



# CoClean

DETALJERT SYSTEMDESIGN

BERZI NAWZAD WASFY

I dette dokumentet vil jeg legge fram min plan for utviklingen av CoClean. Appen har som hovedmål å forenkle fordeling av arbeidsoppgaver i kollektiv og bofellesskap.

## Resurser

CoClean skal være ren og pen, jeg har derfor valgt å unngå å bruke for mange bilder og andre lignende ressurser.

## Ikoner

Det meste av ikoner hentes fra material font, men jeg må selv produsere logo for appen.

## Bilder

De eneste bildene utenom ikoner er profilbildene, dette hentes via deres sosialemedier(om de kobler de sammen) eller så kan de laste opp bilde selv.

## Lyd

Appen er lydløs utenom notification, og for dette bruker jeg Android sin egen notification lyd.

## Firebase

Firebase er en platform som kan potensielt gjøre arbeidet mitt mye enklere, den kan ta seg av verifisering, database og brukshistorikk. Med Firebase vil jeg lett kunne implementere databasen og «beskjedsystemet» uten måtte bygge noe infrastruktur. Jeg kan ved senere anledning også utvikle appen til å støtte både iOS og Windows i tillegg til Android.

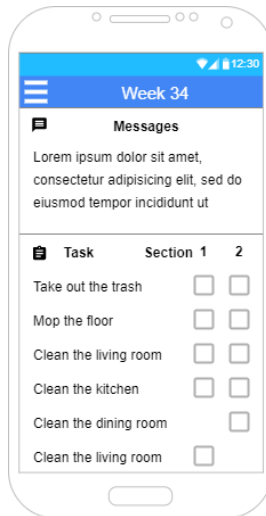
## Problemer som kan oppstå

Jeg har mange ideer, de fleste har jeg tatt med her, men jeg kan tenke meg at noen ting kan bli komplisert, spesielt forholdene mellom klassene mine, hva som burde håndteres clientside og hva som burde som burde håndteres serverside.

Jeg kommer trolig til å bruke mye tid på selve design: flyten og interaksjoner, bakenddelen tenker jeg blir relativt enkelt da appen ikke er så komplisert.

## Design

### Hovedsider

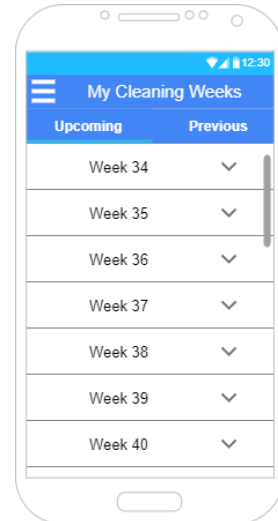


#### This Week - forsiden

Denne siden blir landingpage etter innlogging, og herfra finner man det man trenger, her kan de se hva slags arbeidsoppgaver bruker må gjøre denne uken, her kan man også se på nyeste beskjedene som er lagt ut.

#### My Cleaning Weeks

På denne siden kan brukeren gå gjennom både kommende og tidligere uker, her kan brukeren finne ut hvilke arbeidsoppgaver han har framover, samt at han kan sende forespørsel om å bytte arbeidsoppgave/vaskeuke med andre som bor i kollektivet.

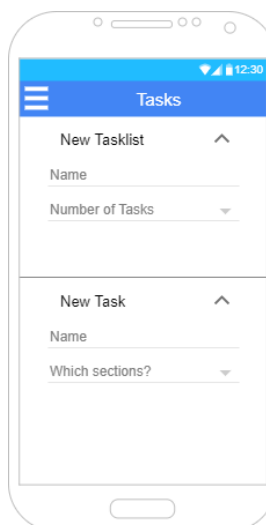
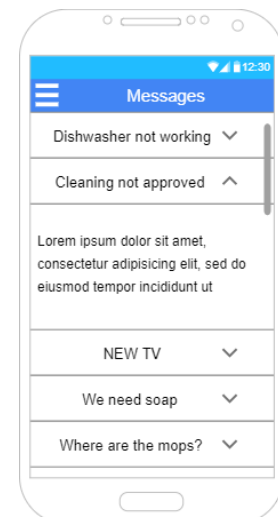


#### Apartment

Her kan brukeren se hvem som bor i samme kollektiv som bruker, samt romnr, herfra kan man også opprette nye rom, sette rom som tom og man kan også finne info som bursdagsdato.

