatisert i den grad at hverken selger eller budgiver trenger å være tilgjengelig i det budrunden går mot avslutning. Det skal også være mulig å følge auksjonene

i sanntid, både på internett og via håndholdte enheter. I tillegg skal det være like enkelt og oversiktlig å legge inn bud via håndholdte enheter som via pc.

For hvert bud som legges inn skal budgiver identifiseres elektronisk ved hjelp av en autentiseringsløsning. Budgivers identitet skal heller ikke være synlig for andre.

Alle brukere skal kunne velge å følge budgivningen via sms. De skal da motta en sms for hvert nye bud som registreres. For budgivere og selgere skal sms mottas automatisk med beskjed om innhold i høyere bud.

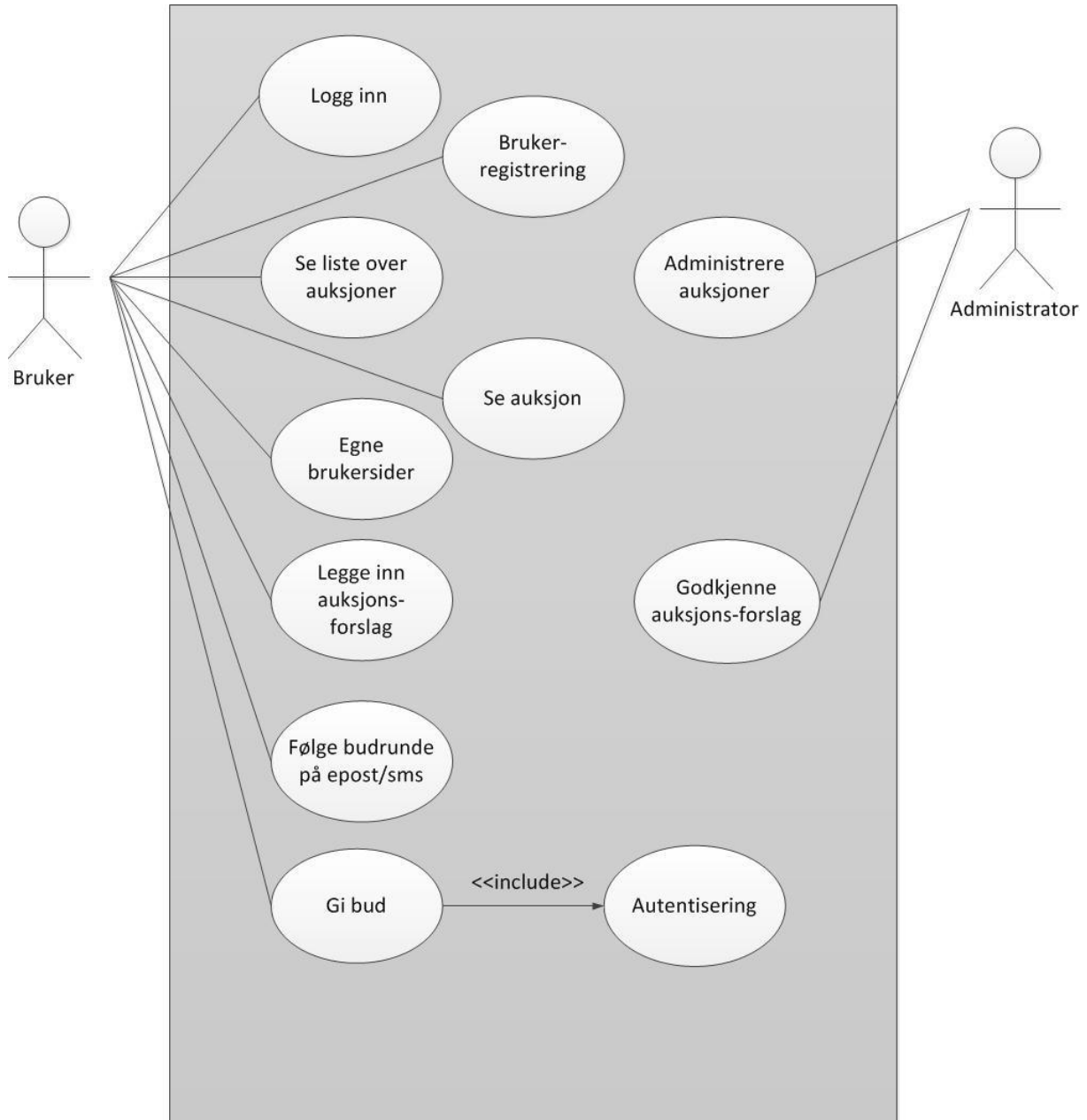
Selger skal selv kunne bestemme en auksjons varighet (startdato og sluttdato). I tillegg skal selgere kunne legge inn startnivå på budgivningen, samt angi en minstepris som vil bli akseptert. Denne minsteprisen kan ligge skjult eller åpent. På samme måte skal en budgiver kunne definere et nivå for maksbud som vedkommende er villig til å by dersom budgivningen skulle nå dette nivået. En budgivers maksbud skal ikke være synlig for andre.

Som en ekstra sikkerhet skal selger ha mulighet til å godkjenne eventuelle budgivere før disse budgiverne kan legge inn bud.



## 2.4 OVERORDNEDE KRAV TIL SYSTEMET

## USE CASE DIAGRAM



Figur 2.4.1 Use Case Diagram

**Kommentar:** Siden løsningen ikke inneholder noen restriksjoner mot at administrator kan gi bud osv, vil administrator i tillegg ha tilgang til nøyaktig samme funksjonalitet som den vanlige brukeren. Vi har derimot valgt å utelate disse relasjonene fra diagrammet.

## OVERORDNEDE USE CASE BESKRIVELSER

| Use Case           | Logg inn   |
|--------------------|--|
| <b>Aktør</b>       | Bruker   |
| <b>Hensikt</b>     | Bruker skal kunne logge inn på nettsiden for å få tilgang til begrenset funksjonalitet.  |
| <b>Beskrivelse</b> | Brukeren angir et selvvalgt brukernavn og passord for å logge på nettsiden. Etter innlogging får brukeren tilgang til og gi bud, opprette et annonse/auksjonsforslag, samt egne brukersider. |

| Use Case           | Brukerregistrering  |
|--------------------|---|
| <b>Aktør</b>       | Bruker  |
| <b>Hensikt</b>     | Brukeren skal kunne registrere seg på siden.  |
| <b>Beskrivelse</b> | Dersom brukeren ikke har konto, skal han/hun kunne registrere seg for å kunne gi bud eller opprette et annonse-/auksjonsforslag. Denne registreringsmuligheten skal være knyttet til det å faktisk gi bud eller opprette annonse. |

| Use Case           | Se liste over auksjoner   |
|--------------------|---|
| <b>Aktør</b>       | Bruker  |
| <b>Hensikt</b>     | Bruker skal kunne se en liste over alle tilgjengelige auksjoner.  |
| <b>Beskrivelse</b> | Brukeren følger menylenken «Eiendommer» og får en visning over alle auksjoner som ikke er avsluttet. Brukeren kan i tillegg søke blant disse auksjonene for å finne ønskede auksjoner. En enkel auksjon kan velges blant disse for å få mer detaljert informasjon om auksjonen. |

| Use Case           | Se auksjon  |
|--------------------|---|
| <b>Aktør</b>       | Bruker  |
| <b>Hensikt</b>     | Bruker skal kunne se detaljert informasjon om en gitt auksjon.  |
| <b>Beskrivelse</b> | Brukeren velger en enkelt auksjon og får vist detaljert informasjon angående valgt auksjon. Her vil all relevant informasjon om auksjonen ligge samt funksjonaliteten for å søke om godkjenning som budgiver og eventuelt gi bud. |

| Use Case           | Se/bestille produkter   |
|--------------------|---|
| <b>Aktør</b>       | Bruker  |
| <b>Hensikt</b>     | Brukeren skal kunne se en oversikt over tilgjengelige produkter og bestille produkt.  |
| <b>Beskrivelse</b> | Brukeren følger menylenken «Produkter» og får så en liste over alle tilgjengelige produkter med overordnet informasjon om disse. Brukeren kan velge et produkt blant disse for å få mer detaljert informasjon om valgte produkt. En bestillingslenke kan velges for å finne bestillingsskjema hvor brukeren kan bestille et ønsket produkt. |

| Use Case           | Egne brukersider  |
|--------------------|---|
| <b>Aktør</b>       | Bruker  |
| <b>Hensikt</b>     | Brukeren skal kunne holde oversikt over sin profil.   |
| <b>Beskrivelse</b> | Etter innlogging får brukeren tilgang til egne brukersider. Her skal brukeren kunne holde oversikt over sine bud på auksjoner, salgsannonser, |

abonnementer, samt brukerprofil. Brukeren skal også kunne redigere sin brukerprofil, sine abonnementer, og annonser/auksjoner dersom disse ikke er startet.

| Use Case           | Følge budrunde på epost/sms   |
|--------------------|---|
| <b>Aktør</b>       | Bruker  |
| <b>Hensikt</b>     | Bruker skal kunne følge en ønsket budrunde på epost eller sms.  |
| <b>Beskrivelse</b> | Brukeren skal ha mulighet til å abonnere (følge med) på ønskede auksjoner ved epost eller sms. Brukeren vil da få epost eller sms når bud legges inn på auksjonen, samt når auksjonen starter og slutter. |

**Kommentar til beskrivelsene:** Som aktører i de ovennevnte beskrivelsene har vi kun nevnt brukeren. De samme beskrivelsene vil også gjelde for administrator da administrator har tilgang til den samme funksjonaliteten mm.

| Use Case           | Administrere auksjoner  |
|--------------------|---|
| <b>Aktør</b>       | Administrator   |
| <b>Hensikt</b>     | Administrator skal kunne administrere informasjon på nettsiden.   |
| <b>Beskrivelse</b> | Administrator skal ha mulighet til å enkelt administrere innhold på nettsiden. Dette gjelder visning og redigering av auksjoner, oversikt over abonnementer, samt visning og redigering av produkter. |

| Use Case           | Godkjenne annonseforslag  |
|--------------------|---|
| <b>Aktør</b>       | Administrator   |
| <b>Hensikt</b>     | Godkjenne et annonseforslag fra bruker og publisere annonsen på siden   |
| <b>Beskrivelse</b> | Når du er pålogget som administrator vil du få opp en oversikt over annonseforslag som avventer registrering. Når du åpner et annonseforslag vil all dataen som brukeren tastet inn lastes inn i feltene. Du kan så gjøre endringer på de nødvendige feltene før du velger om du ønsker å publisere eller forkaste forslaget. Brukeren som la inn annonseforslaget vil bli informert om annonseforslaget ble godkjent eller ikke på mail. |

| Use Case           | Redigere auksjon, begrenset  |
|--------------------|--|
| <b>Aktør</b>       | Administrator, Bruker (Selger)   |
| <b>Hensikt</b>     | Redigere informasjon om salgsobjektet, Ikke informasjon relatert til selve auksjonen (Startnivå, minstepris, start og sluttdato etc.)                    |
| <b>Beskrivelse</b> | Brukeren skal til en hver tid ha mulighet til å redigere feltene i sin annonse. Brukeren vil bli videresendt til denne formen hvis auksjonen er påbegyt. |

| Use Case           | Redigere auksjon, Alle felter   |
|--------------------|---|
| <b>Aktør</b>       | Administrator, Bruker (Selger)  |
| <b>Hensikt</b>     | Redigere all informasjon om et salgsobjekt  |
| <b>Beskrivelse</b> | Brukeren skal til en hver tid ha mulighet til å redigere feltene i sin annonse. Brukeren vil bli videresendt til denne formen hvis auksjonen ennå ikke er påbegynt slik at han/hun kan redigere samtlige felter i annonsen. |

| Use Case           | Budgiver Søknad  |
|--------------------|--|
| <b>Aktør</b>       | Bruker (Budgiver)  |
| <b>Hensikt</b>     | Søke om status som budgiver, Sende all nødvendig informasjon slik at en bruker kan avgjøre om du har finansieringen i orden.   |
| <b>Beskrivelse</b> | Dersom du er interessert i å by på en bolig er du nødt til å sende inn en budgiver søknad med personlig informasjon samt en bekrefelse på at du har finansieringen i orden. Denne søknaden blir videresendt til selger slik at selgeren kan avgjøre om han ønsker å gi interessenten lov til å by på boligen eller ikke. |

| Use Case           | Godkjenne budgiver søknad   |
|--------------------|---|
| <b>Aktør</b>       | Bruker (Selger)   |
| <b>Hensikt</b>     | Godkjenne eller avslå en budgiver søknad  |
| <b>Beskrivelse</b> | Selgeren vil få opp en notifikasjon på sine sider dersom det er registrert en ny budgiver søknad. Han vil også motta en epost hver gang det registreres en ny budgiver søknad. På godkjenningssiden vil selgeren få opp søknadsinformasjonen og valget om han ønsker å godkjenne søknaden eller ikke. Dersom han velger nei vil han også ha muligheten til å skrive en kort begrunnelse på hvorfor søknaden ikke ble godkjent. Dersom budgiveren blir godkjent vil han legges til listen med godkjente budgivere. Han vil også bli notifisert om utfallet på epost. |

| Use Case       | Stille spørsmål til selger         |
|----------------|------------------------------------|
| <b>Aktør</b>   | Bruker (Budgiver)                  |
| <b>Hensikt</b> | Videresende et spørsmål til selger |

|                    |   |
|--------------------|---|
| <b>Beskrivelse</b> | Eventuelle kjøpere skal ha mulighet til å stille selgeren spørsmål på epost, Dette gjøres via denne formen. |
|--------------------|---|

| Use Case           | Laste opp/Endre salgsoppgave, takst, budskjema                                     |
|--------------------|--|
| <b>Aktør</b>       | Bruker (Selger)  |
| <b>Hensikt</b>     | Laste opp salgsoppgave, takst og budskjema. Samt å ha muligheten til å endre disse |
| <b>Beskrivelse</b> | Brukeren skal ha muligheten til å laste opp salgsoppgave, takst og budskjema.      |

| Use Case           | Registrer auksjonsforslag   |
|--------------------|---|
| <b>Aktør</b>       | Bruker (Selger)   |
| <b>Hensikt</b>     | Legge inn ett annonseforslag som må godkjennes av administrator   |
| <b>Beskrivelse</b> | For at en auksjon skal bli publisert på siden må brukeren først fylle ut et standard skjema med informasjon om salgsobjektet. Etter at brukeren har fylt ut de påkrevde feltene og lastet opp auksjonen vil denne legges i databasen. feltet 'published' blir satt til 0 som default, auksjonen vil derfor ikke publiseres på siden før en administrator har godkjent auksjonene og eventuelt gjort de nødvendige endringene. |

## 2.5 KVALITETSMESSIGE KRAV

Server/hostingtjeneste skal være tilgjengelig til enhver tid, dvs at den skal ha høyest mulig oppetid. Dette er viktig da løsningen inneholder sanntids-auksjoner.

Nettsiden og dens funksjonalitet skal respondere raskt for å gi en behagelig brukeropplevelse. Dersom dette ikke er mulig skal brukeren varsles om at nettstedet arbeider.

## 2.6 DETALJERTE USE CASE BESKRIVELSER

| Use Case                           | Logg inn   |
|------------------------------------|--|
| <b>Aktør</b>                       | Bruker   |
| <b>Pre-betingelser</b>             | Ingen  |
| <b>Post-betingelser</b>            | Brukeren får tilgang til annen funksjonalitet.   |
| <b>Trigger</b>                     | Brukeren velger lenken «Logg inn».   |
| <b>Normal hendelsesflyt</b>        | <ol style="list-style-type: none"> <li>1. Pop-up vinduet med login-skjema åpnes.</li> <li>2. Bruker taster inn sitt brukernavn og passord.</li> <li>3. Joomla! login-modul sammenligner brukerdata mot database.</li> <li>4. Bruker logges inn på nettsiden ved godkjenning av brukerdata.</li> <li>5. Funksjonalitet for innloggede brukere gjøres tilgjengelig.</li> </ol> |
| <b>Variasjoner/feilsituasjoner</b> | <ol style="list-style-type: none"> <li>2. Bruker taster feil brukernavn og/eller passord. <ul style="list-style-type: none"> <li>- Bruker videresendes til en ny login-side og får beskjed om hendelsen.</li> <li>- Prosessen gjentas fra trinn 2.</li> </ul> </li> </ol>  |



| Use Case                    | Brukerregistrering  |
|-----------------------------|---|
| Aktør                       | Bruker  |
| Pre-betingelser             | Bruker ønsker å registrere seg som budgiver eller selger.   |
| Post-betingelser            | En brukerkonto opprettes og bruker logges inn.  |
| Trigger                     | Bruker trykker registreringslenken ved budfunksjonaliteten i en auksjon.  |
| Normal hendelsesflyt        | <ol style="list-style-type: none"> <li>1. Bruker angir gyldig informasjon. Dette er fullt navn, brukernavn, passord, gjentakelse av passord, e-postadresse, samt en automatisk generert sikkerhetskode som vises på skjermen.</li> <li>2. Registreringsskjemaet testes for den angitte informasjonen.</li> <li>3. En brukerkonto opprettes dersom gyldige data er angitt.</li> <li>4. Bruker logges inn.</li> </ol> |
| Variasjoner/feilsituasjoner | <p>2. Angitt informasjon er ugyldig eller mangelfull.</p> <p>- Brukeren påminnes hvilke felter som er mangelfuller eller ugyldige.</p>  |

| Use Case         | Se liste over auksjoner           |
|------------------|-----------------------------------|
| Aktør            | Bruker                            |
| Pre-betingelser  | Ingen                             |
| Post-betingelser | Ingen                             |
| Trigger          | Menyenlenken «Eiendommer» velges. |

|                             |   |
|-----------------------------|---|
| Normal hendelsesflyt        | <ol style="list-style-type: none"> <li>1. Nødvendig informasjon om alle auksjoner som ikke er utgått hentes fra databasen.</li> <li>2. En liste over auksjonene genereres og vises for brukeren.</li> </ol> |
| Variasjoner/feilsituasjoner | Ingen   |

| Use Case                    | Se auksjon   |
|-----------------------------|--|
| Aktør                       | Bruker og administrator  |
| Pre-betingelser             | Ingen  |
| Post-betingelser            | Ingen  |
| Trigger                     | Brukeren velger en auksjon fra listen over auksjoner eller taster inn auksjonens URL.  |
| Normal hendelsesflyt        | <ol style="list-style-type: none"> <li>1. Detaljert informasjon om den aktuelle auksjonen hentes fra databasen.</li> <li>2. En visning genereres og vises for brukeren.</li> </ol> |
| Variasjoner/feilsituasjoner | Ingen  |

| Use Case         | Se/bestille produkter                          |
|------------------|--|
| Aktør            | Bruker   |
| Pre-betingelser  | Ingen  |
| Post-betingelser | Bestilling sendes til administrator på e-post. |
| Trigger          | Menyenken «Produkter» velges.                  |

|                             |   |
|-----------------------------|---|
| Normal hendelsesflyt        | <ol style="list-style-type: none"> <li>1. Overordnet informasjon om tilgjengelige produkter hentes fra databasen.</li> <li>2. En liste over produkter genereres for brukeren, samt en felles bestillingslenke.</li> <li>3. Brukeren velger et produkt i listen for å få mer detaljert informasjon om produktet.</li> <li>(4). Bestillingslenke velges for å bestille et produkt. <ol style="list-style-type: none"> <li>(4.1). Bestillingsskjema genereres for brukeren.</li> <li>(4.2). Bruker angir personlig informasjon og ønsket produkt.</li> <li>(4.3). Bestilling sendes til administrator per e-post.</li> </ol> </li> </ol> |
| Variasjoner/feilsituasjoner | <p>(4.2). Bruker angir falsk informasjon i bestillingen.</p> <p>- E-post vil fremdeles sendes til administrator. Administrator vil kontrollere e-posten for ugyldig informasjon.</p>  |

| Use Case             | Egne brukersider   |
|----------------------|--|
| Aktør                | Bruker   |
| Pre-betingelser      | Bruker er logget inn.  |
| Post-betingelser     | Ingen  |
| Trigger              | Menylenke «Min Side» velges.   |
| Normal hendelsesflyt | <ol style="list-style-type: none"> <li>1. Informasjon innhentes fra databasen om brukerens budhistorikk, abonnementer, kjøp, samt eventuelle annonser brukeren har opprettet.</li> <li>2. Visning genereres og vises til brukeren, med muligheter for å</li> </ol> |

|                                    |       |
|------------------------------------|-------|
| redigere annonser og abonnementer. |       |
| Variasjoner/feilsituasjoner        | Ingen |

| Use Case                    | Følge budrunde på e-post/sms  |
|-----------------------------|---|
| Aktør                       | Bruker  |
| Pre-betingelser             | Bruker er logget inn.   |
| Post-betingelser            | Abonnement for brukeren opprettes i databasen.  |
| Trigger                     | Brukeren velger lenken for å følge auksjonen i en gitt auksjon.   |
| Normal hendelsesflyt        | <p>1. Et vindu åpnes for brukeren hvor det angis om auksjonen skal følges på e-post eller sms.</p> <p>(1.1). Bruker angir telefonnummer dersom sms abonnement ønskes.</p> <p>2. Abonnementet registreres i databasen.</p> |
| Variasjoner/feilsituasjoner | <p>1. Bruker forsøker å abonnere på auksjon uten å være innlogget.</p> <p>- Bruker påminnes at han/hun ikke er logget inn.</p>  |

## 2.7 SUPPLEMENTÆRE KRAV

Følgende kommer lavere prioriterte momenter til systemet. Det vil si at de ikke er krav i den forstand at de er nødvendige i en fungerende løsning, men bør derimot være med i det komplette verktøyet som gjør eiendomsmegler overflødig.

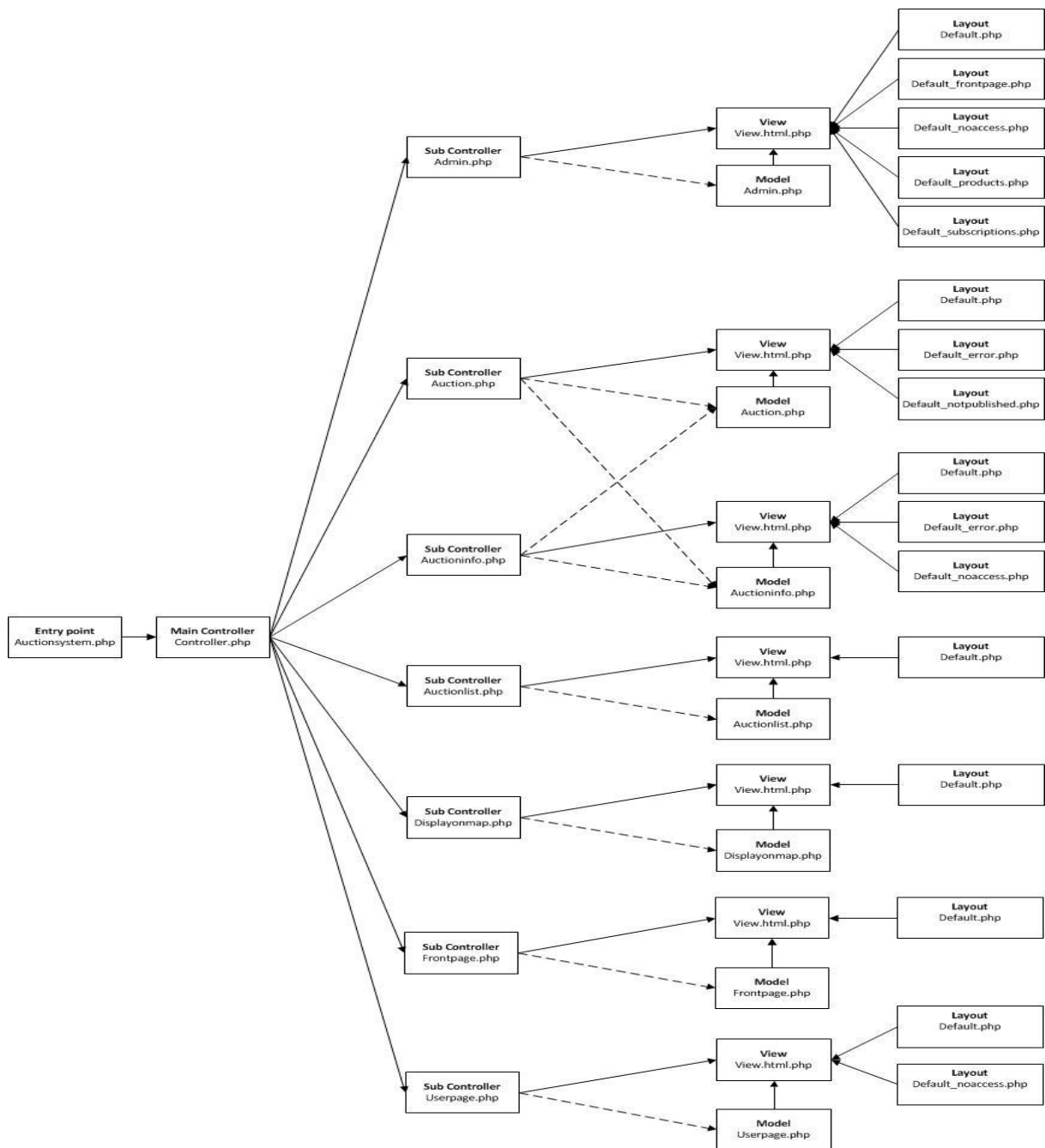
Løsningen bør inneholde:

- Koblinger mot nødvendige instanser (Tinglysingen, Infoland o.l.) for å samle inn data om eiendommer.
- Funksjonalitet for selger slik at han/hun kan bestille fotograf, takstmann osv.
- Verktøy som lar selger av eiendom generere egen avisannonse.
- Verktøy der selger kan holde oversikt over interessenter av eiendommen. Dette gjelder status for salget, antall nedlastninger av prospekt, besøkende på annonsen ol.
- Supportfunksjon mot BoligHandel1 (direktesamtaler/chat).
- Flexibilitet med tanke på et internasjonalt marked. Det vil for eksempel si tilpassede løsninger iht andre lands lovgivninger.

