

---

# EASYDEV

---

---

## GRUPPE TUXMIN

---

Espen Zaal s198599  
Lukas Larsed s198569  
Petter Knagenhjelm Lysne s198579

17. november 2014



### Sammendrag

**H**er skal vi skrive et sammendrag over rapporten. Som sist kan vi bruke den coole lettrine pakken til å få en heftig stor bokstav i begynnelsen på avsnittet.

# Innhold

<b>1</b>	<b>Introduksjon</b>	<b>5</b>
1.1	Om rapporten . . . . .	5
1.2	Formål . . . . .	5
1.3	Tolkning av oppgaven . . . . .	6
1.4	Mål . . . . .	6
1.5	Tekniske detaljer . . . . .	6
1.5.1	Utviklingsmiljø . . . . .	6
1.5.2	Krav til programvare . . . . .	6
1.5.3	Versjonshåndtering . . . . .	7
<b>2</b>	<b>Prosessdokumentasjon</b>	<b>8</b>
<b>3</b>	<b>Produktdokumentasjon</b>	<b>9</b>
<b>A</b>	<b>Diagram</b>	<b>10</b>
<b>B</b>	<b>Fremtaging av GUI</b>	<b>16</b>
B.1	Revisjon 1 . . . . .	16
B.2	Revisjon 2 . . . . .	20

# Eksempler

# Figurer

A.1	UML . . . . .	11
A.2	Brukercase . . . . .	12
A.3	MVC - første utkast . . . . .	13
A.4	Kontroller og GUI . . . . .	14
A.5	Tabellmodell og output . . . . .	15
B.1	Fremtaging: Velkomstbilde . . . . .	16
B.2	Fremtaging: Boligsøk . . . . .	17
B.3	Fremtaging: Meglersøk . . . . .	17
B.4	Fremtaging: Megler GUI . . . . .	18
B.5	Fremtaging: Boligregistrering . . . . .	18
B.6	Fremtaging: Saksbehandling . . . . .	19
B.7	Fremtaging 2: Søkervindu . . . . .	20
B.8	Fremtaging 2: Meglervindu . . . . .	21

**Tabeller**

# Kapittel 1

## Introduksjon

### 1.1 Om rapporten

Denne rapporten består av flere kapitler som kan leses hver for seg og som har hvert sine formål.

**Introduksjonen** vil gå gjennom litt av forutsetningene for oppgaven, målene vi har satt oss, tolkningen av oppgaven og valgene vi har tatt på bakgrunn av det.

**Prosessdokumentasjonen** vil beskrive aspektet ved arbeidet vårt. Hvordan vi kom sammen som en gruppe, bestemte oss for fremgangsmåte og utfordringene vi har stått overfor underveis.

**Produktdokumentasjonen** er av det veldig tekniske slaget. Det er gitt mange illustrasjoner og kodeeksempler på utvalgte metoder og funksjonalitet, slik at det skal være overkommelig for utenforstående å sette seg inn i programmet.

**Testrapporten** vil beskrive de tester vi har utført, hvordan vi har utført dem og hvilke resultater de gav.

**Brukerdokumentasjonen** vil både være inkludert i dette dokumentet, samt som et frittstående dokument. Den dokumentasjonen vil gi brukeren oversikt over hvordan en bruker programmet og hvilke muligheter programmet gir.

### 1.2 Formål

Lage et datasystem som kan håndtere boligformidling i utleiemarkedet. Utleiere skal få sine tilbud presentert og ha mulighet til å nå potensielle leietakere. Boligsøkende skal kunne registrere sine opplysninger og sende inn ønsket bolig som finnes tilgjengelig for utleie. Firmaet skal registrere alle leieforhold som opprettes slik at boligen og leietakeren ikke lenger vil være registrert som henholdsvis ledig boligsøkende.

## 1.3 Tolkning av oppgaven

## 1.4 Mål

Følgende mål ble satt opp ved begynnelsen av arbeidet med oppgaven:

**Robusthet** Det var viktig at vår kode skulle være så generell så mulig slik at vi enkelt kan innføre eventuelle tillegg eller endringer i programmet med den hensikt at kodebasen blir enkel å vedlikeholde. Koden skal derfor bygges opp med hjelp av arv og polyformisme i så stor grad som mulig. Det skal være lite bruk av parametere eller identifikatorer som låser komponenter annet enn i konstanter som skal være låst.