#### Messages

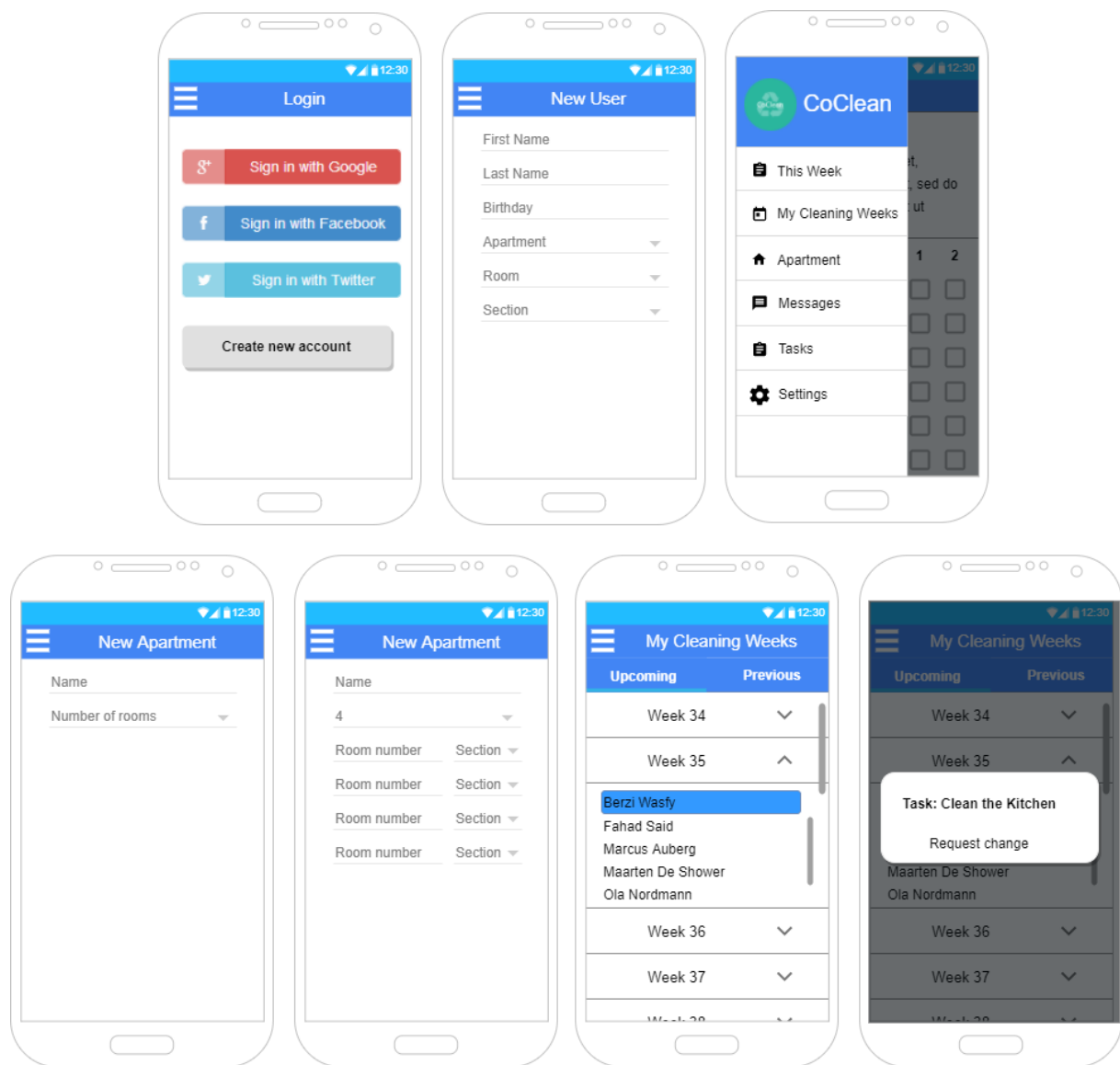
En samleside for alle beskjedene som har blitt lagt ut, herfra kan bruker holde seg oppdatert om ting som skjer i kollektivet.