### 3. DESIGN

Systemet baseres på Joomla! CMS. Vi har derfor ikke designet arkitekturen i systemet selv, men har måttet forholde oss til hvordan Joomla! har løst dette. Dette er nærmere beskrevet i kapittel fire.

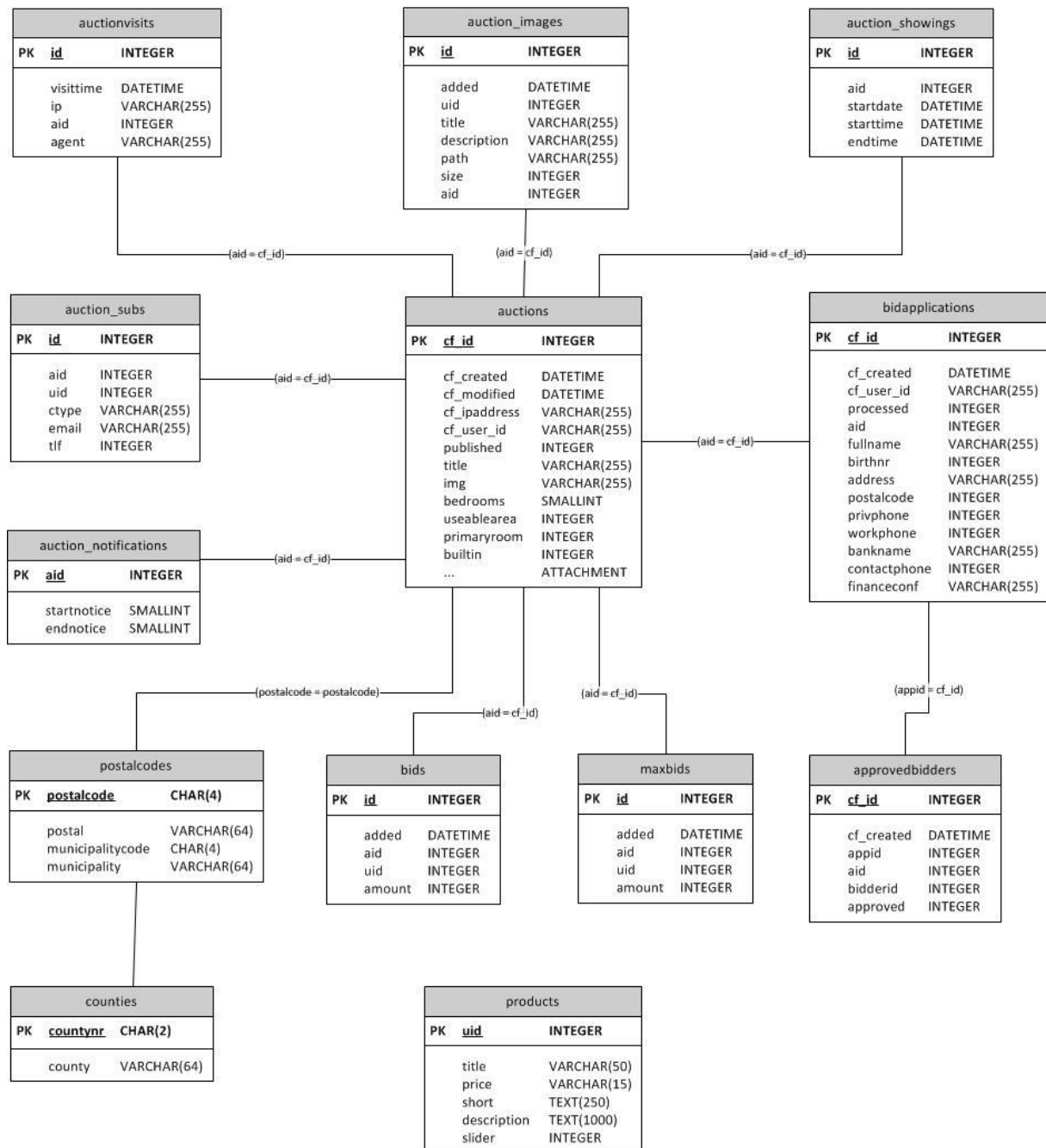
#### 3.1 KOMPONENT DESIGN



Figur. 3.1.1 Auksjons-komponentens struktur.

Figuren viser de forskjellige bestandelene i auksjonskomponenten. De heltrukne pilene viser en direkte assosiasjon, mens de stiplede viser en indirekte assosiasjon. For en nærmere beskrivelse av komponentens virkemåte og dens bestandeler, se kapittel fire.

### 3.2 DATA VIEW



Figur. 3.2 Databasemodell

Figuren viser de aktuelle tabellene for å ta vare på relevant informasjon. Dette er tabeller som ikke medfølger Joomla!-installasjonen, men som er nødvendige for de øvrige delene av systemet. Tabellen «auctions» er kun et utdrag av den totale tabellen da denne inneholder svært mye informasjon. En kort beskrivelse av tabellene følger:

- «Auctionvisits» benyttes til å ta vare på besøksinformasjon til en gitt auksjon. Grunnen til dette er at statistikk kan genereres dersom ønskelig.
- «Auction\_images» benyttes til å ta vare på bilder til auksjonene.
- «Auction\_showings» lagrer alle visninger som er lagt til i de ulike auksjonene.
- «Auction\_subs» tar vare på alle abonnementer til auksjonene slik at applikasjonen vet hvilke brukere som skal varsles ved hendelser i auksjonene.
- Kjernen i informasjonsstrukturen er tabellen «auctions». Denne tar vare på all informasjon om auksjonene som ikke er naturlig å legge i andre tabeller.
- «Bidapplications» lagrer alle søknader om godkjenning som budgiver.
- «Auction\_notifications» benyttes til å holde oversikt over hvilke auksjoner som er startet og avsluttet og hvorvidt det er sendt ut varslinger om dette.
- «Postalcodes» inneholder alle postnumre og poststeder i Norge, samt deres kommuner med kommunenummer.
- «Counties» inneholder alle norske fylker. Sammenhengen mellom postnummer tabellen og fylke tabellen er at de to første tallene i kommunenummeret tilsvarer fylkesnummeret.
- «Bids» lagrer alle gjeldende bud til de ulike auksjonene.
- «Maxbids» inneholder budgiverens maks-bud i de ulike auksjonene.
- «Products» lagrer alle tilgjengelige produkter.
- «Approvedbidders» lagrer alle behandlede budgiversøknader.



## 4. REALISERING

### 4.1 CMS

Vi måtte fra begynnelsen av bestemme oss for om vi ønsket å benytte et Content management system på løsningen våres. Ingen av oss hadde noe erfaring med CMSer fra før. Vi satte derfor av litt tid til å lese om Content Management Systemer for å se på fordelene og ulempene.

Trendene peker mot at en større og større andel av alle nettsider benytter en eller annen form for CMS. Den 1. april 2010 benyttet kun 17,6 % av alle nettsider ett CMS, 1. januar 2012 var dette tallet oppe i 29 %[1].

#### **Fordeler ved bruk av CMS**

- Tidkrevende å oppdatere statiske sider
- Vi slipper å utvikle masse funksjonalitet som følger med et CMS
- Optimalisert før søkemotorer
- Det er lettere for oppdragsgiver å bygge videre på siden med ferdige Extensions og lignende
- Oppdragsgiver har ingen erfaring med webutvikling, det vil derfor bli lettere for dem å vedlikeholde og oppdatere siden.
- Oppdragsgiver har allerede erfaring med bruk av CMS (Joomla)
- Ferdige Extensions kan spare oss for mye utviklingstid

#### **Ulemper**

- Mye funksjonalitet og ny kode å sette seg inn i
- Bruker mye ressurser

Siden fordelene utveide ulempene i såpass stor grad, følte vi oss ganske sikre på at det ville lønne å benytte et CMS. Vi følte at dette ville spare oss for mye utviklingstid siden vi mer eller mindre ville blitt nødt til å utvikle mye av funksjonaliteten du finner i et CMS selv dersom vi hadde valgt å kode alt fra bunnen.

#### 4.1.1 VALG AV CMS

Det finnes utallige forskjellige CMS systemer å velge mellom. Vi valgte å gjøre det enklere for oss selv ved og kun fokusere på de mest populære gratis plattformene.

Under har vi tatt med en oversikt over de mest 5 mest brukte CMS systemene på markedet. 69,7 % av alle nettsider benytter ikke et CMS system. Wordpress brukes på 16,4 % av alle nettsider på nettet og har en CMS markedsandel på 54,1 %<sup>[2]</sup>.

| # CMS            | Websider som bruker dette (Totalt / CMS andel) |
|------------------|--|
| <b>WordPress</b> | 16,4 % / 54,1 %                                |
| <b>Joomla</b>    | 2,8 % / 9,2 %                                  |
| <b>Drupal</b>    | 2,0 % / 6,7 %                                  |
| <b>vBulletin</b> | 1,2 % / 3,8 %                                  |
| <b>Blogger</b>   | 0,9 % / 3,1 %                                  |

Tallene over sier ingen ting om bruksområdet til de forskjellige CMS systemene. Wordpress er desidert det mest brukte CMS systemet, men dette betyr ikke nødvendigvis at dette er det riktige valget for oss. Vi må også anta at en stor andel av Wordpress sidene på denne oversikten er blogger, et marked som virkelig har eksplodert de siste årene.

Vi valgte derfor å se litt nærmere på bruksområdet til de forskjellige CMS systemene. Vi fant grafiske fremstillinger over hvilke typer sider de forskjellige plattformene brukes mest til<sup>[3]</sup>. Fordelingen var relativt lik for alle plattformene så dette ga oss ikke noe bedre grunnlag for å foreta ett valg.

Etter og ikke å ha blitt særlig mye klokere av å lese om CMSene bestemte vi oss for å installere og teste de forskjellige CMSene opp mot hverandre. Etter en dag med testing konkluderte vi med at Wordpress virket som det absolutt mest brukervennlige CMS systemet, men det virket ikke like «komplett» som de andre. Joomla og Drupal virket langt mer komplekse og hadde mye mer funksjonalitet og innstillingsmuligheter.

Etter å ha lest om fordeler og ulemper av de forskjellige CMS systemene virker det ganske likegyldig hva vi gikk for å bruke. Alle virket veldig solide med et solid arsenal av gratis extensions og templates tilgjengelig. Det kom veldig tydelig frem i mye av det vi leste at Joomla brukere «elsker Joomla» og at Drupal brukere «elsker Drupal». Vi fikk også inntrykket av at Wordpress ikke var like fleksibelt og utviklervennlig som de to andre alternativene. Selv om dette er noe som skal ha bedret seg betydelig den siste tiden.

Vi benyttet også cmsmatrix [4] for å sammenligne funksjonaliteten til de tre CMS systemene.

Etter å ha undersøkt det som var verdt å undersøke følte vi ikke at vi hadde nok kunnskap om webutvikling eller plattformene for å kunne ta et godt overveid valg. Vi valgte så å sende en mail til oppdragsgiver for å høre deres mening. De ønsket at vi skulle benytte Joomla siden de benytter Joomla på sin nåværende side.

Etter en kort intern diskusjon kom vi fram til at vi skulle benytte Joomla.

## 4.2 HVORDAN FUNGERER JOOMLA

Den følgende teksten skal beskrive i korte trekk hvordan Joomla! fungerer og hva det baserer seg på. Det er på ingen måte en detaljert fremleggelse av hva Joomla! er og hva det kan gjøre, men en kort innføring i tekniske løsninger og bruk av Joomla!.

### 4.2.1 INTRODUKSJON – HVA ER JOOMLA! ?

Skal man lage en side på web, uten at man har kunnskap om html, php, JavaScript og lignende, vil et CMS (Content Management System) være svært aktuelt. Joomla! er nettopp et slikt CMS og vektlegger nettopp det å være enkelt å bruke, samtidig som det ikke skal begrense seg for profesjonelle utviklere. Joomla! er altså et hjelpemiddel som strukturerer innholdet for deg som skal lage en side på web, uansett om du er nybegynner eller profesjonell.

Joomla! er et gratis GPL-lisensiert (GNU General Public License) Content Management System (CMS), oversettes til Publiseringverktøy på norsk, som baseres på åpen kildekode, objekt orientert programmering og designmønsteret Model-View-Controller (MVC).

Inkluderer funksjonalitet som "sidecaching", RSS feed, mulighet for å printe ut sider, nyhetsflasher, blogger, spørreundersøkelser, søkemotor og støtte for internasjonalisering. Joomla er kodet i PHP og benytter seg av en database for lagring av data.

Frem til versjon 2.5, kunne man bare benytte seg av MySQL i Joomla. Men nå kan også MS SQL benyttes[5].

Vi vil gå nærmere inn på hvordan Joomla fungerer i detalj på de neste sidene, men i hovedsak er tanken bak Joomla (og andre CMS) at man skal kunne oppdatere og administrere fra hvor som helst og når som helst via en datamaskin med nettilgang.

Joomla deles opp i en frontend og en backend. Frontend er selve nettsiden slik den skal vises til publikum med all dens funksjonalitet, informasjon, artikler og annet siden måtte tilby. Backend er hvor man administrer og endrer innhold. Samt legger til ny funksjonalitet og velger hvordan siden (frontend) skal vises frem. Denne måten gjør at bruk av spesiell programvare blir unødvendig, og som nevnt tidligere, at man kan logge seg på og endre fra hvor som helst, så lenge maskinen har nettilgang.

---

#### 4.2.2 HISTORIE

Joomla 1.0 ble utgitt den 16. september 2005, som en slags relansering av Mambo. Et annet publiseringverktøy som baserer seg på åpen kildekode. Teamet bak Joomla er det samme som stod bak den siste versjonen av Mambo (som var 4.5.2.3.). Den første Joomla-versjonen hadde for øvrig tilnavnet Sunrise.

Årsaken til navnendringen kom som et resultat av en uoverensstemmelse mellom Miro Corporation, Trademarkernehaverne av Mambo-navnet, og de som i sin tid startet Mambo CMS som en åpen kildekode-løsning.

Versjon 1.5 ble lansert 22. januar 2008. Dette er den første versjonen som fikk såkalt 'Long Term Support' (LTS). LTS versjoner utgis ved hver tredje store eller mindre utgivelser og vil være støttet inntil tre måneder etter den neste LTS versjonen er utgitt.

Versjon 1.6 ble utgitt 10. januar 2011. Med denne versjonen kom "Access Control List" (ACL), som i denne sammenhengen kan sees på som en liste med tilgangsnivåer som er tilknyttet et objekt. Dvs. mulighet for at en administrator har tilgang til funksjonalitet en registrert bruker ikke har. Og videre at en registrert bruker har tilgang til for eksempel artikler en tilfeldig gjest som er innom siden ikke har. ACL er mye mer enn disse to eksemplene, men de viser litt til hva ACL er.

I tillegg fikk versjon 1.6 også muligheten for at en bruker kunne definere et hierarki av kategorier, samt at det var store forbedringer på administratorgrensesnittet.

Versjon 1.7 ble lansert 19. juli, 2011. Seks måneder etter 1.6 kom ut. I denne versjonen var det primært sikkerheten og sikkerhetsfunksjonalitet som var vektlagt.

Versjon 2.5 ble lansert 24. januar, 2012, seks måneder etter 1.7. Dette er den neste LTS-versjonen etter 1.6. Denne utgivelsen skulle egentlig navngis 1.8, men ble endret for å tilpasses et nytt navngivelsessystem for versjoner hvor hver LTS-utgivelse skal være en x.5-versjon. Denne versjonen er også den første som kan benytte seg av andre relasjonsdatabaser enn MySQL.

Neste utgivelse av Joomla (3.0) skal etter planen utgis i september 2012. I utgangspunktet skulle den vært utgitt Juli 2012, men på grunn av at frivillige var ukomfortable med Januar/Juli-mønster, ble den endret til et September/Mars-mønster[6].

---

#### 4.2.3 KORT FORKLARING AV DE TEKNISKE LØSNINGENE OG BEGREPENE I JOOMLA

HVA ER ET CONTENT MANAGEMENT SYSTEM (CMS) ?

Et Content Management System eller Publiseringverktøy på norsk, omhandler publiseringer, endringer, utvikling, vedlikehold og redigering av innhold på nettsider.

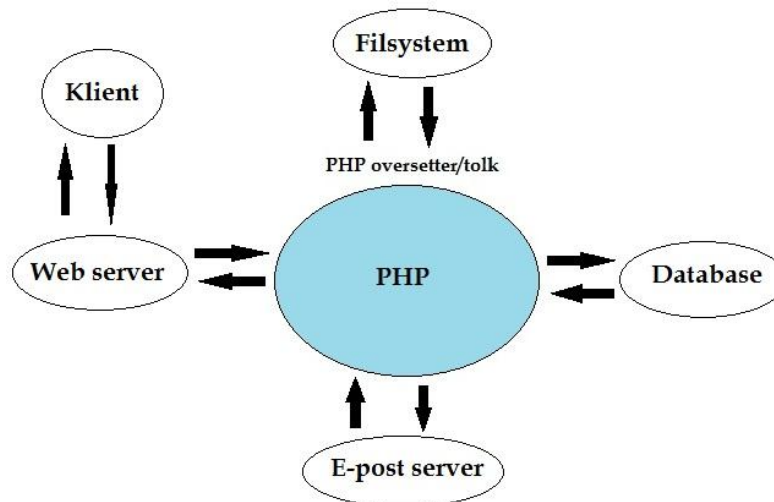
Inneholder verktøy for å lage, strukturere og presentere websidene. I tillegg finnes det ofte utvidelser som gjør at man kan få spesiell funksjonalitet på sin side[7, 8].

### **Hovedtrekk**

Kjernefunksjonaliteten i de forskjellige CMS vil variere en del, men under er det listet opp en del punkter som mer eller mindre gjelder for større CMS-systemer:

- Tillater en stor mengde mennesker å dele og bidra til lagret data.
- Kontrollerer tilgangen til forskjellige data via brukertilgang (tidligere nevnte ACL).
- Gjør det enkelt å lagre og hente data
- Validitet av data og samsvar data
- Reduserer dobbeltlagring
- Gjør rapportskrivning enkelt
- Forbedrer kommunikasjon mellom brukere
- Definerer data som hva som helst (dokumenter, bilder, tekst, video, artikler, m.mer)
- Hurtig fornyelse av innhold
- Bedre konsistens og struktur
- Enklere navigasjon
- Økt fleksibilitet
- Høyere sikkerhet og beskyttelse av innhold for mindre datakyndige personer
- Høy kapasitet for vekst
- Lavere vedlikeholdskostnader

## HVA ER PHP



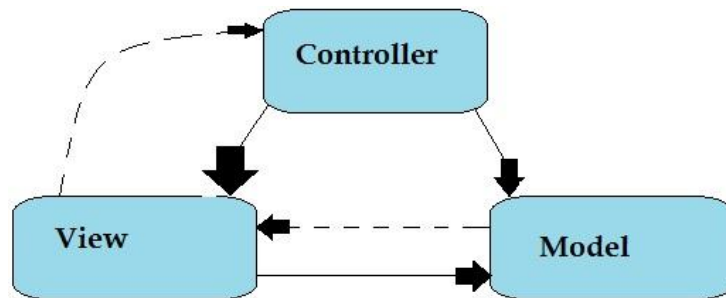
Figur 4.2.1. Bildet illustrerer hvordan PHP fungerer i bruk.

- Forkortelse for PHP: Hypertext Preprocessor.
- PHP er et skriptsspråk som egner seg spesielt for webutvikling. Det kan brukes direkte i HTML.
- Meste av syntaksen er lånt ifra C, Java og Perl, men med noe unik PHP-spesifikk syntaks.
- Brukes når man vil lage dynamiske websider. Ordet Hypertekst er en tekst med linker man kan klikke på, og en pre-prosessor er et program som bearbeider teksten før den vises frem[9].

## HVA ER MYSQL OG RELASJONSDATABASE

MySQL er en relasjonsdatabase basert på SQL (Standard Query Language) eller Standard Spørrespråk på norsk. I en relasjonsdatabase kan du lagre informasjon (data) i tabeller, i rader, og sortere data via kolonner. Lagret slik at man enkelt ved hjelp av spørringer (Query) kan hente data ut igjen ved behov[10].

## HVA ER MODEL-VIEW-CONTROLLER (MVC)



Figur 4.2.2. Bildet illustrerer et enkelt MVC-konsept. De heltrukne linjene representerer en direkte assosiasjon, mens de stiplede linjene en indirekte assosiasjon (via for eksempel et brukergrensesnitt).

I programutvikling er Model-View-Controller et designmønster som deler samspillet mellom brukere og applikasjoner i tre roller:

- Modellen (Forretningslogikk)
- Utseendet (Brukergrensesnittet)
- Kontroller (Input fra bruker)

Man kan se på kjernen ved MVC, og også det som har hatt mest betydning for senere rammeverk, som det man kan kalle separat presentasjon. Ideen er å lage en klar grense mellom de objektene vi bruker for å modellere vår oppfatning av den virkelige verden, og det å presentere objektene på skjermen. Dette forstås med at modellen i vårt tilfelle, skal kunne fungere som et selvstendig objekt uten noe form for referanse til selve presentasjonen. Modellen er også designet slik at den støtter flere forskjellige presentasjoner, og gjerne samtidig[11].

### Model

Modellen sin arbeidsoppgave er å tilby de nødvendige dataene som et View skal benytte seg av. I Joomla! og vårt prosjekt handler dette stort sett opp om å hente ut data i fra databasen, og levere dette i ønsket format vil et View. Fordelen med å ha en selvstendig modell, er at flere View kan benytte seg av samme modell. *Modellen* håndterer væremåten og dataene til



applikasjonens område (eller domene), reagerer på forespørsler om dets tilstand, som regel ifra et View, og reagerer på forespørsler om å endre tilstand, som regel ifra Controlleren. I handlingsbaserte systemer gjør modellen Viewet oppmerksom på at informasjonen nå er endret, så Viewet kan reagere og oppdatere hva som vises til bruker.

## **View**

*Viewet* viser modellen i en passende form for samhandling, typisk et brukergrensesnitt. I Joomla! og vårt prosjekt benytter et View seg av funksjoner som ligger i modellen for å hente ut de data og strukturere disse i brukergrensesnittet, altså på skjermen. Flere View kan eksistere for en modell til forskjellige formål. Et View har typisk en til en-kommunikasjon med en fremviser (display surface) og vet hvordan det skal vise frem informasjon gjennom det.

## **Controller**

*Controlleren* mottar input ifra bruker og utfører en handling basert på inputen ved å utføre kall til Modellen. Controlleren tar imot input fra bruker og instruerer modellen og Viewet til å utføre handlinger basert på inputen. Slik at det som vises til bruker oppdateres og viser ny informasjon basert på den utførte handlingen.

## **Generelt**

På denne måten separerer man data og presentasjon slik at for eksempel datahåndteringen ikke endres av endringer i brukergrensesnittet og vice versa.

MVC kommer i forskjellige varianter, men i hovedsak fungerer det slik:

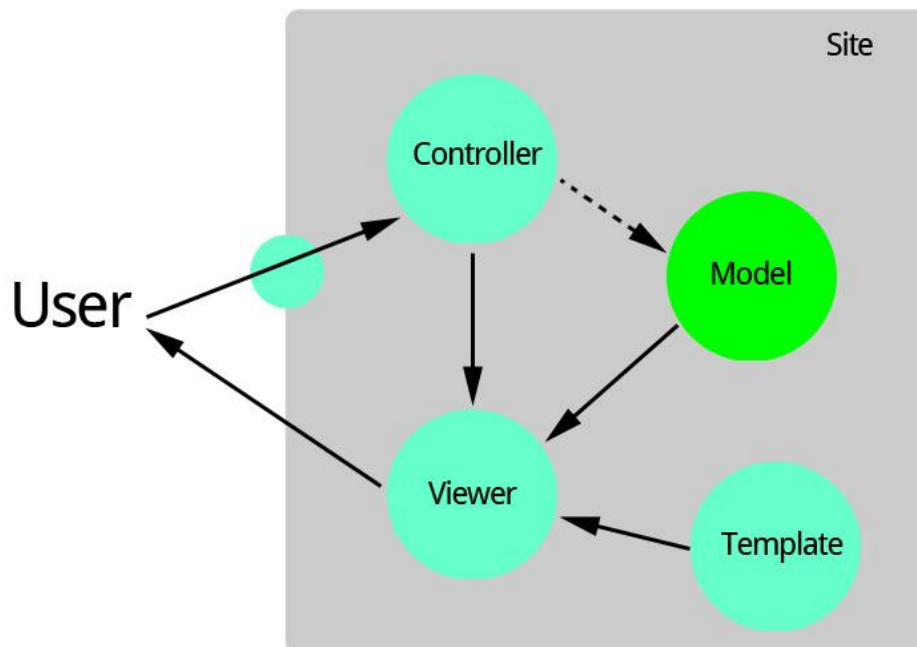
- Brukeren har et eller annet samspill med brukergrensesnittet, for eksempel trykker på en knapp på musa.
- Controlleren håndterer inputen fra brukergrensesnittet og konverterer hendelsen til en passende brukerhendelse som modellen forstår.
- Controlleren gjør modellen oppmerksom på brukerhendelsen som muligens vil endre modellens tilstand. (For eksempel oppdaterer en handlekurv eller liknende.)