**MVC** Programmet skal være bygget opp etter MVC<sup>1</sup>-arkitektur slik at logikk og brukergrensesnitt er helt avskjermet fra hverandre og all informasjonsutveksling blir håndtert via en kontroller. Det skal være så mye gjenbruk av kode som mulig. De metoder og variabler som kan være `static` skal være det.

**Intuitivt brukergrensesnitt** Brukergrensesnittet skal være enkelt og oversiktlig slik at en bruker som ikke er kjent med programmet kan foreta boligsøk og sende forespørsel til meglerfirmaet. En ny megler skal rask starte opp i sin modul og på kort tid kunne bli kjent med programmets funksjonalitet.

**Faglig utfordring** Det var et mål at vi strakk oss langt i forhold til å komme opp med løsninger som ikke bare løser oppgaven i henhold til pensum, men på en måte som er mest mulig slik vi tror at man ville gjort det i næringslivet. Det vil si å ikke ta snarveier, velge `JTable` foran `JList`, bruke MVC, osv.

## 1.5 Tekniske detaljer

### 1.5.1 Utviklingsmiljø

Prosjektet er utviklet i Eclipse IDE<sup>2</sup>. Ikoner og annen grafikk er opprettet eller editert i Gimp<sup>3</sup>. Det er brukt en bootstrap FlatUI *her skal vi legge til link til flat ui og beskrive denne kort*. Generelle ikoner (Open source) er hentet fra flaticons.net. Innledende struktur over klasser ble opprettet som UML diagram med ArgoUML. Hele prosjektet er lagd i tegnoppsett UTF-8 og det er ikke brukt noen norske bokstaver i kode eller kommentarer.

### 1.5.2 Krav til programvare

Eventuell beskrivelse om hvilken webserver som programvaren må kjøre på eller hvilken software som denne er testet på.

---

<sup>1</sup>eng. Model View Controller

<sup>2</sup>eng. Integrated Development Environment

<sup>3</sup>The GNU Image Manipulation Program



### 1.5.3 Versjonshåndtering

Til versjonhåndtering brukte vi GIT via terminal og innebygd støtte i utviklingmiljøer (IDE). Lagring av prosjektet ble gjennomført sentralt via en repository på github. Repository for gruppen er privat frem til innlevering av prosjektoppgaven og kommer til å gjøres tilgjengelig for publikum etter at deadline for prosjektet har utløpt. Kildekoden og prosjektets historikk vil da være tilgjengelig fra følgende linker:

**Kildekode**

<https://github.com/CervecerosCodigo/EasyDev>

Lagret som JavaScript prosjekt.

**Rapport**

<https://github.com/CervecerosCodigo/EasyDevRapport>

L<sup>A</sup>T<sub>E</sub>X kode

## Kapittel 2

# Prosessdokumentasjon

Dette kapitlet beskriver kort om prosessen fra prosjektet startet og tiden frem mot innlevering. Det er ikke brukt mye tid på hvem som har gjort hva, og til hvilken tid. Vi har heller delt det opp slik at vi skriver om hvorfor vi har gjort som vi har gjort og hvilken utfordringer vi støtte på underveis.