#### Tasks

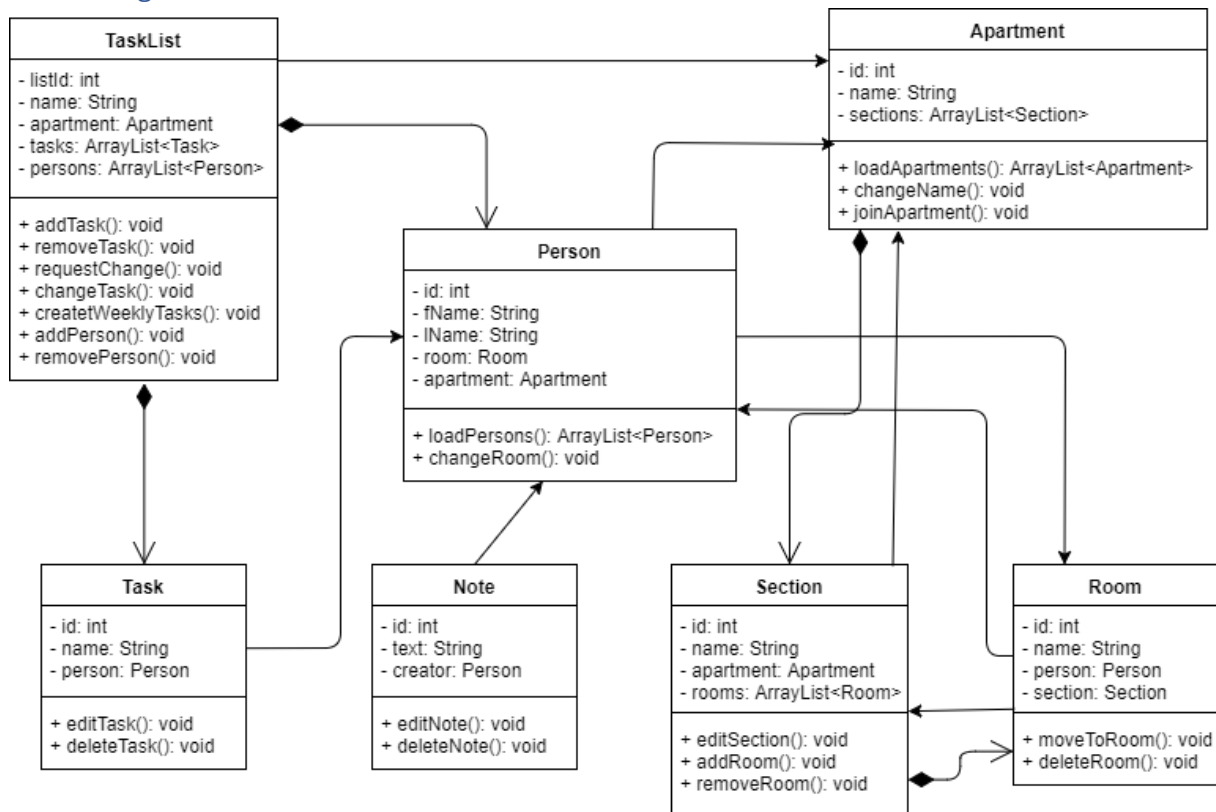
Herfra vil brukern kunne opprette ny TaskList og nye Task til kollektivet.

## Sekundæresider



## Klassedokumentasjon

### Klassediagram

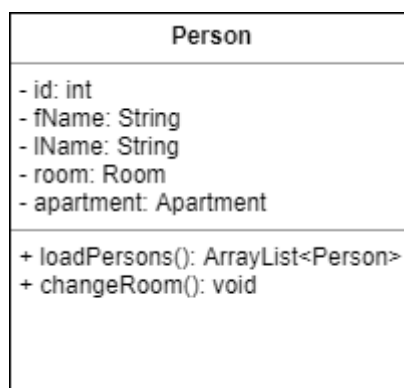


### Person

**Navn:** public class Person

**Hensikt:** Hensikten med denne klassen er å holde på viktig informasjon om brukerne.

**Variabler og metoder:**



**Konstruktør:**

```

public Person(String fName, String lName){
    this.fName = fName;
    this.lName = lName;
}

```

**Operasjoner:**

*public ArrayList<Person> loadPersons()*  
 returnerer en ArrayList med alle personer for det kollektivet sender tilhører.