- Brukergrensesnittet utfører en spørring til modellen for å lage et passende grensesnitt (i vårt tilfelle en oppdatert handlekurv på selve siden). Akkurat dette steget kan variere litt med tanke på hvem som trigger oppdateringen av grensesnittet.
- Brukergrensesnittet venter på neste handling fra bruker, som igjen starter hele syklusen.

En applikasjon (deler eller en funksjonalitet på en side) basert på et MVC-mønster kan være en samling av MVC-trekanter (se figur), som hver har ansvar for forskjellige deler av brukergrensesnittet.

### MVC i Joomla

I Joomla har MVC blitt implementert på en litt annen måte enn den generelle oppfatningen. I tillegg til Modellen, Viewet og Kontrolleren har det blitt lagt til ett «Entry point» og en «Viewer» (et view).



Figur 4.2.3 Den solide linjen representerer en direkte assosiasjon, den prikkede linjen indikerer en indirekte assosiasjon.

## 4.2.4 JOOMLA I PRAKSIS

### JOOMLA! BACKEND

Backend er som nevnt tidligere den delen av Joomla der man kan endre på hele oppbyggingen av selve siden. Alt fra registrerte brukere, artikler, kategorier artikler ligger i, menyer, plugins, egne og andres utvidelser med mye mer. Vi skal nå gå igjennom litt hvordan administratorsiden ser ut, og hva de enkelte delene gjør.

### OVERBLIKK ADMINISTRATORSIDEN

The screenshot shows the Joomla! Administrator interface for 'BoligHandel'. The top navigation bar includes 'Dashboard', 'Users', 'Articles', 'Menus', 'Extend', 'Configure', and 'Help'. A 'Statistics Overview' sidebar on the right displays the following data:

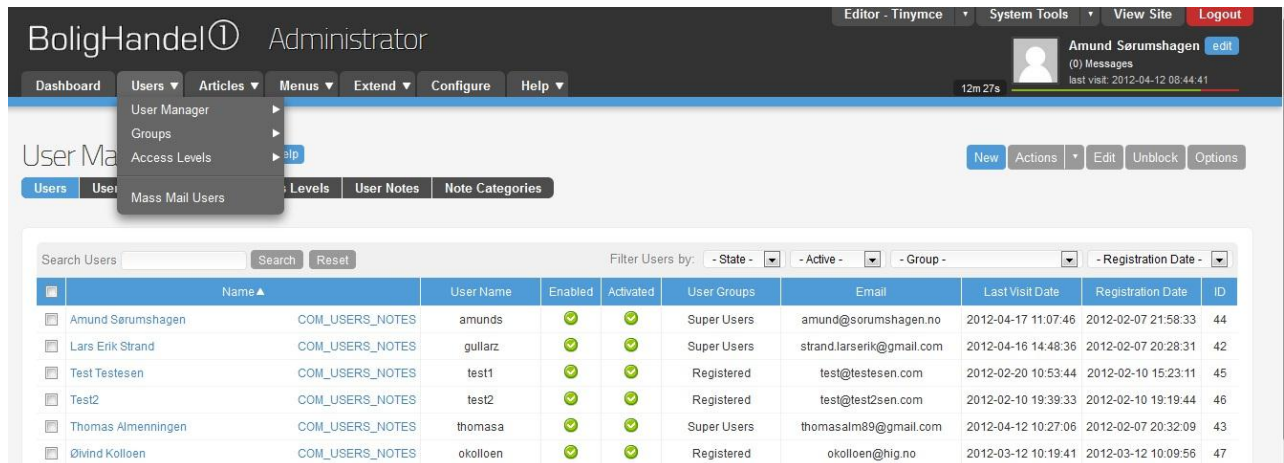
| Metric                  | Value |
|-------------------------|-------|
| Current Active Users    | 0     |
| Active Guests           | 0     |
| Active Registered       | 0     |
| Unique Visits Today     | 2     |
| Unique Visits Yesterday | 0     |
| Visits This Week        | 12    |
| Visits Previous Week    | 2     |

Figur 4.2.4. Administratorsiden

Først må det nevnes at selve utseendet på både frontend (siden) og backend (administrator) velges ut ifra en mal. Installasjonen av Joomla 2.5 inneholder 5 maler for selve siden, og 3 maler for administrator-delen. Man kan lage egne maler, men det er store muligheter for å sette sitt eget preg på en allerede eksisterende mal. Som vi ser har også administratorsiden (backend) flere menyer for å kategorisere de endringene og tilpasningene man kan utføre.

Selv om de fleste templatene på administratorsiden (backend) er bygd opp relativt likt, vil denne detaljerte gjennomgangen av Joomla Backend gjelde spesifikt for vårt prosjekt. Men det vil gi en pekepinn på hvordan ting vil fungere i backend.

## USERS (BRUKERE)



The screenshot shows the Joomla! Administrator interface. The top navigation bar includes 'Dashboard', 'Users', 'Articles', 'Menus', 'Extend', 'Configure', and 'Help'. The 'Users' menu is open, showing options like 'User Manager', 'Groups', 'Access Levels', and 'Mass Mail Users'. Below the menu, there is a search bar and a table of users.

| Name               | User Name | Enabled | Activated | User Groups | Email                     | Last Visit Date     | Registration Date   | ID |
|--------------------|-----------|---------|-----------|-------------|---------------------------|---------------------|---------------------|----|
| Amund Sørumshagen  | amunds    | ✓       | ✓         | Super Users | amund@sorumshagen.no      | 2012-04-17 11:07:46 | 2012-02-07 21:58:33 | 44 |
| Lars Erik Strand   | gullarz   | ✓       | ✓         | Super Users | strand.larsenik@gmail.com | 2012-04-16 14:48:36 | 2012-02-07 20:28:31 | 42 |
| Test Testesen      | test1     | ✓       | ✓         | Registered  | test@testesen.com         | 2012-02-20 10:53:44 | 2012-02-10 15:23:11 | 45 |
| Test2              | test2     | ✓       | ✓         | Registered  | test@test2sen.com         | 2012-02-10 19:39:33 | 2012-02-10 19:19:44 | 46 |
| Thomas Almenningen | thomasa   | ✓       | ✓         | Super Users | thomasalm89@gmail.com     | 2012-04-12 10:27:06 | 2012-02-07 20:32:09 | 43 |
| Øivind Kolloen     | okolloen  | ✓       | ✓         | Registered  | okolloen@hig.no           | 2012-03-12 10:19:41 | 2012-03-12 10:09:56 | 47 |

Figur 4.2.5. Brukeroversikt

Som vi får et inntrykk av ifra den overliggende menyen, kategoriserer man altså administratorsiden etter hva man ønsker å gjøre. Menyen vi nå har åpnet handler om det meste som omhandler registrerte brukere. Som kan være alt fra administratorer, spesielle brukere og til vanlige registrerte brukere.

Joomla har forhånds-lagde menyer (kalt Manager), hvor man vil få presentert en liste med alle aktuelle data for den kategorien. Den listen vi ser i dette bildet er et utdrag av "User Manager".

*User Manager* tar seg av alt som er direkte knyttet til en bruker. I denne menyen kan man legge til nye brukere, endre brukere og slette brukere.

I tillegg kan man knytte eksisterende brukere opp mot grupper og vi de et tilgangsnivå som begrenser tilgang til informasjon på selve siden (frontend) eller gir de utvidede rettigheter som gjør at de kan for eksempel legge til eller endre egne artikler på siden (frontend).

*Groups* brukes til å opprette grupper man kan knytte brukere opp imot. Dette for å gruppere registrerte brukere for eksempel for tilgangskontroll. *Administratorer, Skribenter og Brukere* kan være tre eksempler på grupper med forskjellig tilgangskontroll.

*Access Levels* brukes for å sette tilgangsnivået til en bruker. Joomla! er bygget opp slik at forskjellige brukere har forskjellige tilgangsnivåer. For eksempel kan man differensiere brukere etter de som bruker siden gratis, eller de som betaler en sum i året. Da vil innholdet

kunne tilpasses de forskjellige tilgangsnivåene slik at de som bruker siden gratis ikke har tilgang til all informasjonen de som betaler har. Ved å ha tilgang på administratornivå, vil man for eksempel kunne sette det opp slik at administratorer vil kunne legge ut artikler eller annet innhold via Frontend. På denne måten vil siden kunne oppleves svært forskjellig ut ifra hvilke tilgangsnivå man har som bruker.

## ARTICLES (ARTIKLER)

The screenshot shows the BoligHandel Administrator interface. The top navigation bar includes 'Dashboard', 'Users', 'Articles', 'Menus', 'Extend', 'Configure', and 'Help'. The 'Articles' menu is expanded, showing options for 'Article Manager', 'Category Manager', 'Featured Articles', and 'Media Manager'. Below the navigation, the 'Article Manager' section is visible, featuring a search filter and a table of articles.

|                          | Title ▲   | Status | Category | Ordering | AC |
|--------------------------|---|--------|----------|----------|----|
| <input type="checkbox"/> | Alle hjerter gleder seg<br>(Alias: alle-hjerter-gleder-seg) | ✔      | News     | 14       | Pt |
| <input type="checkbox"/> | BH1 - Nå større og bedre<br>(Alias: bh1-na-storre-og-bedre) | ✔      | News     | 15       | Pt |
| <input type="checkbox"/> | Brukbart<br>(Alias: brukbart)                               | ✔      | News     | 12       | Pt |
| <input type="checkbox"/> | Enda en nyhet   | ✔      |          |          |    |

Figur 4.2.6 Artikkel behandling

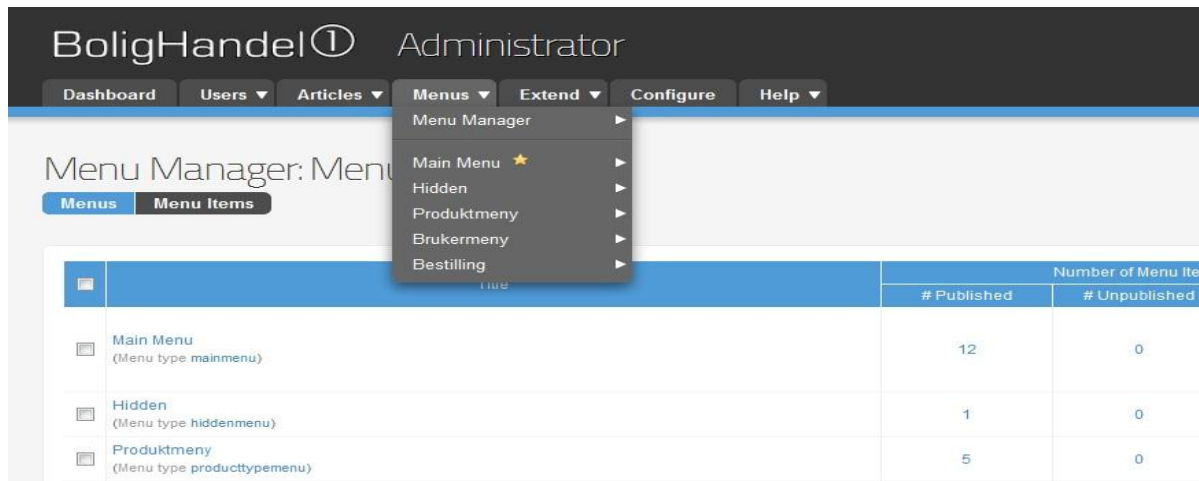
Nå er vi i *Articles*-menyen og listen vi ser utdrag av ligger under *Article Manager*. Som man kan tenke seg er dette menyen som omhandler artikler som ligger i relasjonsdatabasen.

I *Article Manager* ligger en oversikt over alle artikler som ligger i relasjonsdatabasen. En artikkel trenger ikke være publisert, altså å ligge på siden. I *Article Manager* kan man legge til nye artikler, endre innhold på eksisterende eller slette de.

I *Category Manager* ligger kategori-hierarkiet artiklene kan plasseres under. Å kategorisere artikler gjør at det er mulig å vise artikler fra en kategori under et bestemt menyvalg på selve siden (frontend). Hvis vi for eksempel hadde hatt en kategori som het *Biler* kunne et valg *Vise Biler* i menyen på siden vist frem artikler som er lagt under denne kategorien.

*Featured Articles* viser alle artikler som har fått denne *statusen*. Med *featured* mener vi *merket* eller fremvist artikkel. Joomla inneholder funksjonalitet som gjør at disse artiklene kan gis spesiell oppmerksomhet på selve siden (frontend).

## MENUS (MENYER)



Figur 4.2.7. Menyoversikt

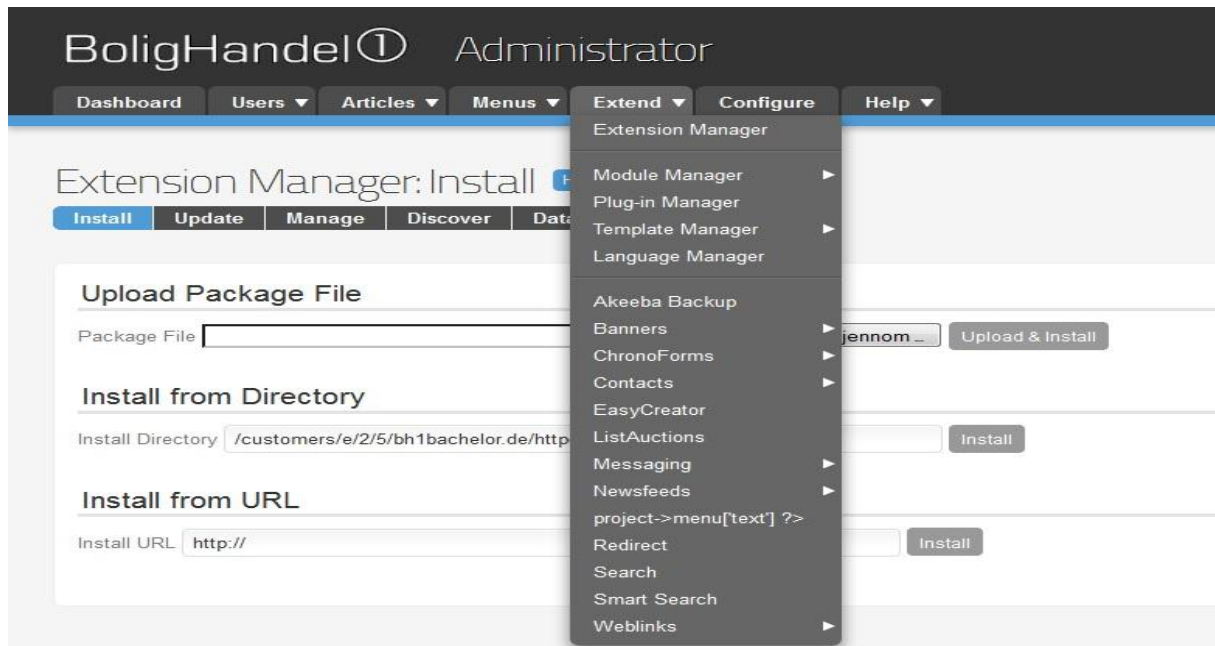
*Menus* og *Menu Manager* er en oversikt over hvilke menyer man har laget. Dette viser hvilken modul menyen er koblet til (moduler forklares senere) og hvor modulen er plassert på siden. En meny inneholder menyvalg og kan inneholde undermenyer hvis ønskelig. Et menyvalg vil som regel inneholde en link til en artikkel eller andre mer spesialiserte sider som kan inneholde annen funksjonalitet.

De andre valgene under *Menu Manager* er egne menyer man har laget.

## EXTENSIONS (UTVIDELSER)

Det som gjør Joomla til et meget kraftig CMS er antall brukere og videre antall utvidelser som er laget. Dette gjør at de fleste vil kunne finne moduler eller komponenter som gjør det de er ute etter uten at de trenger å ha noe kunnskap om programmering eller hvordan Joomla egentlig fungerer under overflaten. Man vil kunne finne kommersielle utvidelser og ikke-kommersielle gratis utvidelser som dekker det meste av hva man måtte ha bruk for.

Det strenge rammeverket Joomla opererer under sørger også for at muligheten for feil i utvidelsene er veldig lave.



Figur 4.2.8. Utvidelser

Bildet viser et utdrag av *Extend*-menyen og underliggende *Extensions Manager*. Via *Extensions Manager* kan man installere utvidelser i form av *komponenter, moduler, plug-ins, maler og andre utvidelser*.

I *Module Manager, Plug-in Manager, Template Manager* og *Language Manager* kan man administrere de utvidelsene man har installert. Dette vil være veldig individuelt fra utvidelse til utvidelse, men i grove trekk vil dette gå ut på å tilpasse alternativene til utvidelsen slik at den mest mulig gir den funksjonaliteten du er ute etter.

## KOMPONENTER

En komponent er egentlig en separat applikasjon. Man kan se på en komponent som noe som har sin egen funksjonalitet, dets egen tabell i relasjonsdatabasen og sin egen presentasjon. Så hvis du installerer en komponent, legger du altså til en applikasjon på siden din. Eksempler på komponenter er forum, nyhetsbrev, fotogalleri, etc. Selv om disse vil inngå i siden din under den malen du har valgt, kan de sees på som separate applikasjoner.

Uttrykket *separate applikasjoner* kan man forstå som noe som godt kunne vært et eget system på en egen webside alene. En komponent vil som sagt vises på selve siden i ditt system og bare en komponent vil vises samtidig. En meny vil da være en bryter mellom de forskjellige komponentene.

En komponent er altså i korte trekk en mulighet til å legge til funksjonalitet på siden[12].

Komponenter er det som står for det meste av innhold på din Joomla-side. En komponent kan ha en eller flere visninger (View) som står for struktureringen av innholdet og hvordan det vises på siden[13]. Joomla! består av komponenter, og alt man ser på en Joomla!-side vises gjennom en komponent.

En komponent i Joomla! er bygget etter designmønsteret MVC (Model-View-Controller). Og hver komponent kan sees på som en MVC-trekant i bruk.

## MODULER

Moduler er utvidelser som presenterer en bestemt bit av informasjon på websiden din. Det er en måte å presentere informasjon som allerede finnes. Dette kan utvide funksjonaliteten til en applikasjon som allerede er en del av siden. For eksempel en login-modul som lar registrerte brukere logge inn på siden og med det få tilgang til mer funksjonalitet eller artikler enn de ville hatt som gjester.

Forskjellen mellom *moduler* og *komponenter* er ikke veldig lett å se i utgangspunktet, men hvis vi drar inn eksemplet fra *komponenter*, ville ikke en modul kunne fungert som et selvstendig system og nettside, da den bare vil presentere informasjon som allerede finnes eller legge til funksjonalitet til en eksisterende applikasjon.

Moduler kan tillegges et bestemt område på en side. For eksempel vil menyer, som nevnt tidligere, måtte legges i en modul, for så å legges til siden. En modul på en Joomla!-side er alt som ligger i områder utenfor innholds-feltet.

Moduler i Joomla! er små programmer som har ansvaret for å vise ulike typer innhold på forskjellige måter. Det finnes svært mange modultyper, og Joomla! inneholder allerede ved



standard installasjon mange moduler man kan bruke som utfører forskjellige typer oppgaver. Alt av menyer, bannere, reklame og som nevnt tidligere en innloggings-boks omtales i Joomla! som moduler[14].

For å vise en modul i Joomla! må den tildeles en posisjon på siden.

## PLUGINS

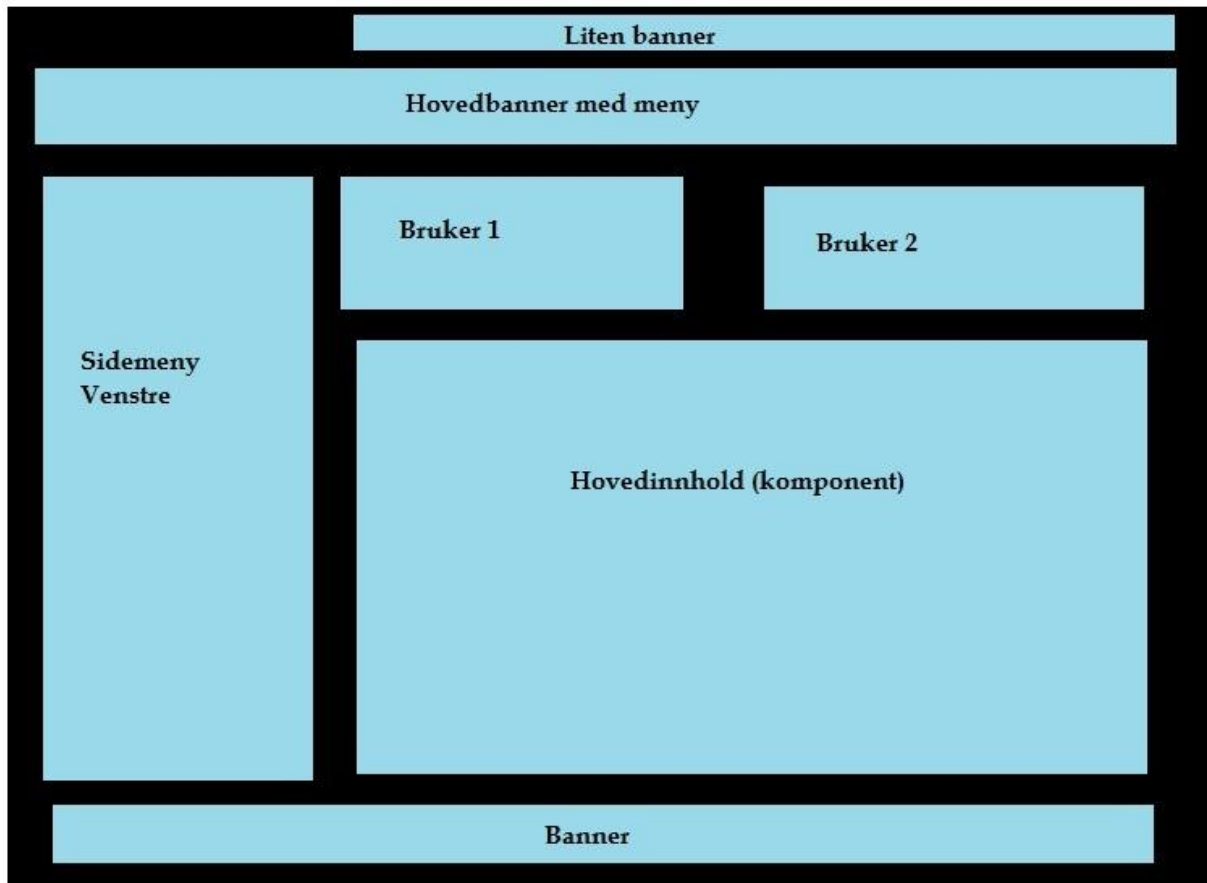
En plugin er en funksjon som utføres på en bit av Joomla! før denne delen vises. Denne biten kan være på selve innholdet, på editoren, på hele systemet, etc. Det gir muligheten til å forbedre og utvikle funksjoner på innhold. For eksempel kan en plugin være bokmerker, handlevogn eller en plugin som viser en logo eller et bilde på steder der man har lagt til en {visbilde}-tag i teksten i en artikkel som vises[15].

## GENERELT

Joomla! er bygget opp av komponenter, moduler og plug-ins. Selv om vi har prøvd å forklare så godt som mulig, er det ikke så lett å se på en Joomla!-side, og se hva som er hva. Gjennom eksemplene som er nevnt over, er det mulig å identifisere hva som er hva på en Joomla!-side, men for skjønne helt sammenhengen er man nødt til å se i koden, da de flyter litt inn i hverandre.

## JOOMLA! FRONTEND

Frontend er som nevnt tidligere den siden som er meningen skal ligge på web for. Der hvor informasjon og annen funksjonalitet skal vises til de riktige brukerne. Frontend består først og fremst av en mal, som kort fortalt er et slags kart med områder på siden hvor ting kan legges.



*Figur 4.2.9. Bildet skal illustrere hva som menes med kart over områder på siden. Dette viser hvor man kan legge forskjellige moduler, og hvor de da vil vises. Vil man for eksempel legge en sidemeny i området "Sidemeny venstre", kan denne igjen bestå av flere forskjellige menyer, hvor man sorterer de etter hvordan de skal presenteres. Valgte komponent med valgt view vises i "Hovedinnhold(komponent)". Dette er ment som en forklarende illustrasjon, og hver template har forskjellige "side-kart".*

Denne måten å presentere informasjon og andre ting gir hver enkelt store muligheter til å endre utseendet og personlig-gjøre sin webside.

## 4.3 KOMPONENTUTVIKLING

Joomla komponenter er delt i 2 deler. Du har front-end som brukes til å vise innholdet og vil være tilgjengelig for de besøkende på siden og back-end som brukes for å administrere innholdet og som kun være tilgjengelig for de som er innlogget i back-end.

### 4.3.1 KOMPONENTENS OPPBYGNING

På de følgende sidene skal vi gå igjennom komponentens oppbygning og hvordan de forskjellige delene fungerer sammen

#### ENTRY POINT

Hver gang komponenten kjøres vil denne koden alltid kjøres først. I Joomla kjernen finnes det en klasse for å håndtere kontrollere kalt JController. Denne klassen må utvides for å brukes i vår komponent.

Site/auctionsystem.php

```
<?php
//No direct access to file
defined('_JEXEC') or die('Restricted access');
//Joomla controller library
jimport('joomla.application.component.controller');

// Require the controller
require_once( JPATH_COMPONENT.DS.'controller.php' );
// Create the controller
$controller = new AuctionSystemController();
// Perform the Request task
$controller->execute( JRequest::getVar( 'task' ) );
// Redirect if set by the controller
$controller->redirect();
```

Koden over oppretter en kontrollere (hovedkontrolleren) og initierer et kontrollere objekt av klassen AuctionSystemController. Koden 'defined(' JEXEC') or die' vil sjekke om konstanten '\_JEXEC' er definert i applikasjonen. Hvis ikke vil den avslutte og det eneste du vil se er «Restricted access». Dette er lagt inn for å hindre direkte aksess til filen. Noe du gjør ved å legge inn følgende kode i nettleseren.