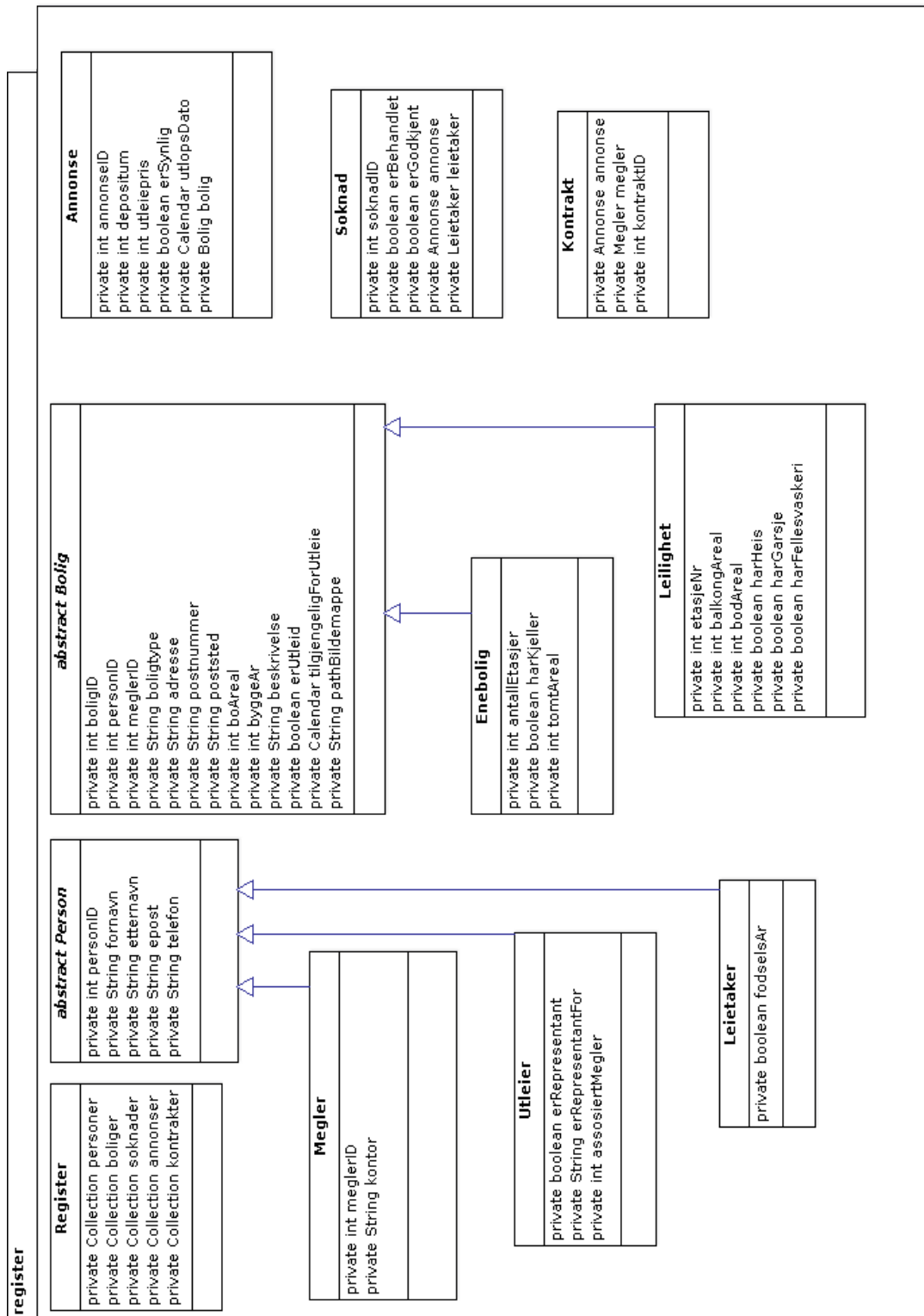
Både prosessdokumentasjonen og produktdokumentasjonen er full av referanser til de kapitlene og seksjonene der man kan lese mer om de forskjellige løsningene, og kodeksempler på hvordan det er løst.