*public void changeRoom()*  
 Metode for å bytte rom eller kollektiv.

## TaskList

**Navn:** public class TaskList

**Hensikt:** Styrer alle oppgaver for kollektivet.

### Variabler og metoder:

TaskList
- listId: int - name: String - apartment: Apartment - tasks: ArrayList<Task> - persons: ArrayList<Person>
+ addTask(): void + removeTask(): void + requestChange(): void + changeTask(): void + createWeeklyTasks(): void + addPerson(): void + removePerson(): void

### Konstruktør:

```
public TaskList(String name, Apartment apartment,
ArrayList<String> tasks){
    this.name = name;
    this.apartment = apartment;
    this.tasks = tasks;
}
```

### Operasjoner:

```
public void addTask()
Legger til en Task i tasks
```

```
public void removeTask()
Fjerner en Task fra tasks
```

```
public void requestChange()
Sender forespørsel om å bytte vaskeuke/arbeidsoppgave.
```

```
public void changeTask()
Endrer vaskeuke/arbeidsoppgave.
```

```
public void CreateWeeklyTasks()
Fordeler arbeidsoppgaver for alle medlemmer av kollektivet.
```

```
public void addPerson()
Metode for å legge til Person i persons.
```

```
public void removePerson ()
Metode for å fjerne Person fra persons.
```

## Task

**Navn:** public class Task

**Hensikt:** Oppretter og styrer arbeidsoppgaver.

**Variabler og metoder:**

Task
- id: int - name: String - person: Person
+ editTask(): void + deleteTask(): void

**Konstruktør:**

```
public Task(String name, TaskList taskList){  
    this.name = name;  
}
```

**Operasjoner:**

```
public void editTask()
```

Metode for å endre arbeidsoppgave.

```
public void deleteTask()
```

Metode for å slette arbeidsoppgave.

## Note

**Navn:** public class Note

**Hensikt:** Oppretter og styrer meldinger/notater som blir vist på forsiden.

**Variabler og metoder:**

Note
- id: int - text: String - creator: Person
+ editNote(): void + deleteNote(): void

**Konstruktør:**

```
public Note(String text, Person creator){  
    this.text = text;  
    this.Person = Person;  
}
```

**Operasjoner:**

```
public void editNote()
```

Metode for å endre meldinger.

```
public void deleteNote()
```

Metode for å slette meldinger.

## Apartment

**Navn:** public class Apartment

**Hensikt:** Oppretter og styrer alle kollektiv/bofellesskap

**Variabler og metoder:**

Apartment
- id: int - name: String - sections: ArrayList<Section>
+ loadApartments(): ArrayList<Apartment> + changeName(): void + joinApartment(): void

**Konstruktør:**

```
public Apartment(String name){  
    this.name= name;  
}
```

**Operasjoner:**

```
public ArrayList<Apartment> loadApartments()
```

Oppretter en ArrayList med alle instanser av Apartment.

```
public void changeName()
```

Metode for bytte navn på en Apartment.

```
public void joinApartment()
```

Metode for bli med i et kollektiv

## Section

**Navn:** public class Section

**Hensikt:** Oppretter og styrer seksjoner

**Variabler og metoder:**

Section
- id: int - name: String - apartment: Apartment - rooms: ArrayList<Room>
+ editSection(): void + addRoom(): void + removeRoom(): void

**Konstruktør:**

```
public Section(String name, ArrayList<String> rooms){  
    this.name= name;  
    foreach(String in rooms){  
        Room rm = new Room(name)  
        this.rooms.add(rm);  
    }  
}
```

**Operasjoner:**

```
public void editSection()
```

Metode for redigere instanser av Section

```
public void addRoom()
```

Metode for å legge til rom i en Section.

```
public void removeRoom()
```

Metode for å fjerne rom i en Section.



## Room

**Navn:** public class Room

**Hensikt:** Oppretter og styrer rom

**Variabler og metoder:**

Room
- id: int - name: String - person: Person
+ moveToRoom(): void + deleteRoom(): void

**Konstruktør:**

```
public Room(String name){  
    this.name= name;  
}
```

**Operasjoner:**

```
public void moveToRoom()
```

Metode for tilegne Person til Room

```
public void deleteRoom()
```

Metode for slette rom.