ROOT/components/com\_auctionsystem/auctionsystem.php

Dette er et sikkerhetstiltak i Joomla. Det er viktig at dette legges til i alle filer som kjører PHP kode. Dette vil hindre hackere i å kjøre tomme filer som vil avsløre viktige detaljer om systemet.

## HOVEDKONTROLLER

Hovedkontrollerens oppgave er å velge riktig Subkontroller og evt. kjøre en funksjon i denne. Hvis det ikke er medsendt noen oppgave eller «Task» vil den kjøre «display» funksjonen i den valgte kontrolleren som default.

site/controller.php

```
<?php
//No direct access to file
defined('_JEXEC') or die('Restricted access');
//Joomla controller library
jimport('joomla.application.component.controller');

class AuctionSystemController extends JController
{
    //Default task for JController
    function display() {
        //Gets the view parameter
        $view = JRequest::getVar('view');
        //Sets the controller name
        $controllername = 'AuctionSystemController'.$view;
        //Gets the controller controllers/viewname.php
        require_once(JPATH_COMPONENT.DS.'controllers'.DS.$view.'.php');
        // Create the Controller
        $controller = new $controllername;
        // Performs the requested task
        $controller->execute(JREQUEST::getCmd('task'));
        // Redirected if set by the controller
        $controller->redirect();
    }
}
```

View navnet blir hentet ut vha. JRequest som brukes av Joomla for å filtrere data.

Subkontrollerene ligger i mappen Site/controllers/ og har samme filnavn som view navnet.

For å velge hvilke View som skal vises må vi legge følgende kode til URL-en.

```
index.php?option=com_auctionsystem&view=viewname
```

Dersom du for eksempel ønsker å se en liste over alle auksjonene må vi velge Viewet «`auctionlist`».

```
index.php?option=com_auctionsystem&view=auctionlist
```

## SUBKONTROLLERER

I vår komponent har vi hele 7 subkontrollere, 1 til hvert view. Vi ønsket å dele opp koden slik at alle funksjoner som skulle håndtere brukerinteraksjon i de forskjellige viewene skulle ligge i den tilhørende subkontrolleren.

Site/controllers/auctionlist.php

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
jimport('joomla.application.component.controller');

// Auctionlist Controller

class AuctionsystemControllerAuctionlist extends JController{

    public function getModel($name = '', $prefix = '', $config = array(
('ignore_request' => true))
    {
        return parent::getModel($name, $prefix, array('ignore_request' =>
false));
    }

    //Default Task
    public function display(){
        //Sets template var to default
        JRequest::setVar( 'layout', 'default' );
        //Sets the view
        $view = $this->getView(JRequest::getWord('view', 'auctionlist'), 'html');
        //Adds the model
        $view->setModel($this->getModel( 'auctionlist', 'AuctionSystemModel' ));
        //Displays the view
        parent::display(true); // $view->display();
    } //function
}
```

AuctionSystemControllerAuctionList klassen er en utvidelse av Joomla sin JController klasse. Koden over sørger for at det tilhørende viewet legges til og at den tilhørende modellen med sine funksjoner vil være tilgjengelig i viewet.

Dersom det er ønskelig kan du legge til flere modeller i kontrolleren. Slik at det blir lettere å bruke funksjonalitet opp igjen.

For å legge til flere modeller benyttes følgende kode.

```
$auction = $this->getModel('auction');  
$view->setModel($auction);
```

For å benytte andre modellers funksjoner i et Viewet må du bruke følgende kode, forutsatt at modellen er lagt til i kontrolleren.

```
$data = $this->get('functionname', 'modelname')  
//For example  
$status = $this->get('AuctionStatus', 'auction')
```

Det er viktig å merke til at funksjonens hele navn i modellen vil være `getAuctionStatus()`

## MODELLENE

Modellen er delen av komponenten som innkapsler applikasjonens data. Den vil ofte inneholde rutiner for å håndtere og manipulere denne dataen, i tillegg til funksjoner for å hente ut data. I våres tilfelle vil den inneholde funksjoner for å legge til, fjerne, oppdatere og hente ut data.

Site/models/auctionlist.php

```

<?php
// No direct access to this file
defined('_JEXEC') or die('Restricted access');
// import Joomla modelitem library
jimport('joomla.application.component.modelitem');

// Auctionlist Model

class AuctionsystemModelAuctionlist extends JModelItem
{
    //Gets the auctionlist from the database
    public function getAuctions() {
        //Gets the database object
        $db = JFactory::getDBO();
        //Selects all rows and fields from the auction table
        $q = 'Select * from #__auctions';
        //Sets the query
        $db->setQuery($q);
        //Retrieves an associated list
        $auctions = $db->loadAssocList();
        //Returns the list
        return $auctions;
    }
}

```

AuctionSystemModelAuctionlist er en utvidelse av Joomla sin JModelItem klasse. I dette eksemplet inneholder den funksjon som henter ut en assosiativ array som inneholder alle auksjoner med tilhørende data som ligger lagret i databasen.

#### VIEWENE

Viewet er delen av komponenten som gjengir modellens data på en måte egnet for interaksjon. Viewet henter data fra modellen og legger denne dataen inn i et template som så blir presentert for brukeren.

Site/views/auctionlist/view.html.php

```

<?php

// No direct access to this file
defined('_JEXEC') or die('Restricted access');
// import Joomla view library
jimport('joomla.application.component.view');

/**
 * HTML View class for the Auctionsystem
 */
class AuctionsystemViewAuctionlist extends JView
{
    // Overwriting JView display method
    function display($tpl = null)
    {
        // Assign data to the view

        // Check for errors.
        if (count($errors = $this->get('Errors')))
        {
            JError::raiseError(500, implode('<br />', $errors));
            return false;
        }
        // Display the view
        parent::display($tpl);
        $this->setDocument();
    }

    function setDocument() {
        //Includes the necessary Files
        $document =& JFactory::getDocument();
        $document->addStyleSheet(JURI::base() .
'components/com_auctionsystem/assets/css/auctionsystem.css');
    }

    function generateAuctionList() {
        $auctions = $this->get('Auctions');
        if($auctions){
            $str = "<ul id="auctions">";
            foreach($auctions as $auction)
                $str .= "<li>{$auction['title']}</li>";
            $str .= "</ul>";
            return $str;
        }
    }
}

```

AuctionSystemViewAuctionlist klassen er en utvidelse av Joomla sin JView klasse. Vanligvis vil vi assignere data til \$this->item->data i display funksjonen. I dette tilfelle å lage en egen



funksjon for å generere en uordnet liste over auksjonene. Vi har også opprettet en «setDocument» funksjon for å inkludere de nødvendige filene.

```
Parent::display($tpl)
```

Vil vise en av template filene vi finner i Site/views/auctionlist/tmpl mappen. Dette vil være default.php dersom annet ikke er oppgitt. Dersom du ønsker å ha flere Templates tilknyttet et View må de navngis etter følgende mønster default\_\_templatename.php. Du kan veksle mellom disse ved å skrive om display funksjonen i kontrolleren og viewet til å ta imot enda en parameter «layout» og sette \$tpl variabelen lik denne. Dersom verdien av \$tpl variabelen er «map» vil den vise filen site/views/auctionlist/tmpl/default\_\_map.php.

## TEMPLATES

Templates er vanlige PHP filer som brukes til å vise data fra viewet på en særegen måte.

site/views/auctionlist/tmpl/default.php

```
<?php defined('_JEXEC') or die('Restricted access');?>
<html>
<body>
<h1>Auksjonsliste</h1>
<div id="auctionlist">
<?php echo $this->generateAuctionList(); ?>
</div>
</body>
</html>
```

Det er et simpelt eksempel på hvordan en template fil kan se ut. Den kaller funksjonen generateAuctionList() fra viewet som printer ut en uordnet liste over auksjonene som ligger lagret i databasen.

---

### 4.3.2 BRUKERINTERAKSJON I KOMPONENTER

Her skal vi ta for oss ett eksempel for hvordan du kan poste en form og behandle denne i kontrolleren. For å prøve å vise hvordan de forskjellige delene av en komponent fungerer sammen.

Den følgende koden vil ligge i template filen `site/views/auctionlist/tmpl/default.php`

```

jQuery('#textSearch').submit(function() {
    jQuery.ajax({
        type: 'POST',
        url: 'index.php?option=com_auctionsystem&view=auctionlist&task=textSearch&format=raw',
        data: jQuery(this).serializeArray(),
        success: function(tmp) {
            data = eval ('('+tmp+')');
            jQuery('#auctions li').remove(); //Removes all li elements
            for (i=0; i<data.length; i++) {
                //Appends the returned elements to the ul
                jQuery('#auctions').append('<li>'+data[i].title+'<li>');
            }
        }
    });
});

```

```

<form id="textSearch" onSubmit="return false">
<?php echo JHTML::_('form.token'); ?>
<label>Ord i tittelen</label>
<input type="text" name="search">
<input type="submit" value="Søk">
</form>

```

Over har vi en enkel form med tilhørende jQuery submit funksjon. Vi ønsker å søke etter annonser som har en tittel som inneholder en del av strengen brukeren taster inn. Når brukeren poster formen kjøres submit funksjonen og listen oppdateres.

```

<?php echo JHTML::_('form.token'); ?>

```

Legger til en Form Token i formen. Form tokens brukes for å forhindre CSRF (Cross Site Request Forgery) angrep. Form tokens legges til alle POST forms og GET spørringer som kan modifisere noe i Joomla systemet.

Format=RAW er en output type, denne legges til URL-en for å forhindre at vi får returnert html data fra viewet sammen med funksjonsdata.

Når brukeren trykker Søk kjøres JavaScript funksjonen som igjen kaller en funksjonen `textSearch()` i `auctionList` kontrolleren. Dette vil være uvant for mange siden de fleste er vant til å ha en egen PHP fil som brukes til å prosessere data.

```

public function textSearch() {
    //Checks if token is valid
    JRequest::checkToken() or die( 'Invalid Token' );
    //Gets the searchText parameter value through JRequest
    $searchText = JRequest::getVar('searchText');
    $db = JFactory::getDBO();
    $q = "SELECT title from #__auctions WHERE title LIKE '%
$searchText%'";
    $db->setQuery($q);
    $auctions = $db->loadAssocList();
    echo json_encode($auctions);
}

```

Funksjonen vil først sjekke at det er en gyldig token før den henter ut og returnerer en liste over alle titlene som inneholdt søkestrengen.

---

#### 4.3.3 JDATABASE

Joomla har et sofistikert databaselag som skal forenkle bruken for utviklere. Joomla kan bruke forskjellige SQL databasesystemer og kjører i flere forskjellige miljøer med forskjellige tabell prefikser. I tillegg til disse funksjonene, vil klassen automatisk opprette en database tilkobling.

#### FORBEREDE EN SPØRRING.

De følgende kodelinjene gjelder for alle spørringer.

```

// Get a database object
$db =& JFactory::getDBO();
$query = "SELECT * FROM #__example_table WHERE id = 999999;";
$db->setQuery($query);

```

Joomla vil automatisk oversette #\_\_ til prefiksen som er brukt på din Joomla installasjon.

#### SPØRRING RESULTATER:

Joomla sin databaseklasse kommer med mange forskjellige funksjoner for å hente ut resultatene fra en spørring. Under har vi listet opp de mest brukte.

```

$result = $db->loadAssoc(); //loads an associative array
$result = $db->loadResult(); //Loads a single value
$result = $db->loadRow(); //Loads the entire row as a numeric array
$result = $db->loadAssocList(); //Returns an indexed array of
associative arrays

```

## UTFØRE SPØRRINGER

For å utføre en spørring finnes det flere funksjoner.

```
//Returns an appropriate database resource if successful, or false  
$result = $db->query();
```

Brukes til å utføre instruksjoner for å oppdatere, sette inn eller slette data fra databasen. For å få mer utfyllende informasjon om en spørring kan du bruke følgende funksjoner.

```
$db->getAffectedRows();
```

Vil returnere informasjon om hvor mange rader i databasen som ble påvirket av spørringen.

```
$db->explain();
```

For mer utdypende informasjon om resultatet

```
$db->insertid();
```

Dersom du setter inn data i en tabell hvor det finnes en AUTO\_INCREMENT kolonne, vil insertid() returnere den siste genererte verdien.

## 4.4 KOMPONENTUTVIKLINGSPROSESSEN

Vi begynte først å lete etter ferdige komponenter vi kunne benytte for spare tid. Vi fant dessverre ingen auksjonskomponenter som hadde funksjonaliteten vi ønsket. Vi ble derfor nødt til å utvikle våres egen komponent.

Det første vi gjorde var å lese igjennom Joomla sin egen MVC komponent utviklings tutorial [16] og utviklet en enkel «hello world» komponent. Dette var en veldig enkel komponent med en kontroller, en modell ett view. Vi ble i begynnelsen veldig overveldet over hvor mange filer som skulle til for å lage noe så simpelt.

Vi hadde aldri kodet etter MVC designmønsteret før så hele konsepter var helt nytt for oss. Det i tillegg til at det var mye ny kode å sette seg inn i gjorde at det gikk litt tregt i begynnelsen.

Vi fant fort ut at komponenten vi skulle utvikle ville bli langt mer komplisert enn komponenten i tutorialen. Våres komponent kom til å bestå av en hovedkontroller, flere subkontrollere, flere views som hver kunne ha flere layouts. Vi ville derfor trenge mer lesestoff for å finne ut hvordan komponenten kunne bygges ut. Vi var først innom biblioteket og lånte 2 bøker om Joomla, men ingen av disse inneholdt noe som helst om komponentutvikling. Vi stakk innom biblioteket igjen men de hadde ingen bøker om komponentutvikling til Joomla 1.7 eller 2.5 i sine systemer. Noe som førte til at vi prøvde å lete etter mer informasjon på internett, men fant ingenting som var tilgjengelig for offentligheten. Vi fant flere sider som tilbød treningsvideoer og utfyllende dokumentasjon, men dette kostet penger.

Etter et møte med veileder ga han oss et tips om at vi burde ta kontakt med Monica Strand som underviser i et fag hvor de bruker Joomla som en del av undervisningen. Hun ba oss så ta kontakt med en student som skulle ha mer kunnskap til komponentutvikling enn henne selv.

Før vi rakk å få til et møte med studenten hadde vi allerede klart å løse de problemene vi hadde med komponenten ved å studere koden i ferdig utviklede komponenter. Det ble derfor ikke nødvendig å avholde dette møtet.

Skallet til komponenten var nå på plass. Og vi kunne nå begynne med utviklingen av de forskjellige komponentsidene.

---

#### 4.4.1 AUKSJONSLISTE-VISNING

Auksjonsliste siden er siden hvor en bruker kan se en liste over alle publiserte auksjoner. Som standard vises kun de auksjoner som ikke er avsluttet, men brukeren kan selv velge hvilke auksjoner som vises ved hjelp av søkeseksjonen. Søk kan her gjøres på to måter: søk

etter ord i annonsen samt søk etter gitte attributter. Auksjonene kan også sorteres etter gitte parametre (prisantydning, størrelse mm). Listen genereres dynamisk vha. JavaScript.

The screenshot displays a web interface for viewing auction listings. On the left is a sidebar with search filters, and on the right is a main area with a list of three properties. At the top right, there is a sorting dropdown set to 'Kvm Pris Lav-Høy' and a 'Vis annonser i kart' button. Below the sorting are navigation buttons for 'Første', 'Førrige', '1', '2', '3', 'Neste', and 'Siste'.

**Search Filters (Left Sidebar):**

- Antall Treff: 21
- Ord i annonsen: [input field]
- Søk: [button]
- Radio buttons:  Vis aktive,  Vis alle ikke utgått,  Vis alle
- Fylke: Velg Fylke [dropdown]
- Kommune: [input field]
- Pris: [input 0] - [input 10000000]
- Størrelse (m²): [input 0] - [input 500]
- Antall Soverom: [input 0] - [input 6]
- Eierform: Velg Eierform [dropdown]
- Boligtype: [input field]

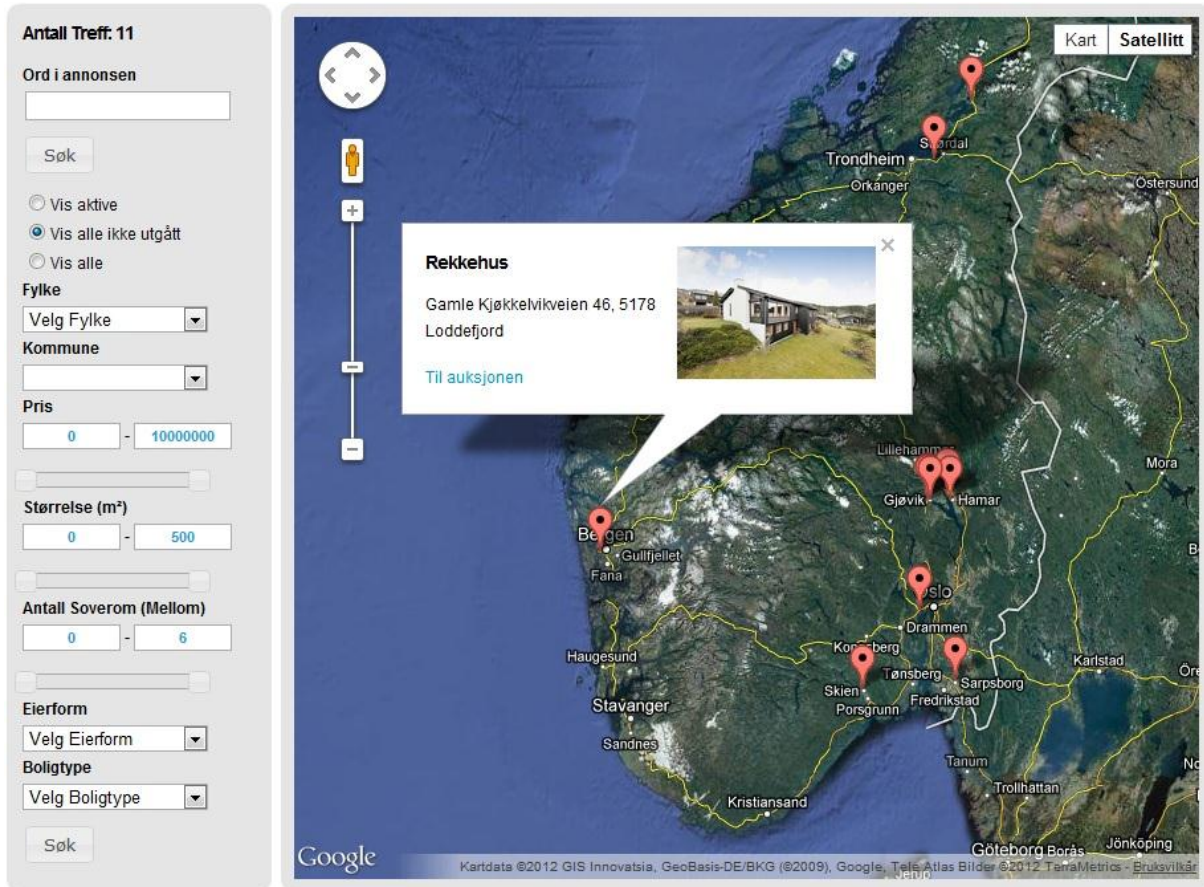
**Property Listings (Right Main Area):**

- Romslig enebolig**  
 Eierform: Eier  
 187 / 163 m²  
 Furnesvegen 103  
 2320, Furnes  
 Gjenstående tid: 6d 5t 55m  
 Prisantydning: 1.500.000,-
- Pen 2-Roms**  
 Eierform: Andel  
 55 / 53 m²  
 Gamlegrensa 8 B  
 3747, Skien  
 Gjenstående tid: 8d 5t 55m  
 Prisantydning: 500.000,-  
 Fellesgjeld: 1.100.000,-  
 Totalt: 1.600.000,-  
 Fellesutgifter: 1.300,-
- Stor enebolig med utleiedel**  
 Eierform: Eier  
 265 / 264 m²  
 Hardhausveien 60  
 8520, Ankenesstrand  
 Gjenstående tid: Avsluttet  
 Prisantydning: 2.500.000,-

Figur 4.4.1 Auksjonslistevisningens utforming.

#### 4.4.2 KARTVISNING AV AUKSJONER

Denne siden inneholder i stor grad samme funksjonalitet som auksjonslisten. I stedet for en liste over auksjoner genereres et kart hvor eiendommenes adresse vises som markører. Man kan i denne visningen angi søkeparametre på samme måte som i auksjonslistevisningen. Til kartmarkørene er det knyttet en event som genererer infovinduer med auksjonsinformasjon når de trykkes. Søkene og markørplasseringene skjer dynamisk vha javascript.



Figur 4.4.2 Kartvisningens utforming.



#### 4.4.3 UTVIKLING AV AUKSJONSSIDEN

Auksjonssiden brukes til å presentere bilder og informasjon om salgsobjektet.

The screenshot displays a web interface for an auction. At the top is a navigation menu with links: Forside, Eiendommer, Produkter, Informasjon, Referanser, Boligblogg, Min Side, and Admin. Below the menu is a header area with a button 'Følg denne auksjonen!' and 'Anbefal', and a timestamp 'Endret: 2012-04-23 12:08:43' with links 'Rediger Annonse' and 'Auksjonsid: 15'. The main content area features a large image of a modern living room with a sofa, dining table, and kitchen island. Below the main image are three smaller thumbnail images. To the right of the main image is a sidebar with several panels: 'Budgivning' with a timer '1 Dag, 8 Timer, 3 Min, 13 Sek' and a bid input field; 'Budhistorikk' with a table header 'Bud' and 'Tidspunkt'; 'Auksjonsinfo' with dates and status; 'Prisinfo' with 'Prisantydning: 3.500.000'; and 'Boliginformasjon' with details like 'Soverom: 4', 'Bruksareal: 208 (m²)', and 'Bygd: 2012'. The main text area contains placeholder text 'saffaffsa Lorem ipsum saffaffsa Lorem ipsum saffaffsa...'.

Figur 4.4.3 Auksjonssidens utforming

Bud historikken, bud nivået og auksjonsstatusen oppdateres automatisk hvert 10 sekund ved hjelp av Ajax slik at brukerne slipper å sitte og oppdatere siden for å følge budrunden

Det er mulig å følge auksjonen på email ved å legge til et abonnement ved å trykke på «Følg denne auksjonen» knappen øverst til venstre.



Vi ønsket å gjøre det så enkelt som mulig å dele boligen på facebook ved å implementere en «like button» øverst på siden. Siden templatens lastes inn i dokumentets «content box» måtte vi få bruke følgende metode for å få lagt til metadata i html dokumentets head

```
$doc = JFactory::getDocument();
$doc->addCustomTag('<meta property="og:site_name" content="Bolighandell" />');
```

Vi har benyttet oss av «Galleria» som bildegalleri. Galleria var et av de få bildegalleriene som støttet bilder av forskjellige størrelse samtidig som den så bra ut.

Vi har benyttet oss av Keith Wood sin jQuery countdown timer for å vise gjenstående tid. Noe vi oppdaget under implementering av timeren var at vi ikke hadde tatt høyde for at serveren kunne ha en annen tidssone enn den vi har i Norge. For å løse dette problemet prøvde vi først å bruke JFactory::getDate(), men denne returnerte også serverens lokale tid og ikke tiden vi hadde valgt i Joomla sin back-end. Vi endte opp med å måtte hente ut den valgte tidssonens offset fra Joomla sin «Config» for å få satt riktig tidspunkt.

```
$offset = JFactory::getConfig()->getValue('config.offset');
date_default_timezone_set($offset);
```

Andre steder i komponenten hvor vi har blitt nødt til å regne ut gjenstående tid har vi gjort følgende.

```
//Gets offset from the Joomla Config
$offset = JFactory::getConfig()->getValue('config.offset');
//Creates a dateTime object with current date and time
$date = new DateTime("now", new DateTimeZone($offset));
//Creates a dateTime object with the auctions end date and time
$endtime = new DateTime($auctions[$i]['endtime'], new DateTimeZone($offset));
//Calculates the difference between the objects
$interval = $endtime->diff($date);
//Formats the output
$auctions[$i]['remaining'] = $interval->format("%dd %ht %im");
```

Det er viktig at begge DateTime objektene har samme tidssone siden diff() funksjon tar med tidssone forskjeller i sine beregninger.

Selve budgivningsfunksjonaliteten er kanskje den viktigste og mest kritiske delen av hele systemet. Oppdragsgiver ønsket et budgivningssystem hvor brukerne må godkjennes av selger før de får lov til å legge inn bud på en eiendom. For å bli godkjent som budgiver må du sende inn en enkel godkjenningssøknad, slik at selgeren får muligheten til å sjekke at du har

finansieringen i orden. Det er så opp til selger om han ønsker å godkjenne søkeren som budgiver eller ikke.

Selve budgivningssystemet er «Proxy bidding» basert [17]. Dette er implementasjon av «English second-price auction» og er brukt på kjente auksjonsnettsteder som for eksempel eBay. Måten dette fungerer på er at personen som ønsker å by legger inn den maksimale verdien han er villig til å gi for produktet. Når andre bud legges inn vil budsystemet vil systemet automatisk regne ut hva det neste budet skal settes til. Dette vil normalt være det nest høyeste budet pluss et pre-spesifisert inkrement. På denne måten slipper budgiverne å følge auksjonen i sanntid, det blir også enklere å by på flere auksjoner samtidig.

---

#### 4.4.4 NYTT BUD FUNKSJONEN

Funksjonen vil først hente inn nødvendig informasjon for å bestemme om budet skal godkjennes. Dette innebærer å sjekke at brukeren har blitt godkjent av selger, at budet er høyt nok og at auksjonen fortsatt er aktiv når budet legges inn.

Hvis dette er i orden vil den sjekke om brukeren tidligere har registrert ett maksbud på denne auksjonen. Hvis dette er tilfelle vil maksbudet oppdateres dersom det nye maksbudet er større enn det gamle. Hvis det ikke er registrert ett maksbud vil dette legges inn.

Algoritmen vil så hente ut de to største maksbudene for den gitte auksjonen for å finne frem til det neste budet som skal legges inn. Hvis det kun ligger ett maksbud inne vil den sjekke differansen fra dette budet til den skjulte minsteprisen selgeren har lagt inn. Dersom budet er større enn minsteprisen vil budet settes lik minsteprisen, hvis ikke vil det bli satt så nærme opp mot minsteprisen som mulig.