## Kapittel 3

# Produktdokumentasjon

Tillegg A

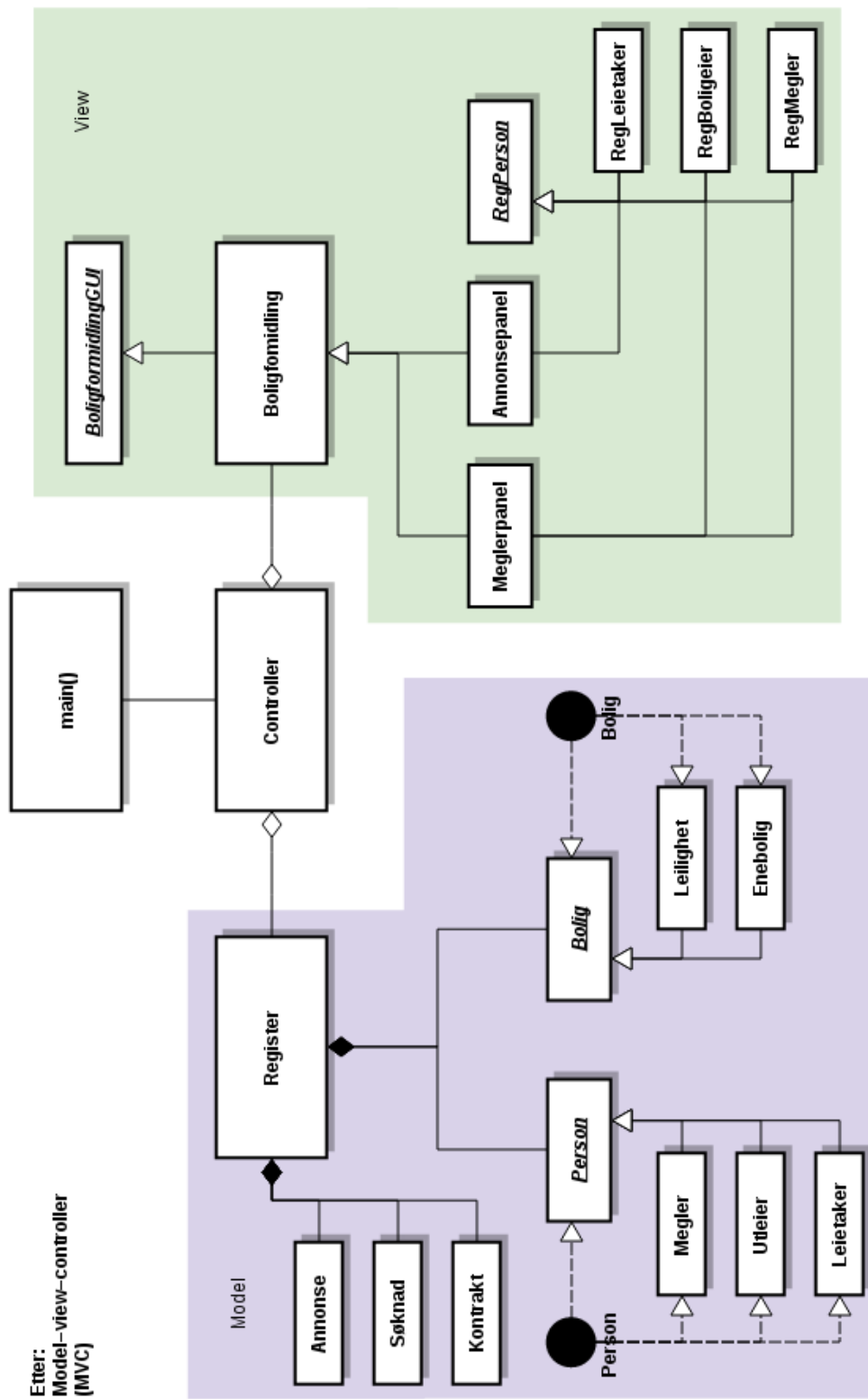
Diagram



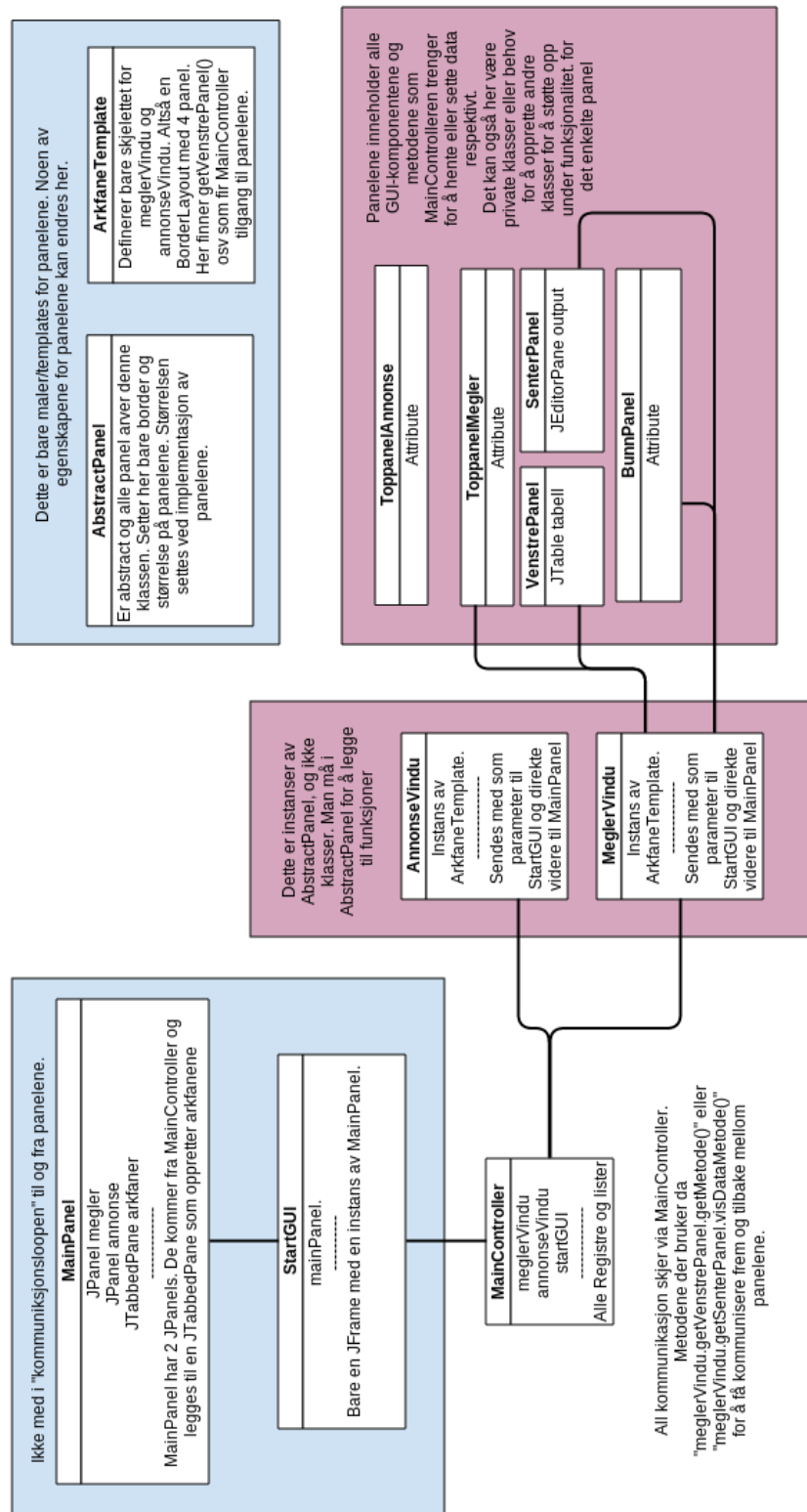
Figur A.1: Innledende UML diagram, brukt for generering av grunnleggende klasser.



Figur A.2: Flyt diagram over mulig user case som kan foretas i brukergrensesnittet.