```

if($i == 1){ //If only 1 Maxbid
    $bidderid = $maxbids[0]['uid'];
    $diff = $maxbids[0]['amount']-$bidinfo['minprice'];
    if($diff >=0) //If MAXBID > asking price
        $nextbid = $bidinfo['minprice']; //current bid = minprice

    else //if MAXBUD < asking price
        $nextbid = $maxbids[0]['amount']; //next bid = users maxbid
    if($nextbid != $bidinfo['currentbid']){ //If no change
        $q = "INSERT INTO #__bids ".
            "`added`, `uid`, `aid`, `amount` ".
            "VALUES('".$currdate."', $bidderid, $id, $nextbid)";
        $db->setQuery($q);
        $bidres = $db->query();
    }
}
}

```

Hvis det ligger flere enn ett maks bud inne vil den finne differansen mellom de to høyeste maksbudene.

- Dersom differansen er større enn det pre-definerte inkrementet vil det neste budet settes til det laveste budet pluss ett inkrement. Hvis budet er lavere enn den gjemte minsteprisen vil det blir justert så nærme opp mot denne prisen som mulig.
- Hvis differansen er mindre enn ett inkrement vil det neste budet settes lik det største av de to.
- Hvis det ligger to identiske maks bud inne vil det vil det eldste av de 2 bli satt som neste bud. Hvis begge budene ble lagt inn akkurat samtidig (samme sekund). Vil den generere ett tilfeldig tall for å velge.

Den vil så sjekke det nye budet opp mot det gamle budet for å se om det er noen endring.

Dersom det ikke er noen endring, eksempelvis når en bruker ønsker å oppdatere sitt maksbud når han allerede leder en auksjonen. Dette gjøres for å forhindre dobbellagring av bud.

```

else{ //If 2 maxbids
    $diff = $maxbids[0]['amount']-$maxbids[1]['amount']; //Diff between bids
    if($diff>=$increment){ //If Diff > increment
        if($maxbids[0]['amount']<=$bidinfo['minprice']) //if maxbid <= minprice
            $nextbid = $maxbids[0]['amount']; //next bid as close to
        else //minprice as possible
            $nextbid = $maxbids[1]['amount']+$increment; //Next bid = min bid+inc
        $bidderid = $maxbids[0]['uid'];
    }
    else if($diff==0){ //If 2 equal maxbids
        $nextbid = $maxbids[0]['amount'];
        if($maxbids[0]['added']<$maxbids[1]['added']) //oldest -> leader
            $bidderid = $maxbids[0]['uid'];
        else if($maxbids[0]['added']==$maxbids[1]['added']){ //if placed at
            $rand = rand(0,1); //the same time, pick random
            $bidderid = $maxbids[$rand]['uid'];
        }
        else
            $bidderid = $maxbids[1]['uid'];
    }
    else if($diff<$increment){ //If DIFF < Increment
        $bidderid = $maxbids[0]['uid']; //Highest bid -> leader
        $nextbid = $maxbids[0]['amount'];
    }
    if($nextbid != $bidinfo['currentbid']){ //If resulting in change
        $q = "INSERT INTO #__bids ". //Update bid
            "(`added`, `uid`, `aid`, `amount`) ".
            "VALUES('".$currdate."', $bidderid, $id, $nextbid)";
        $db->setQuery($q);
        $bidres = $db->query();
    }
}
}

```

Etter at budet er plassert vil funksjonen generere og returnere en beskjed til budgiveren.

Dersom budet fører til at en ny bruker overtar ledelsen av auksjonen vil den gamle lederen motta en epost hvor det står at han ikke lenger leder auksjonen. Det vil også sendes ut en epost til alle som følger budrunden.

#### 4.4.5 UTVIKLING AV MIN SIDE

Er den påloggede brukers private side. Her finner brukerne oversikten over sine salgsobjekter, bud historikk, abonnenter og kjøp.

### Min Side

The screenshot displays the 'Min Side' (My Page) interface. On the left is a sidebar with options: 'Ny annonse +', 'Bestill Produkter', and 'Notifikasjoner (1)'. The main content area has tabs for 'Mine Auksjoner (16)', 'Min Budhistorikk', 'Mine abonnenter', and 'Mine Kjøp'. Below the tabs are four auction listings, each with a photo, title, action buttons ('Rediger', 'Informasjon'), and key dates/statuses.

| Auktionstitel                                  | Start            | Slutt            | Publisert | Status       |
|--|------------------|------------------|-----------|--------------|
| Rekkehus                                       | 14-05-2012 18:00 | 07-08-2012 18:00 | Ja        | Aktiv        |
| Romslig enebolig                               | 30-04-2012 11:00 | 21-06-2012 18:00 | Ja        | Aktiv        |
| Stor villa i holmenkollen, utsikt over fjorden | 16-05-2012 18:00 | 11-06-2012 18:00 | Nei       | Ikke startet |
| Stor Innholdsrik Villa                         | 27-04-2012 18:00 | 30-05-2012 18:00 | Ja        | Aktiv        |

Figur 4.4.4 Min Side

Brukeren vil også få opp et notifikasjonsikon (som vist på bildet over) dersom det ligger budgiver søknader til godkjenning.

#### 4.4.6 UTVIKLING AV AUKSJONSINFORMASJONSSIDE

Auksjonsinformasjonssiden er siden selgeren kan gå inn på for å se informasjon som kun er tilgjengelig for selgeren. Siden inneholder informasjon som antall annonsevisninger, antall abonnemeter, godkjente budgivere og lignende.

Figur 4.4.5 Auksjonsinformasjonssiden

Sidene inneholder ett eget bildegalleri hvor brukeren enkelt kan laste opp og slette bilder på en enkel måte. På siden finner du også funksjonalitet som gjør det enkelt å legge til nye visninger, opplasting av dokumenter og diverse. Brukeren vil også ha mulighet til å slette auksjonen frem til auksjonsstart med alle tilhørende data.

For å laste opp bilder har vi benyttet oss av Uploadify. Uploadify i seg selv vil kun laste opp filene til mappen angitt i innstillingene. Vi måtte i tillegg få lagret opplysninger om filene i databasen. Vi brukte opprinnelig Uploadify v. 3.1.2 men ble nødt til å gå tilbake til v. 2.1.4 siden vi hadde en del problemer med den nyeste versjonen og IE. Med tanke på målgruppen til prosjektet var det ikke aktuelt å ha en filopplaster som ikke fungerer i IE.

```

jQuery('#file_upload').uploadify({ //Init uploadify
  'uploader'   : 'components/com_auctionsystem/assets/uploadify/uploadify.swf',
  'script'    : 'components/com_auctionsystem/assets/uploadify/uploadify.php',
  'cancelImg' : 'components/com_auctionsystem/assets/uploadify/cancel.png',
  'folder'    : 'images/uploads',
  'auto'      : true, //use for auto upload
  'removeCompleted' : true,
  'fileExt'   : '*.jpg',
  'fileDesc'  : 'Image Files',
  'multi'     : true,
  'buttonText': 'Velg Filer',
  'onComplete': function(event, ID, fileObj, response, data) { //insert into database
    var url = '<?php echo JRoute::_('index.php?option=com_auctionsystem&view=auktioninfo&task=uploadImage&format=raw&id=' .JRequest::getInt('id')).'&'. JUtility::getToken() .'=1' ); ?>';
    jQuery.ajax({
      type: 'POST',
      url: url,
      data: {'path': fileObj.filePath, 'size': fileObj.size },
      success: function(tmp){
        data = eval ('('+tmp+')');
        jQuery('#userGallery').hide().append(appendGalleryPicture(data)).fadeIn('slow');
      }
    });
  }
});

```

Uploadify initiering.

---

#### 4.4.7 ADMINISTRATOR SIDE

Administratorsiden var noe vi fra starten av hadde tenkt å ha i back-end delen av komponenten. Noe den også var til vi innså at dette skapte visse problemer når det kom til godkjenning av auksjoner og bruk av Chronofoms. Siden Joomla sin front-end og back-end er separate vil du ikke være pålogget i front-end dersom du er pålogget i back-end. Ergo vil du ikke få tilgang til de nødvendige formene vi har opprettet i Chronofoms fra back-end delen av komponenten. Grunnen til dette er at vi ikke vil ha tilgang til brukerinformasjon om brukere pålogget i back-end i front-end.

Dette førte til at vi til slutt bestemte oss for å legge administratorsiden i front-end.

På administratorsiden vil du få opp en oversikt over alle auksjonene på siden. Du vil også få opp en liste over auksjonsforslag som avventer godkjenning.

## Administrator

The screenshot displays the 'Administrator' interface. At the top, there are navigation buttons: 'Opprett Ny Auksjon +', 'Abbonement', 'Forside', and 'Produkter'. The main content area is titled 'Auksjonoversikt' and contains a table of auctions. Each row in the table lists an address and provides 'Rediger' and 'Info' links. To the right, a sidebar titled 'Auksjonsforslag til godkjenning' shows a list of proposals, with one entry: 'Oberst Angells vei 6, 0787 Oslo'.

| Address                                   | Rediger | Info |
|---|---------|------|
| Kongsvegen 22, 2380 Brumunddal            | Rediger | Info |
| strandgata 13, 2315 Hamar                 | Rediger | Info |
| vidhaugen 139, 7550 Hommelvik             | Rediger | Info |
| Hardhausveien 60, 8520 Ankenesstrand      | Rediger | Info |
| Gamlegrensa 8 B , 3747 Skien              | Rediger | Info |
| Kringsjø 3, 2819 Gjøvik                   | Rediger | Info |
| Enganvegen 48, 7670 Inderøy               | Rediger | Info |
| Frognerveien 29, 0264 Oslo                | Rediger | Info |
| Jongsåsveien 2 , 1338 Sandvika            | Rediger | Info |
| Solliveien 9, 4022 Stavanger              | Rediger | Info |
| Gamle Kjøkkelvikveien 46, 5178 Loddefjord | Rediger | Info |
| St. Hansveien 13 , 1727 Sarpsborg         | Rediger | Info |
| Nordlitoppen 43, 1743 Klavestadhaugen     | Rediger | Info |
| kirkebakken 4, 1767 Halden                | Rediger | Info |
| kråkjordet terrasse 12, 2819 Gjøvik       | Rediger | Info |
| strandgata 4, 2815 Gjøvik                 | Rediger | Info |
| Vestbyvegen 10, 2830 Raufoss              | Rediger | Info |
| Furnesvegen 103, 2320 Furnes              | Rediger | Info |
| Oppenåsen 125, 3518 Hønefoss              | Rediger | Info |
| jessnessvegen 170, 2320 Furnes            | Rediger | Info |
| Kongsvegen 99, 2380 Brumunddal            | Rediger | Info |

Figur 4.4.6 Frontend administrator

Du vil også finne følgende undersider på administratorsiden:

- Abonnemeter: Brukes til å administrere abonnemeter på siden
- Forside: Velge hvilke boliger og produkter som skal vises på forsiden.
- Produkter: Legge til og fjerne produkter

### 4.5 PLUGINS/APIER BENYTTET TIL UTVIKLING AV KOMPONENTEN:

Vi har benyttet oss av flere plugins i komponenten våres. Alle er lisensiert under MIT lisensen som er en gratis lisens. Lisensene er lagt ved i «Assets» mappen i komponenten.

JQUERY [18] & JQUERY UI [19]



jQuery er et JavaScript bibliotek som forenkler HTML traversering, event håndtering, animasjoner og Ajax interaksjoner for kjappere webutvikling.

Siden Joomla bruker MooTools som standard JavaScript rammeverk har dette ført til et par komplikasjoner med bruk av jQuery. Dette har stort sett har latt seg løse ved å bruke jQuery.noConflict(). noConflict brukes for å forhindre problemer som kan oppstå ved bruk av flere JavaScript biblioteker. Mange benytter seg av \$ som funksjon eller variabelnavn. I jQuery sitt tilfelle er \$ kun et alias for jQuery. Ved å bruke noConflict() kan enkelt gi tilbake kontrollen over \$ tilbake til de andre bibliotekene. Noe som fører til at vi må bytte ut \$ variabelen man vanligvis bruker med jQuery, som vist i følgende kode.

```
$('#addNew').button({ icons: { secondary: "ui-icon-plus" }});
//Is replaced with
jQuery('#addNew').button({ icons: { secondary: "ui-icon-plus" }});
```

#### UPLOADIFY [20]

Er en jQuery plugin som gjør det enkelt å legge til funksjonalitet som gjør opplasting av flere filer samtidig mulig. Kommer i to versjoner (HTML5 og Flash).

Vi har benyttet oss av flash versjonene siden HTML5 versjonen ikke er gratis. Flash versjonen støttes ikke av mobile enheter.

#### GALLERIA [21]

Er et JavaScript bildegalleri. Skal forenkle prosessen med å lage et profesjonelt bildegalleri for web og mobile enheter.

#### JQUERY COUNTDOWN 1.5.11 [22]

Er en jQuery plugin for å enkelt kunne legge til en countdown timer som teller ned til et bestemt tidspunkt.

#### JQUERY DATETIMEPICKER [23]

Er en jQuery UI addon som legger til slidere for valg av klokkeslett til jQuery UI sin datetimerpicker

#### GOOGLE MAPS JAVASCRIPT API [24]

API som skal gjøre det enkelt for utviklere å integrere Google Maps på sine nettsider. Tjenesten er gratis å benytte for alle sider som er gratis for alle sine brukere. Forutsetter kun at du har din egen API key.

## 4.6 JOOMLA EXTENSIONS BRUKT I UTVIKLING AV SIDEN

### 4.6.1 CHRONOFORMS [25]

Vi begynte først med å lage forms på den tradisjonelle måten, men innså fort at dette ville bli veldig tidkrevende siden vi var avhengig av å ha mange forms med mange felter. Vi begynte derfor å lete etter et verktøy som ville gjøre dette enklere for oss.

Vi var i begynnelsen bekymret over at det å bruke et verktøy for å lage forms ville føre til at vi ville ha veldig begrensede muligheter når det kom til design og annen funksjonalitet. Etter å ha sett igjennom de gratis alternativene på Joomla sin Extension side ble vi positivt overrasket over hvor mye funksjonaliteten de forskjellige alternativene tilbød.

Etter å ha sett igjennom de tilgjengelige gratis alternativene falt valget på Chronoforms, som virket som det desidert beste gratis alternativet. Chronoforms hadde all funksjonaliteten vi trengte virket bra dokumentert. På hjemmesiden hadde de også et stort forum med over 70 000 poster, hvor utviklerne tar seg tid til å svare på de fleste spørsmålene. Det viste seg i ettertid at dokumentasjonen ikke var så bra som først antatt. Den mest grunnleggende funksjonaliteten fantes det mye dokumentasjon på, men utover dette tok utviklerne seg betalt for ytterligere dokumentasjon.

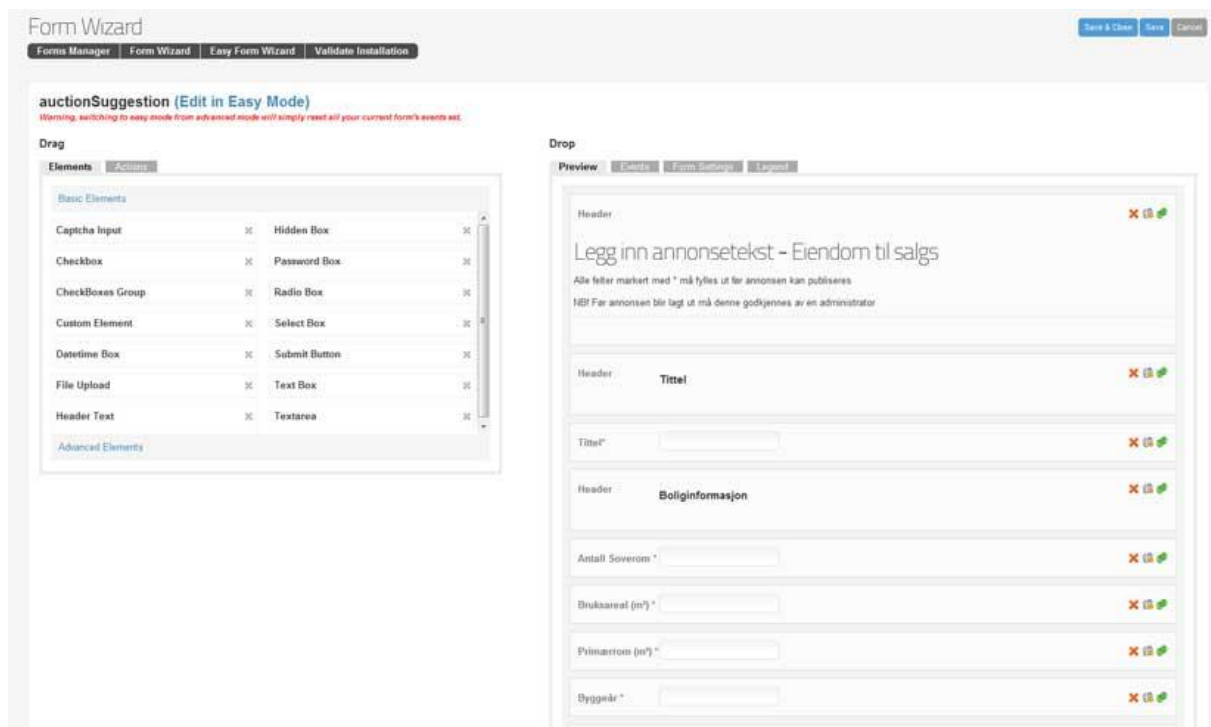
#### HVORDAN VI HAR BRUKT CHRONOFORMS

Her skal vi prøve å gi en kjapp gjennomgang i hvordan man bruker Chronoforms til å lage former. Dette vil forhåpentligvis klargjøre hvorfor vi valgte å benytte ett verktøy for å designe forms.

I dette eksemplet skal vi ta for oss formen som brukes til registrering av annonseforslag.

Det første du må gjøre er å åpne Chronoform sin «Forms manager» fra Extensions menyen i back-end.

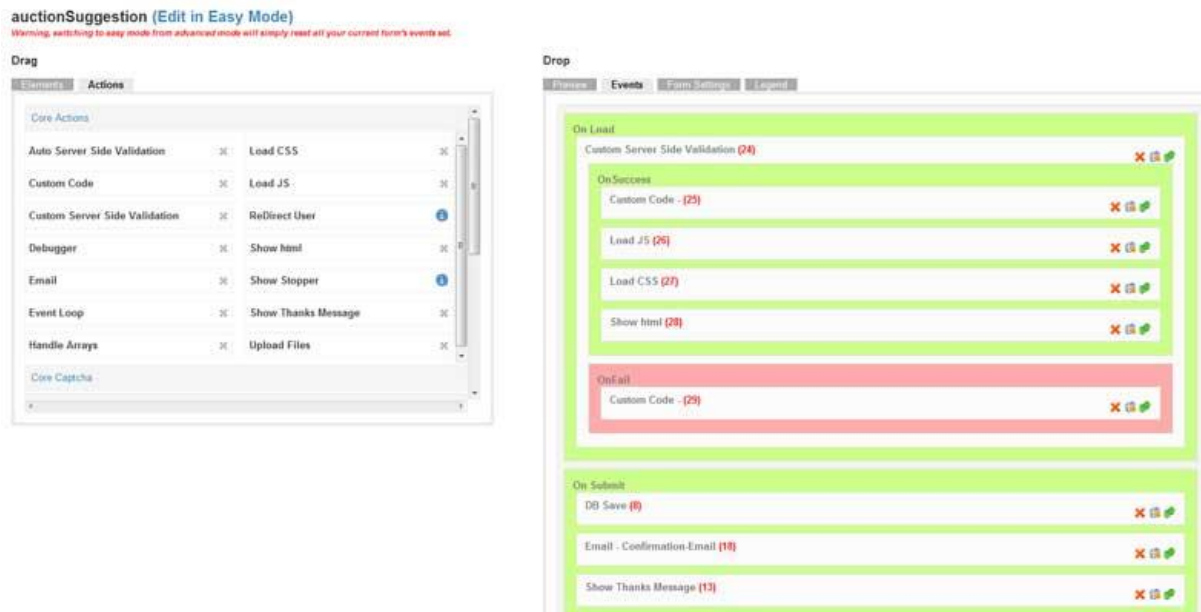
Du oppretter så en ny form. Etter at du har gitt formen ett navn velger du «form wizard» som «Formtype». Dette gjør at vi kan bruke Chronoform sin «Form wizard» til å designe formen med et enkelt dra og slipp grensesnitt.



Figur 4.6.1 Form wizard, Preview

For å legge til felter drar du et element fra menyen til venstre over i «Preview» vinduet. Når du har plassert elementet kan du redigere det ved å trykke på «Settings» knappen til elementet. Det er viktigste at du gir feltet et feltnavn og en etikett tekst. Du kan også legge til instruksjer eller verktøytips dersom det er nødvendig. Dersom feltet skal være obligatorisk må du krysse av for «Required» under validering innstillingene. Der har du også en rekke andre valg for validering av selve innholdet i feltet.

Etter at du har lagt til alle feltene du ønske å ha med bytter du fra «Preview» vinduet til «Events» vinduet. Det viktigste å få med her er «Show html», uten «Show html» vil ikke HTML dataene til formen vises.



Figur 4.6.2 Form wizard, events

I våres tilfelle vil «Events» vinduet være litt mer komplisert. Øverst har vi «On Load» funksjonaliteten. Dette er koden som skal kjøres når formen lastes inn. Vi starter med å legge inn et «Custom server side» element som inneholder en liten kodesnutt som sjekker om brukeren er pålogget eller ikke siden vi ønsker at formen kun skal være tilgjengelig for registrerte brukere.

```
<?php
$user = JFactory::getUser();
if($user->guest)
    return false;
?>
```

Dersom brukeren er en gjest vil brukeren kun få opp «Custom code» elementet som ligger i «OnFail» som vil vise teksten «Ingen tilgang».

Dersom brukeren ikke er pålogget vil den gå inn i «OnSuccess»s hvor vi har 4 elementer.

- Øverst har vi et «Custom code» element. Dette inkluderer de nødvendige filene som google maps sitt API og datetimepicker sin JavaScript fil.
- Load JS elementet inneholder JavaScript koden til som vi ønsker å bruke i formen, i denne formen koden for Google Maps og DateTimePicker funksjonaliteten.
- Load CSS filen inneholder formens CSS kode.
- Til slutt har show html elementet som sørger for at formen vises

Under har vi «On Submit» funksjonaliteten. Dette er koden som skal kjøres etter at brukeren har trykket på send knappen. Før du får lagret data i databasen er du nødt til å ha opprettet en database tabell for formen din. For å slippe å gjøre dette manuelt i databasen kan du bruke «Create table» funksjonen du finner i Chronoforms. Dette gjør du i «Forms manageren» ved å sjekke av formen og trykke på «Create table»

| #  | Field name   | Enabled                             | Type     | Length | Primary key                         | Default | Extra          |
|----|--------------|-------------------------------------|----------|--------|-------------------------------------|---------|----------------|
| 1  | ct_id        | <input checked="" type="checkbox"/> | INT      | 11     | <input checked="" type="checkbox"/> |         | Auto_Increment |
| 2  | ct_ud        | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 3  | ct_created   | <input checked="" type="checkbox"/> | DATETIME |        | <input type="checkbox"/>            |         |                |
| 4  | ct_modified  | <input checked="" type="checkbox"/> | DATETIME |        | <input type="checkbox"/>            |         |                |
| 5  | ct_ipaddress | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 6  | ct_user_id   | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 7  | title        | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 8  | bedrooms     | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 9  | useablearea  | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 10 | primaryroom  | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 11 | bulth        | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 12 | renovatedin  | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 13 | energy       | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 14 | solgtipe     | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |
| 15 | aeform       | <input checked="" type="checkbox"/> | VARCHAR  | 255    | <input type="checkbox"/>            |         |                |

Figur 4.6.3 Chronoforms, create table

«Create table» funksjonen vil automatisk opprette kolonner til alle feltene i formen med det samme navn du har gitt feltene, i tillegg vil den også opprette et par standard felter. Du er selv nødt til å definere datatypene.

Når du har opprettet tabellen kan vi legge til «DB save» i On Submit vinduet. De eneste du er nødt til å gjøre er å velge riktig tabell å lagre data til.

Vi ønsker også å sende ut en bekreftelses email til brukeren om at hans annonseforslag er registrert. For å sende en mail til den påloggede brukerens email adresse må vi legge til et ekstra felt i formen. Vi legger til et «Custom Element» med følgende kode.

```
<?php
$user = JFactory::getUser();
echo "<input type='hidden' name='useremail' value='". $user->
email."/>";
?>
```

Dette feltet vil være gjemt for brukeren og vil inneholde den påloggede brukerens epostadresse. Denne skal vi senere bruke i en «Email» event. For å bruke denne verdien i email eventen må du legge til feltnavnet (useremail) i «Dynamic To» feltet.

For å opprette en beskjed som skal vises til brukeren etter at formen har blitt sendt må vi legge til en «Show thanks message» event nederst i «on submit» vinduet.

---

#### 4.6.2 OVERSIKT OVER FORMER VI HAR OPPRETTET I CHRONOFORMS

Under følger en oversikt over formene vi har benyttet chronoforms til å opprette. Vi har i tillegg prøvd å beskrive hvordan de er bygget opp i korte trekk.

##### AUCTIONSUGGESTION, NEWAUCTION

Formene er såpass like at vi valgte å ta begge under samme punkt. De brukes begge til å legge inn annonser. Den eneste forskjellen er at newAuction legger auksjonen rett ut på siden mens en auctionSuggestion annonsene må godkjennes av en administrator før den publiseres-

For å få tilgang til auctionSuggestion formen må du være pålogget siden. For tilgang til newAuction må du være en administrator på siden. Begge sjekkes i en «Custom server side validation» event når formen lastes inn.

Den påloggede brukeren kan så fylle inn annonse feltene. Det eneste som ikke er standard Chronoforms i formene er bruken av Google Maps for å finne boligens lengde og breddegrad og DateTimePicker for å velge start og sluttidspunktet for auksjonen. Plassering av bolig på kartet har blitt implementert ved å legge til et «custom element» som inneholder et kart. Du kan plassere markøren enten ved å dra den på plass manuelt eller ved å taste inn adressen og postnummeret til eiendommen.

Vi har lagt til en change() funksjon i JavaScript filen som kjøres hver gang du velger deg bort fra adressefeltet.

```

jQuery('#address').change(function(){
  var address = jQuery("#address").val();
  var postnr = jQuery("#postnr").val();
  var address = address+', '+postnr;
  geocoder.geocode( { 'address': address}, function(results, status) {
    if (status == google.maps.GeocoderStatus.OK) {
      map.setCenter(results[0].geometry.location); //Centers the map at the location
      map.setZoom(16); //Zooms in, updates field values
      jQuery('#lat').val(results[0].geometry.location.lat().toString().substr(0, 12));
      jQuery('#lng').val(results[0].geometry.location.lng().toString().substr(0, 12));
      marker.setPosition(results[0].geometry.location); //Place the marker
    } else {
      alert("Geocode was not successful for the following reason: " + status);
    }
  });
});

```

Vi har benyttet oss av Google sin Geocoder for å finne lengde og breddegraden til adressen som har blitt tastet inn. For økt presisjon har vi lagt til postnummeret i søkestrengen. Funksjonen legger Geocoder resultatene inn i lat og lng feltene som ligger gjemt for brukeren. Markøren vil så bli flyttet til stedet Geocoderen returnerte. Dersom resultatet ikke var helt riktig er det mulig å dra markøren på plass slik at resultatet blir mer nøyaktig.

For å få oppdatert feltverdiene hver gang markøren flyttes har vi lagt til en «event listener» som vil oppdatere feltverdiene.

```

marker = new google.maps.Marker({ //Plasserer en markør midt på kartet
  draggable: true,
  map: map,
  position: latLng
});

google.maps.event.addListener(marker, "dragend", function(event) {
  var point = marker.getPosition();
  map.panTo(point);
  jQuery('#lat').val(event.latLng.lat().toFixed(3));
  jQuery('#lng').val(event.latLng.lng().toFixed(3));
});

```

For å velge start og sluttidspunkt til auksjonen har vi benyttet jQuery UI addonen `datetimePicker`. Feltene blir initiert på følgende måte i JS fila

```

jQuery('#starttime').datetimepicker({
  numberOfMonths: 2,
  minDate: 0,
  hourGrid: 2,
  minuteGrid: 30,
  hour: 18,
  minute: 0,
  second: 0,
  dateFormat: 'yy-mm-dd',
  timeFormat: 'hh:mm:ss',
  onClose: function(dateText, inst) {
    var endDateTextBox = jQuery('#endtime');
    if (endDateTextBox.val() != '') {
      var testStartDate = new Date(dateText);
      var testEndDate = new Date(endDateTextBox.val());
      if (testStartDate > testEndDate)
        endDateTextBox.val(dateText);
    }
    else {
      endDateTextBox.val(dateText);
    }
  },
  onSelect: function(selectedDateTime){
    var start = $(this).datetimepicker('getDate');
    jQuery('#endtime').datetimepicker('option', 'minDate', new Date
(start.getTime()));
  }
});

jQuery('#endtime').datetimepicker({
  numberOfMonths: 2,
  hourGrid: 2,
  minuteGrid: 30,
  hour: 18,
  minute: 0,
  second: 0,
  dateFormat: 'yy-mm-dd',
  timeFormat: 'hh:mm:ss',
  onClose: function(dateText, inst) {
    var startDateTextBox = jQuery('#starttime');
    if (startDateTextBox.val() != '') {
      var testStartDate = new Date(startDateTextBox.val());
      var testEndDate = new Date(dateText);
      if (testStartDate > testEndDate)
        startDateTextBox.val(dateText);
    }
    else {
      startDateTextBox.val(dateText);
    }
  },
  onSelect: function(selectedDateTime){
    var end = $(this).datetimepicker('getDate');
    jQuery('#starttime').datetimepicker('option', 'maxDate', new Date
(end.getTime()));
  }
});

```

Vi har sørget for at det valgte datoformatet er i standard MySQL datetime format. Det er i tillegg ikke lov å velge en sluttdato mindre enn startdatoen.

Når formen sendes lagres dataene i databasen og det sendes også ut en bekreftelses email til selgeren på at forslaget er registrert og eller publisert.



## APPROVEAUCTION

Brukes av administratoren til å godkjenne annonser som brukere av siden har lagt inn. For å sjekke at brukeren faktisk er en administrator har vi lagt inn følgende sjekk i «Custom server side validation» eventen som kjøres når formen lastes inn.

```
<?php
$user = JFactory::getUser();
$groups = JUserHelper::getUserGroups($user->id);
if(empty($groups[8])) //If not a member of the super admin group
    return false;
?>
```

Koden over sjekker om brukeren er medlem av superadministrator gruppen. Et alternativ ville vært å definere hva en admin bruker er. En som kan installere nye programmer? En som kan redigere innhold? For å sjekke brukerens rettigheter benytte Joomla sin authorize() funksjon som er tilknyttet Joomla sin ACL.

```
$user = JFactory::getUser();
//Checks if the user can edit all content
if(!$user->authorize('com_content', 'edit', 'content', 'all'))
    return false;
```

Som vil sjekke om brukeren har lov til å redigere alt innhold på siden.

For å laste data inn i feltene i formen har vi brukt «DB Record loader». Denne settes opp slik at den henter ut en ID fra URL-en og laster inn data fra raden med tilsvarende id i auksjonstabellen. «DB record loader» legger kolonne verdiene inn i feltene med tilsvarende navn. For å oppdatere raden i databasen er vi også nødt til å laste inn auksjonens id slik at vi ikke bare lagrer en ny annonse når vi trykker send. Dette gjøres ved å legge til et gjemt felt med navnet cf\_id. Vi har også lagt til en radio box hvor administratoren kan velge om han ønsker å publisere eller slette auksjonen. For å skille mellom slettet og godkjent i «On submit» eventen har vi lagt til følgende kode i en «Custom server side validation» event.

```
<?php
if($_POST['published'] == -1){
    $db = JFactory::getDBO();
    $q = 'DELETE FROM #__auctions WHERE cf_id = '.$_POST['cf_id'];
    $db->setQuery($q);
    $db->query();
    return false;
}
?>
```

Hvis administratoren velger å ikke publisere annonsen blir auksjonsdata slettet og en email sendes ut til brukeren. Hvis den blir godkjent vil annonsen publiseres på siden og selgeren underrettet om dette.

#### CHANGEBIDSCHEMA/CHANGEREDOCUMENT/CHANGETARIFF

Brukes til opplasting/endring av salgsoppgave, takst og budskjema. Det er en separat form for alle dokumentene.

Både administratorene og eieren av auksjonen skal ha tilgang til formen noe som sjekkes med følgende kode.

```
<?php
$user = JFactory::getUser();
$groups = JUserHelper::getUserGroups($user->id);
$db = JFactory::getDBO();
$q = 'Select cf_user_id FROM #__auctions WHERE cf_id =
'.JRequest::getInt('id');
$db->setQuery($q);
$res = $db->loadResult();
//If not auction owner or super admin
if(empty($groups[8]) && $res != $user->id)
    return false;
?>
```

Her må vi hente ut auksjonseierens brukerid og sammenligne denne med den påloggede brukers brukerid, Vi sjekker samtidig om brukeren er en administrator.

Dersom brukeren har lastet opp en gammel fil vil det lenkes til denne øverst. Vi laster inn filnavnet til den gamle filen i et felt som er gjemt for brukeren.

Når brukeren laster opp en ny fil vil vi slette den gamle filen dersom brukeren allerede har lastet opp en fil.

```
<?php
$oldfile = 'images/userfiles/'.$_POST['oldfile'];
if(file_exists($oldfile)) //If file exists
    unlink($oldfile); //Delete file
?>
```

De resterende formene har blitt beskrevet med high leves usecases i kapittel 2.4.

### 4.6.3 ANDRE UTVIDELSER

#### AKEEBA BACKUP [26]

Er et program som gjør det enkelt å ta backup og flytte hele Joomla sider. Vi har brukt Akeeba Backup til å ta kopier av Joomla siden og installere den lokalt på våre datamaskiner mens vi har jobbet med utviklingen

#### EASYCREATOR [27]

Programmet er laget for å forenkle komponentutviklingsprosessen i Joomla. Vi har hovedsakelig benyttet easyCreator til pakking og versjonskontroll.

## 4.7 ANDRE BESLUTNINGER

### 4.7.1 VALG AV KARTTJENESTE:

Vi visste i fra begynnelsen at oppdragsgiver ønsket å få implementert karttjenester på siden. Vi var derfor nødt til å bestemme oss hvilken karttjeneste vi ønsket å benytte.

Thomas hadde på forhånd en del erfaring med bruk av Google Maps JavaScript API og hadde gode erfaringer med dette. Vi var også klare over at Google skulle innføre bruksbegrensninger på Google Maps i begynnelsen av 2012 [28]. Denne grensen ligger i dag på 25,000 spørringer per dag, noe vi så på lite sannsynlig for at vi kom til å overstige. Dersom denne grensen overstiges vil dette koste \$4 per 1,000 spørringer over 25,000.

#### Prisliste

| Service                                    | Usage limit (per day) | 1,000 excess map loads (in U.S. dollars) |
|--|-----------------------|--|
| JS Maps API v3                             | 25,000                | \$4                                      |
| JS Maps API v3 <a href="#">styled maps</a> | 2,500                 | \$4[1] / \$8[2]                          |

|   |        |                 |
|---|--------|-----------------|
| Static Maps API                             | 25,000 | \$4             |
| Static Maps API <a href="#">styled maps</a> | 2,500  | \$4[1] / \$8[2] |
| Street View Image API                       | 25,000 | \$4             |
| JS Maps API v2                              | 25,000 | \$10            |

Prislisten er hentet fra [29]

Etter å ha sett på andre alternative karttjenester som Open Streetmap, Mapnik og QGIS, bestemte vi oss relativt fort for at vi skulle bruke Google Maps, selv om mange av alternativene virket solide. Først og fremst fordi Google Maps er raskt, responsivt, det ser bra ut, kommer med mye godt dokumentert funksjonalitet, støtte for håndholdte enheter og mer.

---

#### 4.7.2 VALG AV SMS-TJENESTE

Oppdragsgiver var interessert i å få implementert SMS-tjenester i løsningen. I form av automatisk varsling på SMS for de som ønsket å følge budrunder og lignende.

Siden en egen SMS router ikke var et alternativ har vi sammenlignet prisene og funksjonaliteten til forskjellige tjenestetilbydere. Under har vi tatt med en prisinformasjon og noen få kommentarer til de aktuelle tilbyderne.

#### SENDENGA [30]

- Må benytte oss av Enterprise pakken for å kunne integreres i systemet
- 1990,- i etableringskostnad, 499,- pr måned, 0.39 pr SMS
- Flere kjente kunder inkludert Telenor, tine aker m. flere

#### PSWIN [31]

- 990,- i etableringskostnad, 190,- pr måned, 0.65,- SMS
- Flere kjente kunder
- Dokumentasjon tilgjengelig på deres nettsider

#### NETCOM [32]

- 0,- i etableringskostnad, 399,- pr måned, 0.65,- pr SMS
- Mulig å forhandle seg til bedre priser

#### KEYSMS [33]

- 0,- i etableringskostnad, 299,- pr måned, 0.39,- pr SMS
- 199,- ekstra i mnd. for å bruke bedriftens navn som avsender
- Flere kjente kunder
- Ser veldig enkelt ut å implementere

Det vil i tillegg komme visse implementeringskostnader dersom oppdragsgiver i etterkant ønsker å implementere dette på nettsiden. Siden vi selv aldri har implementert noe lignende føler vi at det er relativt vanskelig å si noe om hvor lang tid det eventuelt vil ta og implementere dette på siden. Etter å ha sett igjennom de tilgjengelige APIene vil vi anslå at det vil ta 1-2 arbeidsdager å få implementert SMS tjenester på siden. Mesteparten av funksjonaliteten man behøver for å legge til varsling på mobil er allerede implementert på siden, den er bare ikke tatt i bruk.

---

#### 4.7.3 HOSTING

Til utviklingen av vårt prosjekt var det nødvendig med en database. Høgskolen i Gjøvik tilbyr en MySQL-database til alle prosjektgruppene, og det hadde derfor i utgangspunktet ikke vært nødvendig med en egen hosting-tjeneste. Siden vi opprettet prosjekthjemmesiden med Wordpress CMS valgte vi å bruke denne tilbydte databasen til dette. Vi kunne også benyttet den samme databasen til utviklingen av selve prosjektet, men siden vi her tok utgangspunkt i Joomla! CMS ville dette skapt et svært stort antall tabeller som kunne vært problematisk å holde orden i. Med tanke på ryddighet valgte vi derfor å se etter andre hosting-muligheter.

Ved valg av hosting-tjeneste var det primært fem faktorer som spilte en rolle:

- Pris per kapasitet.
- Støtte for nyere versjoner av PHP (PHP 5 og over).
- Ubegrenset trafikk til nettstedet.
- MySQL-database.
- Oppetid

Med utgangspunkt i dette valgte vi å bruke one.com som vårt web-hotell. Dette fordi one.com tilbyr all den nødvendige funksjonaliteten, i tillegg til å være prisvennlige. One.com har også svært høy oppetid[34].

---

#### 4.7.4 LOGIN

Til innlogging på selve nettsiden valgte vi å bruke standardmodulen som følger med Joomla!. Det finnes mange utvidelser for nettopp dette, men vi mente standardmodulen inneholdt all nødvendig funksjonalitet. Vi ville derfor heller modifisere og tilpasse denne til vår løsning. Hovedmodifikasjonen besto i at vi ønsket en «popup»-funksjonalitet for selve login funksjonen. Dette kunne enklest gjøres ved hjelp av en plugin. Her valgte vi å bruke RokBox[35] fordi denne var tilpasset nyeste versjon av Joomla!, samt at mye dokumentasjon på hvordan lage slik «popup»-funksjonalitet var tilgjengelig med denne plugin. I sin enkleste form kan man lage en popup-link ved:

```
<a rel='rokbox' href='...'>Eksempel</a>
```

---

#### 4.7.5 CRON-JOBS

En cron-job er en tidsplanlagt arbeidsoppgave i unix-liknende operativsystemer. Veldig mange servere i dag bruker slike operativsystemer. Cron gjør det mulig å planlegge arbeidsoppgaver slik at disse kjører på bestemte tider eller datoer. I vår løsning ønsket vi å sende automatisert e-post til abonnenter av en auksjon for å varsle om auksjonsstart og auksjonsslutt. Ideen var å kjøre en slik cron-job med jevne intervaller for å sjekke auksjonsstart og auksjonsslutt verdiene i databasen for å avgjøre om e-post skulle sendes til abonnentene. Mange hosting-tilbydere tillater kundene sine å kjøre cron-jobs. Dette gjaldt ikke vår hosting hos one.com, og vi var derfor nødt til å se etter andre muligheter.

Vi søkte derfor etter utvidelser til Joomla! med den samme funksjonaliteten. Løsningen ble «JPrC Cronjobs»[36] da dette var den eneste utvidelsen som var tilpasset Joomla! 2.5. Utvidelsen fungerer ved at man oppretter jobber i Joomla! backend som starter ved valgte tidspunkter. Disse jobbene kan kjøre helt egne filer eller SSH-kommandoer. Det må nevnes at denne utvidelsen ikke er noen fullverdig erstatning for Cron. Utvidelsen krever nemlig at det er trafikk på nettsiden for å kjøre de planlagte oppgavene. Ved stor og jevn trafikk vil dette være tilnærmet lik Cron, men ved lavere og mer sporadisk trafikk kan jobbene avvike noe fra de planlagte tidspunktene. Dette anser vi som akseptabelt da vi ikke kunne finne et bedre alternativ. BoligHandel1 benytter domene.no som hosting-leverandør og det er heller ikke her mulig og kjøre Cron-jobs. Vår løsning er kun tilknyttet Joomla! og er derfor plattformuavhengig. Løsningen vil derfor også fungere etter eventuell implementasjon hos BoligHandel1.

#### 4.8 AUTENTISERINGSTEKNOLOGIER

Autentisering er en metode for å avgjøre om brukeren er den han/hun utgir seg for å være. For at budgivningen skal både være sikker for selger samt juridisk bindende for kjøper er vi avhengig av en fungerende autentiseringsteknologi. En slik teknologi innebærer vanligvis at kjøper bekrefter sin identitet ved et brukernavn (fødselsnummer er ofte brukt) og et personlig passord. Det kreves også gjerne en fysisk enhet som er unik for den aktuelle kjøperen. Dette kan blant annet være en kodebrikke, et elektronisk id-kort, eller et papir med informasjon.

For å utvikle en autentiseringsløsning som holder sikkerhetsmessige krav, kreves spesiell kompetanse. Dette er kompetanse som vi i prosjektgruppa ikke har, og vi må dermed basere oss på å ta i bruk eksisterende teknologi. Det er dessuten lite hensiktsmessig å bruke mye tid og ressurser på å utvikle en slik løsning, da det allerede finnes flere gode autentiseringsteknologier som er tilgjengelig for private aktører/bedrifter. Slike løsninger er derimot betalingstjenester, og siden oppdragsgiver ikke har gitt økonomiske midler til prosjektet, må vi se bort fra selve implementasjonen av en slik autentisering. En fullverdig

nettauksjon må likevel inneholde dette, og vi vil derfor redegjøre for valg av autentiseringsløsning og hvordan denne kan implementeres.

---

#### 4.8.1 HVILKE TEKNOLOGIER FOR AUTENTISERING FINNES?

##### BIOMETRI

Biometri er måling av fysiologiske data, f.eks fingeravtrykk. Bruk av biometri kan være et godt bidrag til å utvikle sikre løsninger, men da aller helst i kombinasjon med andre metoder for å bekrefte at en person er den han gir seg ut for å være.

##### BANKID

BankID er en personlig og enkel elektronisk legitimasjon for sikker identifisering på nett. Det finnes to typer BankID, banklagret og mobillagret. Med banklagret BankID benyttes fødselsnummer og et personlig passord, samt en kode fra en kodebrikke. Med mobillagret BankID benyttes også fødselsnummer, samt mobilnummer og pinkode (tilknyttet sim-kort).

##### MINID

MinID er sammen med BankID de to mest utbredte autentiseringsløsningene i Norge. MinID benytter i likhet med BankID fødselsnummer og personlig passord, samt et kodebrev eller evt kode tilsendt på mobil for sikker autentisering. De fleste er nok kjent med MinID i forbindelse med elektronisk levering av selvangivelsen. MinID brukes til innlogging på mer enn 200 offentlige tjenester. MinID er ikke aktuelt til vår løsning siden det kun tilbys til offentlige etater.

##### BUYPASS ID

Buypass ID er nok en autentiseringsløsning med stor utbredelse. Autentiseringen gjennom buypass fungerer ved at man får utstedt en elektronisk ID på smartkort av Buypass AS, og benytter dette sammen med en pin-kode.

---

#### 4.8.2 FORDELER OG ULEMPER VED DE AKTUELLE TEKNOLOGIENE



## BIOMETRI

### Fordeler:

- Raskt og brukervennlig. Man trenger ikke ha med seg et kort eller huske noen kode.
- I kombinasjon med andre autentiseringsmetoder tilbys svært høy sikkerhet.

### Ulemper:

- De fleste datamaskiner er ikke utstyrt med fingeravtrykksleser. Dette må da kjøpes som en separat enhet med for eksempel usb-tilkobling.
- Det finnes ingen komplett ferdigløsning.
- Krever en komplett tilgjengelig database av avleste fingeravtrykk (finnes ikke).

## BANKID

### Fordeler:

- Tilbyr høyeste sikkerhetsnivå (nivå 4).
- Svært utbredt.
- Enkelt å implementere (ifølge BankID's egne hjemmesider). Kan gjøres selv eller med hjelp av en BankID Partner.

### Ulemper:

- Kan bli kostbart (særlig for små bedrifter, selv om prisene er forhandlingsbare).

## BUYPASS ID

### Fordeler:

- Også enkelt og implementere. Implementeringsguide medfølger dersom man vil gjøre integrasjonen selv, eller man kan gjøre dette i samarbeid med en integrasjonspartner.
- Holder høyeste sikkerhetsnivå.

### Ulemper:

- Ikke like utbredt som BankID.
- Kan også bli kostbart for små bedrifter.

---

#### 4.8.3 VALG AV AUTENTISERINGSLØSNING

Proessen rundt valg av autentiseringsløsning har vært relativt kort. For at budgivningen på auksjonene skal appellere til en størst mulig brukergruppe har vi valgt BankID som autentiseringsløsning. Dette fordi BankID med overkant av 2,7 millioner brukere[37] utvilsomt har størst brukergruppe blant de tilgjengelige teknologiene. Til sammenlikning har BuypassID omlag 2 millioner brukere[38]. Kostnadmessig opererer både BankID og Buypass med tilsynelatende samme veiledende priser for etablerings- og månedsavgift.

Transaksjonsavgiften per autentisering er en del lavere hos Buypass enn hos BankID, men argumentet om brukergruppe bidrar likevel til at vi velger BankID fremfor Buypass. Dessuten er de fleste i dag kjent med BankID gjennom nettbank.

BankID blir i dag også brukt i andre auksjoner, for eksempel hos [www.qxl.no](http://www.qxl.no). Dette var også et argument for å velge BankID, da BoligHandel1 benytter nettopp QXL til å auksjonere sine boliger og da er fortrolige med dette systemet. BoligHandel1 synes å være enig i at BankID er den beste autentiseringsløsningen.

---

#### 4.8.4 VEIEN VIDERE

Nettsteder som benytter BankID kalles gjerne BankID brukersteder. BankID eies av bankene i fellesskap gjennom BankID Norge AS og tjenesten driftes av Nets Norge Infrastruktur.

Bankene kan inngå avtaler med sine kunder om at de skal bli brukersteder. For å bli et BankID brukersted er det flere steg man må gjennom. Disse stegene kan variere litt fra bank til bank, men i hovedsak fungerer det slik:

1. Brukerstedet inngår en avtale med ønsket bank om bruk av BankID.
2. Valgte bank utleverer BankID programvare og dokumentasjon til brukerstedet.
3. Brukerstedet er så selv ansvarlig for implementering og drift av BankID.

4. Når brukerstedet har implementert BankID, skal dette testes mot banken. Dette gjøres ved at brukerstedet kontakter banken og bestiller BankID testsertifikater for online test.
5. Banken gir brukerstedet tilgang til BankID-testsertifikater med passord.
6. Brukerstedet gjennomfører tester mot BankID. Brukerstedet må dokumentere at testen er gjennomført og godkjent ved å signere «Certificate of Conformity» (COC). Banken kontrollerer så dette.
7. BankID produksjonssertifikater bestilles og distribueres.
8. Brukerstedet tar i bruk BankID.
9. Banken er ansvarlig for at informasjon om brukerstedet offentliggjøres i brukerstedsoversikten på BankIDs hjemmesider.

Programvaren som utleveres av banken kalles «BankID server for brukersteder».

Programmet leveres på en egen CD, enten som Java-versjon eller C-versjon. Java-versjonen er plattformuavhengig, mens C-versjonen kjøres på Windows-servere. Selve implementeringen kan gjøres av den aktuelle bedriften selv eller i samarbeid med en BankID-partner. Blant disse BankID-partnerne finner vi store konsulentselskaper som Bekk-Consulting, Avento, Evry m.m.

For å vite mer om kostnadene rundt BankID sendte vi forespørsler til både Totens Sparebank og DNB. Totens Sparebank meddelte at det er få banker som legger ut disse prisene, og refererte videre til Nordeas hjemmesider med følgende pristabell:

| Spesifikasjon  | Pris (ekskl. MVA) |
|--|-------------------|
| Oppstart/etablering BankID server  | Kr 10 000,00      |
| Månedspris BankID server   | Kr 1 000,00       |
| Transaksjonspris BankID Signering  | Kr 7,00           |
| Transaksjonspris BankID Autentisering                                      | Kr 3,00           |
| Transaksjonspris BankID Tilleggsinformasjon – forespørsel på fødselsnummer | Kr 3,00           |

Disse prisene er veiledende og vil variere ut i fra brukerstedets størrelse, og vil samtidig være bankindividuelle. Prisene antas også å være relativt høye og et utgangspunkt for forhandling. Et stort brukersted med mange transaksjoner vil nok forsøke å senke transaksjonsprisen, mens etableringskostnaden vil være grei. For et brukersted med færre transaksjoner vil sannsynligvis etablerings- og månedsprisen være gjenstand for diskusjon.

Dersom BoligHandel1 i etterkant ønsker å gå videre med vår løsning er de avhengig av implementasjon av BankID på nettstedet. Siden de ikke har relevant kompetanse til å gjøre dette selv, må de sannsynligvis leie inn en BankID-partner. Det er derfor svært vanskelig å estimere denne implementasjonskostnaden da ulike konsulentselskaper opererer med veldig ulike timesatser. Siden det også finnes svært mange BankID-partnere kan det også være tidkrevende og finne de rimeligste konsulentene. Det er også vanskelig å anslå hvor mye arbeid som kreves for å gjennomføre implementasjonen, og også dette vil variere fra konsulent til konsulent. Med utgangspunkt i en e-post fra Buypass anslår vi at alt fra en hel arbeidsdag til en hel arbeidsuke vil være normalt for en god utvikler. Med en konsulent-timespris på ca. kr 1 500,- kan denne implementasjonskostnaden havne på alt fra kr 10 000,- til kr 60 000,-.

Den tiltenkte bruken av BankID i vår totalløsning vil være at når en budgiver angir et bud og trykker på knappen for å lagre dette budet, vil autentiseringshendelsen kjøre. Dersom autentiseringen godkjennes vil budet lagres i databasen og dermed være juridisk bindende. Grunnen til at vi velger å plassere autentiseringen til dette tidspunktet er for å holde antall autentiseringstransaksjoner nede på et minimumsnivå, og dermed redusere kostnader. Vi anser det som sannsynlig at hyppigheten på budgivninger vil være mindre enn hyppigheten på for eksempel innlogginger til nettstedet.

## 5. TESTING

### 5.1 Utvikling

Vi har basert arbeidet vårt på en inkrementell systemutviklingsmodell, dermed har vi hele veien fått muligheten til å teste hvert lille steg fremover. Vi har alle sammen hatt en kopi av webløsningen lokalt, ved bruk av EasyPHP, og utviklet på egen maskin og testet fortløpende før vi som gruppe har lagt det til på webløsningen. Ved å arbeide på denne måten, unngår man konflikter rundt for eksempel bruk, endring og oppdatering av tabeller i databasen, oppdatering av menylenker og lignende på siden og generelt sidestruktur.

### 5.2 Eksternt

Våre oppdragsgivere har også hatt egne brukerkontoer på webløsningen, slik at de har kunnet komme med nye ønsker, nye krav eller endringer på eksisterende innhold fortløpende. Dette har fungert meget bra og har åpnet muligheten for å overse og teste løsningen slik at de har kunnet komme med innspill når som helst, og igjen holde nødvendigheten for tidkrevende møter til et minimum.

### 5.3 AUKSJONER

Når utkastet til auksjonsbiten var ferdigstilt, har både gruppa og oppdragsgiver kjørt gjennom flere auksjoner på jakt etter feil, og ikke minst testing av hendelsesflyt for oppdragsgiver sin del.

### 5.5 FOKUS

Det er urealistisk pr. i dag at BoligHandel1 skal ha mange hundre eiendommer til salgs samtidig på websiden. Derfor har vi testet i intervallet en til femti eiendommer samtidig. Fokuset rundt testing av auksjonsfunksjonaliteten har vært stabilitet. Selv om man ikke kan garantere seg mot feil på leverandørsiden, serverfeil etc. er det viktig at en auksjon fungerer 100 prosent. Feilsituasjoner som oppstår rundt auksjonsslutt kan få store konsekvenser spesielt for selger og skulle dette være aktuelt for markedet ville man måtte kjøre intensiv

testing en periode for å plukke opp alle feil som måtte ligge rundt denne biten. Slik at det ved publisering er meget grundig testet.

## 6. AVSLUTNING

### 6.1 DISKUSJON AV RESULTATER

#### AUKSJONSSYSTEMET

Kjernen i oppgaven var å implementere et auksjonssystem for BoligHandel1 på egen side som skal kunne erstatte QXL.no sin funksjon i deres daglige drift. Ved å få dette på sin egen side vil det jo ikke bare erstatte funksjonaliteten QXL.no nå har, men de vil også ha mye mer kontroll over hendelsesflyten så den kan tilpasses nettopp kjøp og salg av eiendom.

Selv om det naturligvis er mye forbedringspotensialet både i auksjonssystemet og på siden generelt, har vi fått opp en fungerende prototyp som kan gjennomføre auksjoner slik vi og oppdragsgiver har ønsket.

Ved å arbeide etter en inkrementell modell har vi og oppdragsgiver gjennom møter hele veien tilpasset funksjonaliteten og hendelsesflyten til auksjonssystemet utfra ønsker de har kommet med og de mulighetene som ligger i Joomla! og gruppas generelle kompetanse.

#### FREMVISNING AV EIENDOMMER

Vi har valgt å vise frem eiendommer slik de fleste andre kjente sider for kjøp og salg gjør det. For å nevne noen, så viser Finn.no, ebay.com og QXL.no frem sine salgsannonser mer eller mindre identisk, og dette er blitt som en standard å regne og vi så ingen grunn til og ikke gjøre det nettopp slik. Dette gjør at eiendommer viser oversiktlig frem med valgt forsidebilde samt de viktigste opplysningene.

Seksjonen med søkeparameterne er også oversiktlig og selvforklarende, og vil gi små rom for menneskelige feil ved bruk. Disse parameterne avgjør naturligvis hva som vises frem i listen, og fungerer som den skal.

Ved å klikke på en eiendom i listen, vil man komme inn på informasjonssiden om akkurat denne eiendommen. Denne siden er også oversiktlig og viser frem de bildene selger har valgt til annonsen øverst, samt relevante opplysninger rundt objektet. Det er også mulig å lese dokumenter vedlagt eiendommen, og totalt sett gjør dette at all informasjon er lett tilgjengelig for potensielle kjøpere.

#### LEGG TIL AUKSJONER

Med brukerprofil har man mulighet til å legge til en auksjon. Her bruker vi bare et enkelt, men relativt stort skjema som gir selger mulighet til å fylle ut, linke til og legge ved alt av informasjon som vil være nyttig og som kreves for denne eiendommen.

#### ANNEN FUNKSJONALITET

Selv om auksjonsfunksjonaliteten er essensen med oppgaven, var det også flere ønsker som gjorde dette til en komplett nettløsning for BoligHandel1. Annen funksjonalitet på siden, samt designmessige ønsker har gjort at oppgaven kan presenteres som en ferdig nettside. Når vi sier ferdig, mener vi på ingen måte at den er klar for bruk, men mer enn prototype eller et utkast til et ferdig system.

Vi har valgt å legge all funksjonalitet i frontend, slik at backend ikke vil brukes til daglig drift av siden. Man trenger ikke som BoligHandel1 sin webansvarlige og logge inn i backend, og bli overøst av konfigurasjonsmulighetene som ligger der, for å gjøre de endringene og utføre de arbeidsoppgavene som kreves.

En bruker med administrator-rettigheter vil kunne legge inn nye, endre eller slette produkter BoligHandel1 tilbyr eller har tilbydd. Han vil også kunne opprette auksjoner eller endre eksisterende. I tillegg kan han bestemme innholdet i sliderne på forsiden.

Som nevnt så vidt over, valgte vi legge dette i frontend slik at man eksklusivt fikk tilgang til endring av den aktuelle funksjonaliteten uten at alle konfigurasjons- og designmulighetene som ligger i backend. Dette gjør at hvem som helst i BoligHandel1 kan gjøre de endringer og oppdateringer som trengs i den daglige driften.

#### DESIGN

Vi har hatt som mål og lage designet stilrent og moderne, men allikevel svært enkelt og lite innhold av fancy løsninger og stylinger. Designet på en side til dette formålet må oppleves som seriøs for eventuelle brukere. Dette er jo i høyeste grad et definisjonsspørsmål, men igjen kan man se til større sider som folk kjenner til, og med det i bakhodet har vi valgt slikt design.

Som nevnt tidligere er denne siden med funksjonaliteten som en prototyp eller utkast og regne. Det vil si at den er veldig "rein og pen" uten noe form for ekstern eller intern reklame bortsett ifra slidere på forsiden. Ved å lage siden såpass ren og pen, ligger det til rette for at BoligHandel1 vil kunne påvirke reklame-innhold, eller annen informasjon som vises ellers på siden.

#### BRUK AV CMS I LØSNINGEN

Ved valg av et CMS, og i vårt tilfelle Joomla!, har vi allerede låst oss litt til hvordan ting kan gjøres og se ut. Ved å starte denne løsningen fra bunn, vil man i mye høyere grad akkurat kunne tilpasse hvordan ting skal gjøres og se ut. Allikevel setter ikke Joomla! noen begrensninger i vårt tilfelle, og det å utvikle en løsning basert på noe proft som Joomla!, øker sikkerheten og integriteten til siden automatisk.

## 6.2 FORBEDRINGSPOTENSIAL

Ved en ferdig publisert løsning på nett, er det noen punkter som må og noen som bør være med:

#### AUTENTISERING

Ved kjøp og salg av eiendom, må man forholde seg norsk lov og bestemmelser rundt eiendomshandel. På grunn av dette er det avgjørende med en sikker autentiseringsløsning som er godkjent til bruk i Norge. Som for eksempel Buypass eller BankID. Uten en godkjent autentiseringsløsning vil denne løsningen ikke kunne brukes til eiendomshandel.

#### HVERDAGEN TIL WEBANSVARLIG I BOLIGHANDEL1



En slik side vil være veldig strukturert, med få eller ingen innslag som skal minne om en personlig side. Derfor vil det å lage ferdige maler tilgjengelig som menyvalg i administrasjonsaksjonen kunne gjøre det svært enkelt å vedlikeholde innhold på siden. Ferdig strukturerte maler for nyheter, offentlig informasjon og annen relatert informasjon samt ferdigkonfigurerte moduler hvor man kan legge inn ekstern og intern reklame vil kunne gjøre det daglige arbeidet med nettsiden svært effektivt.

#### GENERELT

Siden er som sagt et utkast pr. i dag, og en eventuelt ferdigstilt side vil inneholde forskjellig innhold som intern og ekstern reklame, mer offentlig informasjon rundt eiendomshandel og annen informasjon valgt av BoligHandel1. Dette er ikke forbedringspotensial vi hadde ønsket og hatt med i innleveringen, siden dette vil være vesentlig ved en publisert og offentlig løsning. Vi har valgt og ikke bruke tid med oppdragsgiver på ting som kan beskrives som styling av og diverse ekstra innhold på siden.

### 6.3 VIDERE ARBEID

#### UTSEENDE

Hva siden mangler når det kommer til design og utseendet vil være opp til arbeidsgiver å avgjøre, men skulle dette vært realisert er det et par punkter som må med.

#### AUTENTISERING

Som nevnt tidligere må man ha en form for autentiseringsløsning, for å kunne bedrive eiendomshandel i Norge. Eller andre land for den saks skyld. Ved bruk av en autentiseringsløsning, vil det også være strenge krav til maskinvare og vedlikehold av denne. Derfor ville det vært aktuelt for BoligHandel1 og skaffe seg server for denne bruken.

#### DAGLIG DRIFT

Pr. i dag innehar ingen i BoligHandel1 kompetansen som skal til for å implementere en autentiseringsløsning på sin egen side eller vår webløsning, så her trenger man ekstern hjelp. Når autentiseringsløsningen er på plass, vil nok BoligHandel1 med hjelp av kurs eller

lignende kunne tilegne seg kompetansen som skal til for å kunne drifte dette systemet uten ekstern hjelp.

## 6.4 EVALUERING AV ARBEID

### ORGANISERING

I forprosjektet ble det lagt opp et tenkt prosjektløp, hvor vi satt opp inkremitter og fordelte disse på gruppa. Dette ble satt opp før vi bestemte oss for å bruke et CMS. Det ble naturlig nok litt endringer på inkrementene, da vi fikk et CMS som gjorde oppsettet og rammeverket for siden for oss, men på den annen side gikk tiden vi sparte på å sette opp rammeverket med på å sette oss inn i Joomla! I stedet.

Da gruppeleder innehar kompetansen med php og JavaScript ifra tidligere, ble det hans oppgave og sette seg inn i Joomla! Først, for så å fordele kunnskapen videre etter hvert. De andre gruppemedlemmene brukte første tiden på å gjennomgå det grunnleggende i nevnte skriptsspråk og tilegne seg den kunnskapen som behøvdtes for å gå i gang med Joomla!.

### ARBEIDSFORDELING

Som nevnt hadde gruppeleder erfaring med php og JavaScript ifra tidligere, derfor falt mye av utviklingen i starten og spesielt det med auksjonsfunksjonaliteten på han.

For de andre gruppemedlemmene gikk det med en del tid i starten til å lære seg de nevnte skriptsspråkene, slik at man etter hvert kunne sette seg grundigere inn i Joomla!. Vi har jobbet etter en inkrementell modell, og vi har fordelt arbeidet forløpende ved starten av neste inkrement.

## 6.5 ERFARINGER

### PHP OG JAVASCRIPT

Vår kompetanse rundt denne overskriften strakk seg til 1/3 av gruppen. En av medlemmene har gjennom faget WWW Teknologi vært gjennom grunnleggende og mer avanserte ting i disse skriptspråkene.

For de to andre i gruppen, har erfaringen med å måtte lære seg dette underveis, samtidig som man utvikler en løsning vært en god erfaring. I etterkant er det svært enkelt å se tilbake på arbeidet man har gjort og se hva man kan gjøre bedre og mer effektivt neste gang. I IT-bransjen vil man kunne komme borti jobber hvor man er avhengig av å lære seg noe nytt underveis, så derfor er vi meget glad for erfaringen vi har fått rundt dette.

### CMS (CONTENT MANAGEMENT SYSTEM)

CMS og Joomla! Har kommet for å bli. Ingen i gruppa har tidligere arbeidet med et CMS, så her måtte alle gå grundig til verks for å få tak på hvordan i vårt tilfelle Joomla! fungerer. Som et CMS er det nok svært lite man ikke kan løse med Joomla!, men litt av problemet våres ble dokumentasjonen rundt de litt mer avanserte mulighetene ved å bruke denne løsningen. Dette gjorde at å få full oversikt var litt trøblete i starten, og det gikk med mye tid på prøving og feiling, gjennomgang av annen åpen kildekode som omhandler Joomla! samt søk etter nyttig dokumentasjon både på bibliotek og på web. Vi føler dette har gitt oss gode erfaringer og hjulpet oss til å finne god metodikk og gjort oss bedre rustet hvis man skulle komme over noe liknende senere i skole eller arbeidslivet.

### MVC (MODEL-VIEW-CONTROLLER)

MVC er jo ikke forbeholdt Joomla! alene og er mye brukt i program- og applikasjonsutvikling rundt omkring i verden. Å bli kjent med ulike designmønstre er også svært nyttig og bidrar til å utvide perspektivet for hvordan kunne løse en problemstilling når det kommer til selvstendige oppgaver man måtte komme over senere.

### UTVIKLING OG DOKUMENTERING

Når man jobber med et større prosjekt, hvor antall filer og kodelinjer blir svært mange, er det viktig med

## 6.6 KONKLUSJON

### EN REELL OPPGAVE

Å jobbe med et større reelt prosjekt med ekstern oppdragsgiver har vært en meget god erfaring, og veldig viktig som en innføring til liknende prosjekter man vil støte på ute i arbeidslivet. Ser man på rapportdelen av denne oppgaven, har vi mer eller mindre gjort denne delen av bacheloroppgaven i faget Systemutvikling. Forskjellen er at her har vi selv gjort det vi skriver om, noe vi fant enklere og mer lærerikt enn å skrive rundt et fiktivt prosjekt.

### OVERSETTE ØNSKER OG KRAV

Vår oppdragsgiver har som nevnt i rapporten benyttet Joomla til design av sin webside. Men bortsett ifra de ferdige malene som ligger i Joomla, har de ingen kjennskap til hvordan Joomla fungerer bak det ytre, eller de forskjellige kode- og skript-språkene som benyttes i Joomla og på web. Det har vært svært lærerikt og hele tiden måtte “oversette” oppdragsgivers krav og ønsker til en teknisk løsning. Gjennom skriftlig og muntlig kommunikasjon gjennom mail og møter har vi hele tiden utformet krav og ønsker fra “munn til web”. Vi tror mulighetene for å støte på noe slikt senere gjennom konsulentvirksomhet i IT-bransjen er svært sannsynlig, og er glade for at vi fikk erfaring med en slik oppgave.

### TILEGNE SEG KUNNSKAP UNDERVEIS

På det tidspunktet vi startet med prosjektet hadde ingen av oss erfaring med CMS (Content Management System), og to av tre hadde ingen erfaring med verken php eller JavaScript som skriptspråk, samt samhandlingen mellom disse på dynamiske websider og videre samhandling mellom php og relasjonsdatabasen MySQL i vårt tilfelle. Å kalle Joomla! i seg selv “en del av et fagområde” vil ikke medføre riktighet, men ved å se på Joomla! som en innføring i bruk av CMS, føler vi absolutt dette er relevant som fagområde da man i svært mange tilfeller vil kunne benytte seg av nettopp et CMS ute i den virkelige verden. Grunnet tidligere kompetanse har vi hele veien vært nødt til å tilegne oss ny kunnskap for å løse denne oppgaven. Fra å lære seg det grunnleggende i nevnte skriptspråk, til å gå dypt inn i Joomla!, hvordan det fungerer og da spesielt det å konstruere komponenter. Erfaringen med å hele veien tilegne seg ny kunnskap har vært meget lærerikt, og har gitt oss kompetanse

rundt det å jobbe på denne måten og gjort oss mer effektive skulle man komme borti noe liknende i prosjekter senere i arbeidslivet.

## PRODUKTET

Det har vært morsomt å utvikle en reell løsning som med tiden kunne vært brukt på markedet. Med et produkt som det vi har laget, har det hele tiden vært viktig å fokusere på hvordan dette vil oppleves for potensielle brukere. Vi føler vi har fått på plass en oversiktlig og intuitiv løsning, som ikke bør skremme bort de som er litt skeptiske til eiendomshandel på nett i utgangspunktet.

Styrkene ved løsningen er nettopp et enkelt og oversiktlig design. Hvordan selve auksjonen fungerer, er utformet sammen med oppdragsgiver. Vi som utviklere vil akkurat her fungere som brukere med våre synspunkter, men det har vært nyttig å tilpasse akkurat denne delen med hjelp fra bransjefolk.

Svakhetene i systemet pr. nå er absolutt autentiseringsbehovet og at våres testing naturligvis er gjort uten dette. Som nevnt tidligere må løsningen inneholde en form for autentiseringsløsning for at dette skal kunne brukes i markedet. Siden vi ikke har fått muligheten til å implementere dette i løsningen, har vi heller ingen oppfatning om hvordan dette vil påvirke løsningen vi allerede har. Allikevel er det ingenting ut ifra våre undersøkelser som skulle tilsi at for eksempel BankID ikke skal kunne brukes i samhandling med Joomla!.

Å optimalisere siden for mobilbruk ble ikke prioritert da det gikk med mer tid til annen funksjonalitet enn først ventet. Siden fungerer allikevel tilfredsstillende for en del mobiler og mobilnettlekere, men for bruk i markedet måtte man ha gått nærmere inn på dette punktet. Skal det være mulig å delta i budrunden via mobilen, må det optimaliseres for de fleste mobilene på markedet. Det er allikevel mulig, men ikke optimalt i den nå

## 7. REFERANSER

- [1]. [http://w3techs.com/technologies/history\\_overview/content\\_management/all](http://w3techs.com/technologies/history_overview/content_management/all)