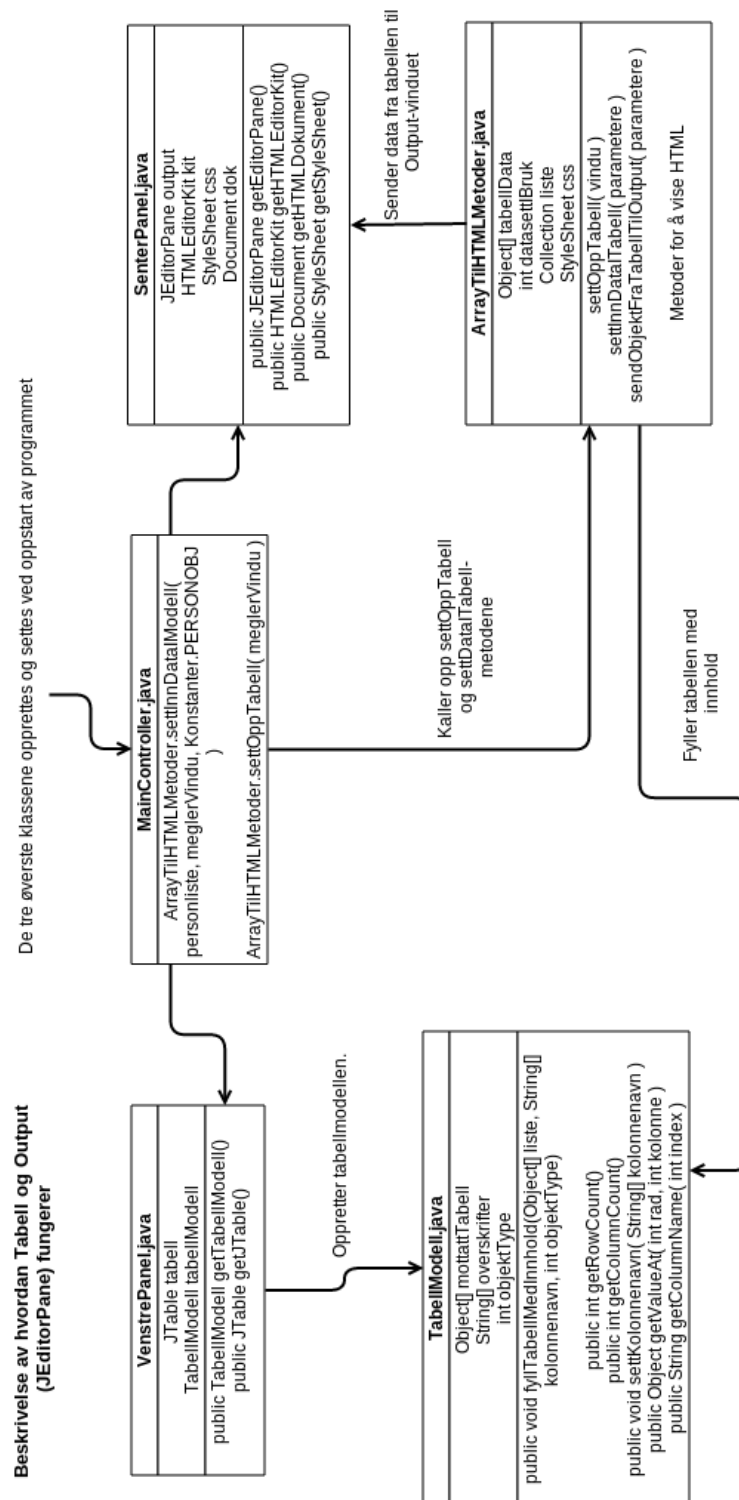


Figur A.3: Innledende diagram over planlagt MVC arkitektur i programmet.



Figur A.4: Flytdiagram over kontroller og GUI klasser.



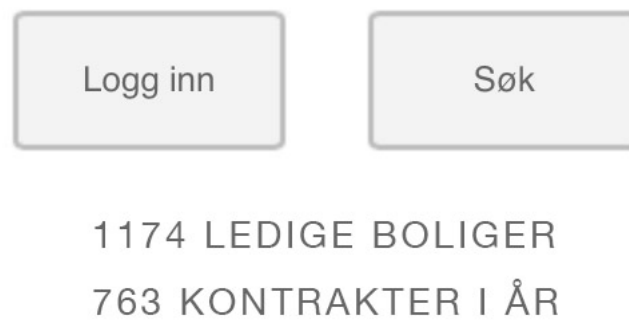


Figur A.5: Flytdiagram over tabellmodell og output.

## Tillegg B

# Fremtaging av GUI

### B.1 Revisjon 1



Figur B.1: Fremtaging: Velkomstbilde

The GUI for 'Boligsøk' features a search interface at the top. It includes three dropdown menus: 'Antall rom' (Number of rooms), 'Boligtype' (Property type), and 'Fylke:' (County). Below these are three input fields: 'Areal min:' (Minimum area), 'Areal max:' (Maximum area), and 'Max utleiepris:' (Maximum rental price). A 'Finn boliger' (Find houses) button is on the left, and a 'Søk valgt bolig' (Search selected house) button is on the right. The main area is a large, empty rectangular box for displaying search results.

Figur B.2: Fremtaging: Boligsøk

The GUI for 'Meglersøk' features a search interface at the top with five input fields: 'Fornavn' (First name), 'Etternavn' (Last name), 'FødselsNr/dato' (Birth number/date), 'Adresse' (Address), and 'KontraktNr' (Contract number). Below these fields are six buttons: 'Finn Bolig' (Find house), 'Finn Person' (Find person), 'Finn Kontrakt' (Find contract), 'Slett Bolig' (Delete house), 'Slett Person' (Delete person), and 'Slett Kontrakt' (Delete contract). The main area is a large, empty rectangular box for displaying search results.

Figur B.3: Fremtaging: Meglersøk

The screenshot shows a web application interface with three tabs: "Registrer", "Søk", and "Saksbehandling". The "Registrer" tab is active. The form contains the following elements:

- Fields for "Fornavn", "Epost", "PostNr", "Byggeår", "Etternavn", "TelefonNr", "Poststed", "Utleiepris (mnd)", "PersonNr", "Adresse", "Areal", and "Antall rom".
- A checkbox labeled "Hus/rekkehus" which is checked.
- A dropdown menu for "Etasjer".
- A text input for "Tomtstørrelse".
- A checkbox labeled "Kjeller".
- A section titled "Andre opplysninger" with a large text area and a date picker (mm / dd / yyyy).
- A button labeled "Registrer".
- A checkbox labeled "Leilighet".
- A dropdown menu for "Etasje".
- Two checkboxes labeled "Heis" and "Balkong".

Figur B.4: Fremtaging: Megler GUI

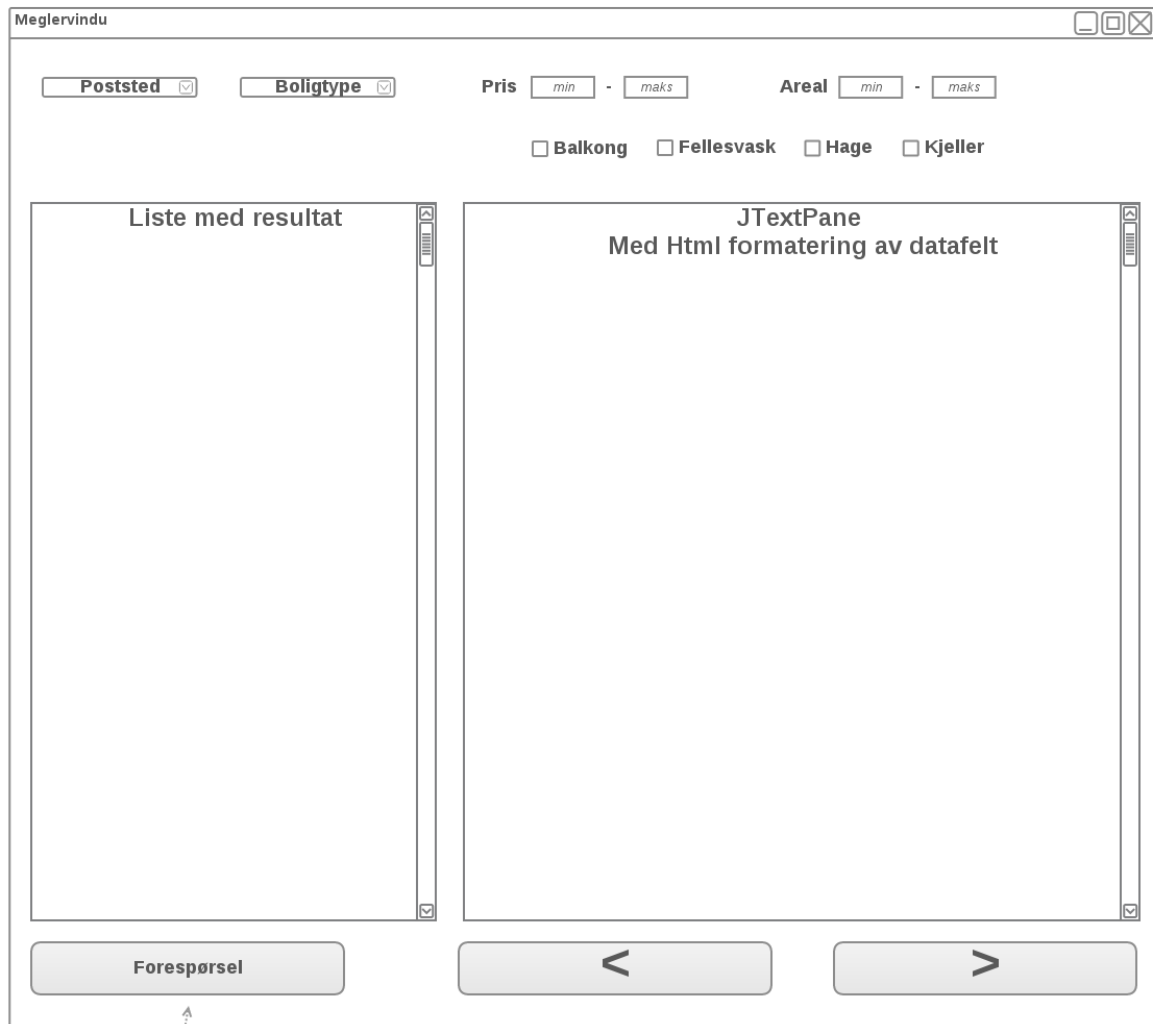
This screenshot shows a similar web application interface to Figure B.4, but for "Boligregistrering". The layout and fields are identical, including the "Registrer" tab, input fields for personal and property details, checkboxes for "Hus/rekkehus" (checked) and "Leilighet", dropdown menus for "Etasjer" and "Etasje", a "Tomtstørrelse" field, a "Kjeller" checkbox, a date picker, and a "Registrer" button.