- [2]. [http://w3techs.com/technologies/overview/content\\_management/all](http://w3techs.com/technologies/overview/content_management/all)
- [3]. <http://trends.builtwith.com/cms>
- [4]. <http://www.cmsmatrix.org/matrix/cms-matrix>
- [5]. <http://en.wikipedia.org/wiki/Joomla>
- [6]. <http://en.wikipedia.org/wiki/Joomla>
- [7]. <http://no.wikipedia.org/wiki/Publiseringssystem>
- [8]. [http://en.wikipedia.org/wiki/Content\\_management\\_system](http://en.wikipedia.org/wiki/Content_management_system)
- [9]. <http://no.wikipedia.org/wiki/PHP>
- [10]. <http://no.wikipedia.org/wiki/MySQL>
- [11]. <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- [12]. <http://www.webverksted.no/joomla/komponenter/hva-er-en-komponent.html>
- [13]. <http://demo.joomla.no/bruke-joomla/utvidelser/komponenter.html>
- [14]. <http://www.joomlaeksperten.com/joomla-design/joomla-moduler-og-modulposisjoner.html>
- [15]. <http://www.webverksted.no/joomla/plugins/hva-er-en-plugin.html>
- [16]. [http://docs.joomla.org/Developing\\_a\\_Model-View-Controller\\_\(MVC\)\\_Component\\_for\\_Joomla!2.5](http://docs.joomla.org/Developing_a_Model-View-Controller_(MVC)_Component_for_Joomla!2.5)
- [17]. [http://en.wikipedia.org/wiki/Proxy\\_bid](http://en.wikipedia.org/wiki/Proxy_bid)
- [18]. <http://jquery.com/>

- [19]. <http://jqueryui.com/>
- [20]. <http://www.uploadify.com/>
- [21]. <http://galleria.io/>
- [22]. <http://keith-wood.name/countdown.html>
- [23]. <http://trentrichardson.com/examples/timepicker/>
- [24]. <https://developers.google.com/maps/>
- [25]. <http://www.chronoengine.com/>
- [26]. <https://www.akeebabackup.com/>
- [27]. <https://github.com/elkuku>
- [28]. <http://googlegeodevelopers.blogspot.com/2011/10/introduction-of-usage-limits-to-maps.html>
- [29]. [https://developers.google.com/maps/faq?hl=no-NO#usage\\_pricing](https://developers.google.com/maps/faq?hl=no-NO#usage_pricing)
- [30]. <http://www.sendega.no/>
- [31]. <http://www.pswin.com/produkter/gateway>
- [32]. <https://netcom.no/tjeneste-bedrift/sms-bedrift-api>
- [33]. <http://keysms.no>
- [34]. <http://www.one.com/no/info/profil>
- [35]. <http://extensions.joomla.org/extensions/multimedia/multimedia-display/5095>
- [36]. <http://extensions.joomla.org/extensions/administration/admin-performance/19490>
- [37]. <https://www.bankid.no/BankID-for-ditt-nettsted/>
- [38]. <http://www.buypass.no/brukersted/bli-brukersted>

## VEDLEGG A – FORPROSJEKT

# PROSJEKTPLAN – BOLIGHANDEL1

Thomas Almenningen

Lars Erik Strand

Amund Sørumshagen

## 1. MÅL OG RAMMER

### 1.1 BAKGRUNN

Dagens budgivning på eiendom skjer gjennom skriftlige og muntlige bud, og bærer preg av tradisjonelle og gammeldagse metoder som gir rom for menneskelig påvirkning og feil. Det er i noen grad tatt i bruk elektronisk budgivning, men dette er begrenset til at man kan registrere bud på nettet, og at megler deretter manuelt tar seg av den videre budhåndteringen mot selger og andre budgivere.

Siden utviklingen går mot at flere velger å selge sin eiendom uten å benytte eiendomsmegler (i Norge og utlandet), vil det i fremtiden være et enda større behov for tilpassede og effektive verktøy som på en sikker måte kobler sammen kjøper og selger.

### 1.2 PROSJEKTMÅL

#### **Resultatmål**

- Få et verktøy som effektiviserer/forenkler boligsalgprosessen



- Inkludere selgeren mer i selve salgsprosessen.
- Gjøre det enkelt for selgeren å holde seg oppdatert
- Forbedre førsteinntrykket til eventuelle kunder, gjøre bedriften mer «attraktiv»
- Forenkle bestillingen av bedriftens produkter

### **Effektmål**

- Øke kapasiteten
- Gjøre det mer appellerende for kunder å bruke bolighandel1 til å selge sin bolig
- Mer tilfredse kunder
- Virke innovative ved å ta i bruk egen løsning for salg av bolig over nett

### **Læringsmål**

Store deler av prosjektet vil gå ut på å jobbe med teknologi gruppemedlemmene har minimal erfaring med på forhånd. Regner vi å sitte igjen med mye ny kunnskap etter endt prosjekt.

- Økt kompetanse innen utvikling på web (PHP/JS/MySQL)
- Økt kompetanse innen utvikling vha. CMS-systemer
- Økte kunnskaper om webdesign for ulike plattformer

## **1.3 RAMMER**

Tidsrammen til prosjektet er i underkant av 5 måneder. Siden skal være ferdig utviklet innen 23. mai 2012. Siden dette er et studentprosjekt vil vi ha minimalt til ingen økonomiske hjelpemidler. Vi vil derfor være nødt til å basere oss på å bruke gratis programvare og utviklingsverktøy. For å få laget en fullverdig løsning vil vi være avhengig av å kjøpe inn noe ferdig utviklet programvare. Det vil derfor være våres oppgave å undersøke alternativene/mulighetene som finnes på markedet og veie disse opp mot hverandre. Slik at oppdragsgiver enkelt kan velge det beste alternativet dersom de ønsker å gå videre med prosjektet.

## **2. OMFANG**

## 2.1. OPPGAVEBESKRIVELSE

Oppgaven går ut på å lage en nettside for kjøp og salg av boliger for BoligHandel1.

Pr. dags dato bruker de kombinert finn.no og QXL i tillegg til sine egne sider for sin virksomhet. Målet med denne oppgaven blir å lage en nettside for deres firma som skal inneholde annonse og budgivingssystem. De kommer fortsatt til å bruke finn.no for å nå ut til flere, men budgivning og kundeinteraksjonen skal foregå på firmaets nettside.

---

### VISUELT

Hovedfokuset rent visuelt blir å lage dette så enkelt og effektivt, at den gjennomsnittlige pc-bruker finner det hensiktsmessig å bruke siden.

BoligHandel1 har jo en kundegruppe fra før, så de var også interessert i at gamle kunder kan “gjenkjenne” bedriften på designet.

---

### TEKNISK

Selve løsningen vil inneholde selve nettsiden, en server med databasen og kommunikasjonen mellom disse. Vi må da “oversette” disse tekniske løsningene til enklest mulig form slik at BoligHandel1 ikke trenger noen form for kompetanse for å bruke dette. Enkle grensesnitt for opplasting av bilder og tekst, samt et form for “skjema” for å lage en annonse – både for BoligHandel1, samt brukere av siden.

Det er også ønskelig at siden skal tilpasses flere plattformer, og vi skal i første omgang tilpasse den for mobile enheter. Funksjonaliteten som da skal prioriteres er selve budgivningen, slik at man kan “by i farten”.

---

### VIDERE UTVIKLING

På bakgrunn av de rammene som er satt – at vi ikke får implementert noen form for autentisering med god sikkerhet som for eksempel Buypass eller BankID, vil oppgaven måtte inneholde en utredning for muligheter og alternativer for veien videre skulle BoligHandel1

velge å ta i bruk denne løsningen, eller liknende løsninger, inkludert en form for autentifisering.

## 2.2 AVGRENSNINGER

BoligHandel1 ville gjerne at løsningen skal være fleksibel med tanke på et internasjonalt marked, samt en utredning på hva slags sikkerhets og autentifiseringsløsninger man vil kunne bruke internasjonalt. Vi velger å se bort ifra dette i første omgang, da dette er veldig individuelt fra land til land og vi vil fokusere på løsningen med BoligHandel1 i tankene. Rent teknisk vil det derimot ikke være noen hindringer å implementere løsningen i et internasjonalt marked.

## 3. PROSJEKTORGANISERING

### 3.2 GRUPPEREGLER

#### Gruppereglene for bacheloroppgave

Gruppemedlemmer er Thomas A. Almenningen, Lars Erik Strand og Amund Sørumshagen.

Gruppereglene er generelle retningslinjer for arbeidet med bacheloroppgaven våren 2012.

1. Gruppeleder er Thomas A. Almenningen. Han vil sitte som gruppeleder gjennom hele prosjektforløpet.
2. Gruppeleder tilegnes retten til å signere på vegne av gruppen.
3. Ved gruppeleders fravær vil Lars Erik Strand påta seg rollen som gruppeleder.
4. Vi regner ikke med at kostnader kommer til å påløpe. Dersom dette skulle skje fordeles kostnadene likt på de tre gruppemedlemmene.
5. Ved beslutningsuenighet benyttes votering. Siden vi er tre stk på gruppa fant vi dette passende. Flertallet vil da avgjøre beslutningen. Ved særdeles stor uenighet skal problemet drøftes med veileder.
6. I første omgang vil Mandager, Tirsdager, og Torsdager være satt av til arbeid med oppgaven. Det forventes at gruppemedlemmene møter på skolen minst to av disse dagene. Disse kravene kan etter hvert endres.
7. Logg skal føres av alle gruppemedlemmene de dager det har pågått arbeid med oppgaven.
8. Ved bruk av kilder er det den enkeltes ansvar å oppføre disse.
9. Det skal avholdes ukentlige statusmøter for prosjektet. Dette for å holde alle gruppemedlemmene oppdatert på fremgang, samt vurdere tiltak ved eventuelle etterskudd ift. planen. Møtet avholdes torsdager.

Signatur:

Thomas A. Almenningen



Lars Erik Strand



Amund Sørumshagen



## 4. PLANLEGGING, OPPFØLGING OG RAPPORTERING

### 4.1 HOVEDINDELING AV PROSJEKTET

Ved valg av systemutviklingsmodell har vi lagt stor vekt på hvordan det vil være naturlig å utvikle prosjektet. Målet vil i første omgang være å få på plass et «skall» som et rammeverk. Derfra vil det være enkelt og trinnvis utvikle og legge til ny funksjonalitet til prosjektet. På grunnlag av dette er en modell som bygger på inkrementell utvikling det eneste aktuelle. Oppdragsgiver har verken uttrykt ønsker eller krav til dette valget, så her står vi fritt til å velge det som passer prosjektgruppa best. Ved en inkrementell tilnærming vil vi med jevne mellomrom kunne vise frem nyutviklet funksjonalitet til oppdragsgiver, noe de har stilt seg svært positive til.

Vi valgte derfor en inkrementell modell. Dette gir oss muligheten til å planlegge og utvikle ny funksjonalitet, implementere disse i totalløsningen og presentere dette for oppdragsgiver. Siden inkrementene blir av ulik lengde valgte vi denne modellen fremfor Scrum. Vi vil også underveis være på utkikk etter aktuell kode som kan gjenbrukes.

### 4.2 PLAN FOR STATUSMØTER OG BESLUTNINGSPUNKTER

Se grupperelger pkt. 5 og pkt. 9.

## 5. ORGANISERING AV KVALITETSSIKRING

### 5.1 DOKUMENTASJON OG KILDEKODE

#### **Dokumentasjon:**

Dokumentasjon av valg underveis i prosessen vil være veldig viktig. Hver gang det tas en større beslutning skal dette dokumenteres, lenker og annet materiell som påvirket beslutningen skal noteres ned slik at det skal være lett å finne igjen. Det skal også skrives en kortere tekst som beskriver prosessen med å komme frem til den valgte løsningen. All

dokumentasjon som angår prosjektrapporten skal lastes opp på «google.docs» slik at den lett blir tilgjengelig for alle gruppelemmene.

### Kildekode:

All ikke-triviell kildekode skal kommenteres på en oversiktlig, enkel og god måte etter at modulen/funksjonen er ferdig utviklet. Eventuelt underveis dersom modulen det jobbes med tar flere dager å ferdigstille.

## 5.2 KONFIGURASJONSSTYRING

Vi kommer til å benytte oss av Apache Subversion (SVN) som versjonskontrollverktøy og TortoiseSVN som klient.

## 5.3 RISIKOANALYSE

Som grunnlag for vurderinger av risiko har vi valgt å bruke følgende modell (fig.5.3.1):

| Sannsynlighet/<br>konsekvens | 1<br>Svært lite sannsynlig | 2<br>Lite sannsynlig | 3<br>Sannsynlig | 4<br>Ganske sannsynlig | 5<br>Svært sannsynlig |
|------------------------------|----------------------------|----------------------|-----------------|------------------------|-----------------------|
| 5<br>Katastrofal             | 6                          | 7                    | 8               | 9                      | 10                    |
| 4<br>Svært stor              | 5                          | 6                    | 7               | 8                      | 9                     |
| 3<br>Stor                    | 4                          | 5                    | 6               | 7                      | 8                     |
| 2<br>Middels                 | 3                          | 4                    | 5               | 6                      | 7                     |
| 1<br>Liten                   | 2                          | 3                    | 4               | 5                      | 6                     |




|   |   |
|---|---|
|  | [Grønt område] Akseptabelt  |
|  | [Gult område] Akseptabelt - vurder videre undersøkelser           |
|  | [Rødt område] Ikke akseptabelt - risikoreduserende tiltak påkrevd |

Fig 5.3.1. Matrise for vurdering av risiko. Hentet fra boka «Risikoanalyse: teori og metoder» fra ROSS Gemini Centre ved NTNU.

Ut ifra denne modellen har vi systematisk satt opp de identifiserte risikoene i en tabell (fig.5.3.2):

| Nr. | Beskrivelse  | Sannsynlighet     | Konsekvens   | Total Risiko   | Strategi   |
|-----|--|-------------------|--------------|----------------|--|
| 1   | Tap av data, dokumentasjon, eller kildekode.                               | Sannsynlig        | Katastrofalt | 8<br>(Høy)     | Gode rutiner for backup under hele prosjektforløpet.                               |
| 2   | Interne tidsfrister/deadlines blir ikke overholdt.                         | Svært sannsynlig  | Middels      | 7<br>(Middels) | God planlegging. Gjerne med noe overhead. Arbeide ekstra for å ta igjen det tapte. |
| 3   | Planlegging av arbeidet baseres på mangelfull kunnskap (urealistiske mål). | Sannsynlig        | Stor         | 6<br>(Middels) | Oppdatere plan etterhvert som kunnskap tilegnes.                                   |
| 4   | Kravsendringer underveis i utviklingen.                                    | Ganske sannsynlig | Liten        | 5<br>(Lav)     | Være åpne for endringer, og ikke lage løsninger som er overfølsomme for dette.     |
| 5   | Langvarig fravær grunnet sykdom e.l.                                       | Lite sannsynlig   | Stor         | 5<br>(Lav)     |  |
| 6   | Uklarheter rundt fordeling av ansvarsforhold og arbeidsoppgaver            | Lite sannsynlig   | Middels      | 4<br>(Lav)     |  |

Fig 5.3.2. Tabell med beskrivelse av risikoer.

Risikoene er satt opp i prioritert rekkefølge.

## 6. PLAN FOR GJENNOMFØRING

### Milepæler

- **27/1-12** Prosjektplan ferdig
- **03/2-12** Prosjekthjemmeside ferdigstilt
- **06/2-12** Utviklingsstart
- **17/2-12** Ferdigstilling inkrement 1
- **12/3-12** Ferdigstilling inkrement 2

- **30/3-12** Ferdigstilling inkrement 3
- **20/4-12** Ferdigstilling inkrement 4
- **27/4-12** Ferdigstilling inkrement 5
- **09/5-12** Ferdigstilling nettside/Implementasjonstart
- **23/5-12** Prosjektinnlevering
- **30/5-12** Plakat utstilling leveres i Fronter
- **30/5-12** Publiseringssavtale leveres til biblioteket
- **06/6-12** Presentasjon

#### **Beslutningspunkter:**

- **03/02-12** Konklusjon vurdering av autentisering-/sikkerhetsløsning
- **03/02-12** Konklusjon vurdering av SMS-tjeneste tilbyder
- **03/02-12** Benytte CMS? Eventuelt hvilken løsningen
- **10/02-12** Valg av løsning mobil nettside.

Det vil også bli tatt en rekke beslutninger underveis. Hovedsakelig angående valg av moduler som kan gjenbrukes.

#### **Oppdeling av inkremerter:**

Under er det satt opp en inndeling av hvilken funksjonalitet det skal jobbes med i de forskjellige inkrementene. Det vil mest sannsynlig forekomme en del endringer underveis i prosessen dersom vi ikke rekker å fullføre all funksjonalitet som planlagt, i så fall vil oppdragsgiver bli med på å prioritere hvilke funksjonalitet som er viktigst. Det er også mulig at det vil oppstå nye krav eller ønsker underveis i prosessen.

#### **«Forprosjekt»**

#### **«Oppstartsfasen»**

##### *Opplæring:*

- Javascript/PHP
- Webdesign (HTML/CSS)
- Databaser (MySQL)
- Sikkerhet
- CMS system



*Databaser:*

- Databasedesign
- Prosedyrer, Sikkerhet, Backup prosedyrer

*Andre avgjørelser:*

- Hvilke ferdige moduler finnes/hva kan vi gjenbruke av kode (Delvis Vurdering)
- Benytte CMS?
- Valg av autentiseringsløsning
- Valg av SMS-tjeneste tilbyder
- Valg av karttjeneste
- Krav til hosting, hva med backup?

**«Inkrement 1»**

- Installasjon av CMS, opprette databasen
- Bruker registrering
- Navigasjon/Undersider
- Login og tilhørende funksjonalitet
- Legge inn annonser/annonseforslag og tilhørende funksjonalitet (Steg for steg)

**«Inkrement 2»**

- Budrunde funksjonalitet (minus BANK ID og SMS-tjeneste)
- Selgerside (Informasjon/Statistikk mm.)
- Legge til/fjerne/editere produkter
- Bestilling av produkter
- Mail funksjonalitet
- Mulighet for å opprette nye sider/undersider (Admin)
- Enkel publisering nye artikler og informasjon på siden.

**«Inkrement 3»**

- Søke funksjonalitet (For å finne frem til bolig)
- Implementere karttjenester
- Generell statistikk
- Informasjonsinnsamling (RSS feeds ol.)
- Informasjonsside/Guider på hvordan bruke siden (Interaktivt)

**«Inkrement 4»**

- Implementasjon av sosiale medier

- Annonse verktøy
- Boligblogg
- Bestilling av eksterne tjenester
- Design
- Tilbakemeldings funksjonalitet
- FAQ funksjonalitet (Strukturert)
- Referanse side
- Nyhetsbrev
- Mobilside
- Eventuelt

**«Inkrement 5» (Eventuelt)**

- Implementasjon av Bank ID
- Implementasjon SMS-Tjeneste

**«Ferdigstilling/Levering»**

- Implementasjon
- Testing
- Opplæring

***Gannt Skjema:***



## VEDLEGG B – TIMELOGGER OG MØTEREFERATER

### ARBEIDSLOGG

#### Gruppens arbeidslogg

##### **Forprosjekt (09.01-12 – 27.01-12)**

Forprosjekt, Kunnskapstilegning PHP og JavaScript, Repetisjon

Thomas A: 30 timer, Lars E: 27 timer, Amund: 28 timer

##### **Oppstartsfase (30.01-12 – 03.02-12)**

Valg av CMS, Div utredelser.

Thomas A: 30 timer, Lars E: 31 timer, Amund: 28 timer

##### **Inkrement 1 (06.02-12 – 17.02-12)**

Siden på nett, bli kjent med Joomla!, undersider og menyer på plass, Opplasting av eiendommer

Thomas A: 85 timer, Lars E: 54 timer, Amund: 30 timer

##### **Inkrement 2 (20.02-12 – 12.03-12)**

Opplæring komponentutvikling, skallet til komponenten på plass. Utvikling av komponentsider påbegynt.

Thomas A: 130 timer, Lars E: 40 timer (Ferie), Amund: 45 timer

##### **Inkrement 3 (13.03.12 – 30.03-12)**

Budgivningsfunksjonalitet implementert, Videreutvikling av komponent, Email-funksjonalitet implementert

Thomas A: 110 timer, Lars E: 49 timer, Amund: 39 timer

##### **Inkrement 4 (02.04-12 – 20.04-12)**

[117]

Design, Bildeopplastning, Bildegallerier, Tilgangskontroll, div. bugfixes, ferdigutvikling av komponentsidene, Cron-Jobs, Login.

Thomas A: 100 timer, Lars E: 58 timer, Amund: 35 timer

**Inkrement 5 (23.04-12 – 27.04-12)**

Bugfixes, rapportskrivning, restrukturering av kode.

Thomas A: 30 timer, Lars E: 25 timer, Amund: 25 timer

**Ferdigstilling/Rapportskrivning (30.04-12 – 23.05-12)**

Testing, bugfixes, rapportskrivning, refleksjonsnotat

Thomas A: 90 timer, Lars E: 110 timer, Amund: 50 timer

**Totalt (21.05-12)**

Thomas A: 605 timer, Lars E: 394 timer, Amund:280 timer

## MØTEREFERATER



### Møtereferat

#### Møte med oppdragsgiver

Dato: 16/1-2012

Tidspunkt: 1430-1530

Sted: Kapp Næringshage

Tilstede: Thomas Almenningen, Lars Erik Strand, Amund Sørumshagen, Asgeir Strandlie, Gaute Strandlie

#### Referat:

Hensikten med møte var å informere oppdragsgiver om hvordan vi tenkte å jobbe med prosjektet. Og hva det innebærer å jobbe på en inkrementell måte. Samt å se litt nærmere på hva slags funksjonalitet de ønsket seg.

**Neste Møte:** Ikke avtalt, Skal avholdes etter at 1 inkrement er ferdigstilt (17/2-2012)



**Møtereferat**

**Møte med veileder**

**Dato:** 13/1-2012

**Tidspunkt:** 1100-1120

**Sted:** K-bygget 2. etg

**Tilstede:** Thomas Almenningen, Lars Erik Strand, Amund Sjørumsdalen, Øyvind Kolloen

**Referat:**

På møte ble det snakket løst og ledig om diverse...

**Neste Møte:** Ikke avtalt



## **Møtereferat**

Møte med veileder.

**Dato:** 20/02-2012

**Tidspunkt:** 1400-1430

**Sted:** Høgskolen i Gjøvik

**Tilstede:** Thomas Almenningen, Lars Erik Strand, Amund Sørumsdalen, Øivind Kolloen

### **Referat:**

- Bruk av Joomla – Hvorfor ? Gaute har erfaring -> Godt argument.
- Sette seg inn i Joomla, vil nødvendigvis ta litt tid i begynnelsen.
- Lete etter dokumentasjon på biblioteket
- Statusrapport – 05.03 kl. 10:00 – En side eller to.
- 14 dager før innlevering, utkast Rapport.
- Internasjonalisering – Ingenting å bry seg om. Skal selges til nordmenn uansett.

**Neste Møte:** Ikke avtalt.





## Møtereferat

Møte med oppdragsgiver

**Dato:** 15/3-2012

**Tidspunkt:** 1530-1700

**Sted:** Kapp Næringshage

**Tilstede:** Thomas Almenningen, Lars Erik Strand, Asgeir Strandlie, Gaute Strandlie

### Referat:

Hensikten med møtet var å vise frem løsningen så langt (etter inkrement 2) og få konstruktive tilbakemeldinger samt forbedringspunkter. BoligHandel1 virket svært fornøyd med arbeidet i denne fasen. Følgende forslag til forbedringer ble lagt frem:

- Godkjenning av bruker som publiserer annonse?
- Endre hvilke felter som vises avhengig av hvilken boligtype/eieform som er valgt i skjemaet for innleggelse av annonse.
- Automatisk generering av salgsoppgave (pdf).
- Evt justere minimumsinkrement på pris ved budgivning (avhengig av eiendommens verdi).
- Mulighet til å slette/endre en auksjon (før auksjonsstart).
- Legge til søkefelt for auksjonsstart/slutt.
- Mulighet for å legge inn tilbakemeldinger, evt også automatikk med å legge disse til referanser.

Oppdragsgiver skal også komme med tilbakemeldinger på hvilke andre felter/informasjon som er nødvendig ved salg av eiendom.

**Neste Møte:** Ikke avtalt, blir nok mot slutfasen av prosjektet.



## **Møtereferat**

Møte med veileder.

**Dato:** 03/05-2012

**Tidspunkt:** 1400-1430

**Sted:** Høgskolen i Gjøvik, K-bygget, 2.etg.

**Tilstede:** Thomas Almenningen, Lars Erik Strand, Amund Sørumshagen, Øivind Kolloen

## **Referat:**

Hensikten med møtet var å levere statusrapport nr. 2, samt og avklare noen spørsmål rundt selve oppgaven og spesielt evalueringen av denne. Vi fikk også noen tips til hva vi burde utdype i selve rapporten. Blant disse var for eksempel implementering og kostnader rundt BankID-tjenesten.

Ellers ble det fremdeles snakket løst og ledig om diverse.

**Neste Møte:** Ikke avtalt.



## Møtereferat

Møte med oppdragsgiver

**Dato:** 03/05-2012

**Tidspunkt:** 1630-1730

**Sted:** Kapp Næringshage

**Tilstede:** Thomas Almenningen, Lars Erik Strand, Amund Sørumshagen, Gaute Strandlie

## Referat:

Dette møtet ble holdt da oppdragsgiver hadde noen punkter med forbedringer som de ville diskutere med oss. Disse punktene var:

- Budgivning: ved to like maksbud blir det nyeste tellende (skal være eldste).
- Mulig feil ved innleggelse av nytt maksbud (gjeldene bud oppdateres ikke riktig)?!
- Automatisk email til vinner av en auksjon.
- Budrundevinner: (vise brukernavn, mer detaljert info ved klikk).
- Farger på energimerkinger.
- Tindeannonse kan fjernes?!
- Obligasjoner/aksjer kan fjernes?!
- Kun være mulig å registrere seg når man skal legge ut annonse eller gi bud. (Flytte linken til registreringsskjemaet).
- Registrering før bestilling av produkt?! (Bekreft konto, evt verifisere med personnummer).
- Mulighet for å laste opp finansieringsbekreftelse ved søknad om godkjenning som budgiver.
- Forbrukerrådet? Tips?

**Neste Møte:** Ikke avtalt.

## VEDLEGG C – STATUSRAPPORTER

# STATUSRAPPORT – BOLIGHANDEL1

## SELVE LØSNINGEN

Selve løsningen begynner å ta litt form. Ser man bokstavelig talt på inkrementene, henger vi nok litt etter. Men utkast og “halv-ferdige” forslag til løsninger er oppe og går. Samtidig blir også planen videre klarere og klarere, så hvordan man skal løse funksjonalitet i senere inkremitter står også klarere.

## RAPPORTEN

Vi har ikke begynte å arbeide på den ferdige rapporten. Men hver enkelt har skrevet utkast til dokumentasjonen rundt den funksjonaliteten man har jobbet med. Slikt sett skrives dokumentasjon underveis, så det finnes allerede en del som skal inn i rapporten.

## DEN ENKELTE

### **Thomas:**

Er den som har gjort mest av arbeidet på selve løsningen. Thomas er også dypere inn i Joomla enn de to andre, naturlig nok på grunn av at han kunne mye rundt WWW Tek. Ifra før. I løsningen vår er vi også nødt til å lage egne komponenter etter hvert, så Thomas har også brukt mye tid på å lære seg dette.

### **Lars Erik:**

Lars Erik har ikke hatt WWW Tek før, og mye av tiden har gått med til å blåse gjennom pensumet der for å komme ajour så fort som mulig. Han har gjort noen enkle ting på siden, men arbeidet med å sette sett ordentlig inn i den avanserte delen av Joomla begynner nå.

### **Amund:**

Samme utgangspunkt som Lars Erik. Mye av tiden har gått med til å blåse gjennom WWW Tek, og prøve å lære seg det så godt som mulig for å kunne bidra mer på selve løsningen. Har også gjort noen småting på selve løsningen, men vil nå også begynne å sette seg tyngre inn i Joomla.

## OPPDRAGSGIVER

Gutta i BoligHandel1 er informert om at selve løsningen ikke har vært 1.pri så langt, på grunn av tiden som er brukt til å faktisk tilegne seg den kunnskapen som er nødvendig. Men vi har hatt kommunikasjon, og har avtalt møte om en tidagers tid, hvor vi skal gå gjennom siden så langt, og justere ting etter deres ønsker.

# STATUSRAPPORT 2 – BOLIGHANDEL1

Thomas Almenningen

Lars Erik Strand

Amund Sørumshagen

## Inkrementene

Ifølge den oppsatte prosjektplanen skulle vi nå vært ferdige med selve løsningen, og i gang med testfasen/bugfixing. Vi kan ikke si at vi har greid dette. Blant punktene i prosjektplanen er det noen vi ikke har rukket å utføre, både på grunn av tid og prioriteringer.

Inkrement 2:

- Selgerside (Informasjon/Statistikk mm.) – Dette er gjort, men det hadde vært ønskelig med mer funksjonalitet her (statistikk over «hits» på auksjoner ol.).

Inkrement 3:

- Informasjonsinnsamling (RSS feeds)

Inkrement 4:

- Bestilling av eksterne tjenester.
- Tilbakemeldingsfunksjonalitet.
- FAQ funksjonalitet.
- De to ovennevnte er funksjonalitet vi anser som lette og implementere, men likevel tidkrevende. Disse har derfor blitt nedprioritert.
- Mobilside!

Kommentar til testfasen: mye av dette har blitt gjort underveis. Bugs har blitt fikset da de har blitt oppdaget. Implementasjonen av det ferdigstilte produktet hos Bolighandel1 er usikkert hvorvidt skal gjøres. Løsningen vil jo ikke være fullverdig uten en autentiseringsteknologi (BankID).

## RAPPORTEN

Rapportskrivingen er i gang, men vi har enda mye jobb å gjøre her. Skrivningen til nå har basert seg på hva de enkelte gruppemedlemmene har gjort, samt valg av teknologier ol. De mer generelle punktene i rapporten (forord, innledning, kravspesifikasjon osv.) er ikke påbegynt. Neste uke vil dermed være prioritert rapportskriving og vi håper å få ferdigstilt et forslag til sluttrapport i slutten av uka (11.05.12).

## VEDLEGG D – PROSJEKTAVTALE



HØGSKOLEN I GJØVIK

## PROSJEKTAVTALE

mellom Høgskolen i Gjøvik (HiG) (utdanningsinstitusjon),

Bolig Handel 7  
 \_\_\_\_\_ (oppdragsgiver), og

\_\_\_\_\_ (student(er))

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

1. Studenten(e) skal gjennomføre prosjektet i perioden fra \_\_\_\_\_ til \_\_\_\_\_.

Studentene skal i denne perioden følge en oppsatt fremdriftsplan der HiG yter veiledning.

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

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

Besvarelsen med bilder og tegninger, modeller og apparatur så vel som programkoding, kildetekst, data base, taper mv. som inngår som del av eller vedlegg på bibliotekets del av skolens internett-sider. Dette avhenger av at studentene skreiver under på en egen avtale hvor de gir biblioteket tillatelse til at deres hovedprosjekt blir gjort tilgjengelig for bruk av de samme sidene. Oppdragsgiver og partene gis det et

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

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

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

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

12. Deltakende personer ved prosjektgjennomføringen:

HiGs veileder (navn): \_\_\_\_\_

Oppdragsgivers  
kontaktperson (navn):

*Asgeir Strandlie, Grete Strandlie*

Student(er) (signatur): \_\_\_\_\_

dato \_\_\_\_\_

dato \_\_\_\_\_

dato \_\_\_\_\_

dato \_\_\_\_\_

Oppdragsgiver (signatur):

*Belig Handell v/ Grete Strandlie*

dato *26.1.2012*

IMT Dekan/prodekan (signatur): \_\_\_\_\_

dato \_\_\_\_\_