Figur B.5: Fremtaging: Boligregistrering



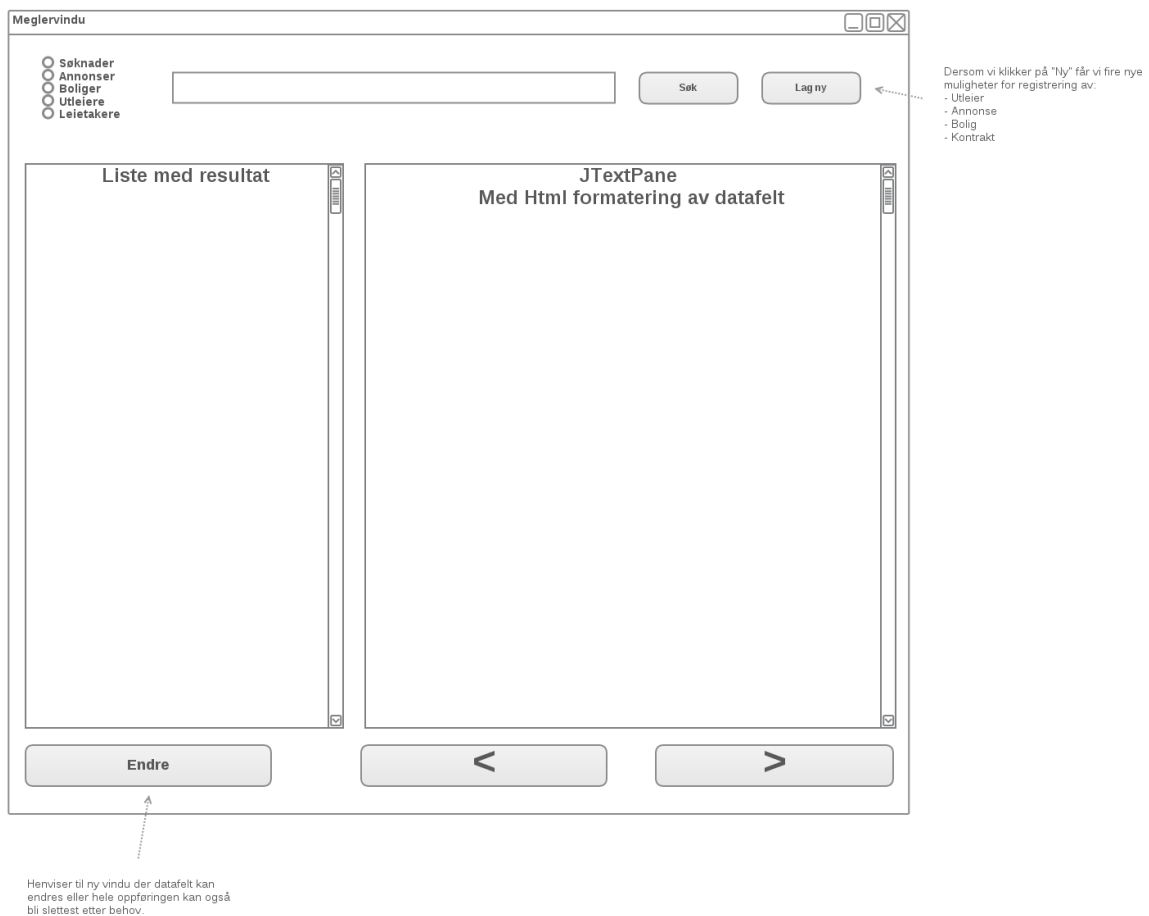
Figur B.6: Fremtaging: Saksbehandling

## B.2 Revisjon 2



Henviser til ny vindu der datafelt kan endres eller hele oppføringen kan også bli slettet etter behov.

Figur B.7: Fremtaging 2: Søkervindu



Figur B.8: Fremtaging 2: Meglervindu