

INSTITUTT FOR DATATEKNOLOGI OG  
INFORMATIKK

PROSJEKT 4

IT2810 WEBUTVIKLING

---

**ARKITEKTUR**

---

*Forfattere*

Pål Christian IVERSEN

Martin L. KARLSTRØM

Thomas MARKUSSEN

Fride SKARSETH

Ole Anders STOKKER

8. november 2017

# 1 Applikasjonen

Applikasjonen er en film- og seriemanager. Applikasjonen skal ha funksjonalitet til se nyheter om nye, aktuelle filmer og serier. Dette er implementert som en «*Discovery*»-side, og ligger som hovedsiden av applikasjonen. Om man har sett en film eller serie kan man legge til disse i et eget «*Library*», der man har oversikt over alle filmer og serier brukeren har sett, figur 1.

Applikasjonen skal også ha funksjonalitet til å søke på alle filmer, serier og serieepisoder i «*Search*». Brukeren skal ha mulighet til å legge til disse i en egen «*Watchlist*», der man har en oversikt over filmer og/eller serier man har lyst til å se i fremtiden. I teorien vil *Library* og *Watchlist* ha lik logikk, men ulikt innhold, se figur 2.

I «*Search*» skal brukeren kunne sortere søket etter forskjellige parametere som navn, dato og rating, og deretter filtrere søket etter filmer og serier/episoder, figur 3. *Library* og *Watchlist* skal også ha funksjonalitet til å søkes i og filtreres.

# 2 Komponenter

## Discover

Gir brukeren en oversikt over aktuelle og nyutgitte filmer/serier. Hvis brukeren er logget inn kan en legge til filmen/serien direkte i «*Library*» eller «*Watchlist*».

## Card

Gir en visuell visning av en gitt film eller serie. Komponentten mottar data som skal visualiseres fra enten «*Search*», «*Library*» eller «*Watchlist*».

## Library

Gir brukeren en oversikt over alle filmer eller serier han/hun har merket som sett. Her har brukeren også mulighet til å sortere og søke innad i sitt bibliotek, endre visning av elementer, samt mulighet for sletting fra eget bibliotek.

## Watchlist

Gir brukeren oversikt over alle filmer eller serier han/hun har lagt til i *Watchlist*. Dette er filmer/serier brukeren har planer om å se i fremtiden. Her har brukeren også mulighet til å sortere og søke innad i sitt bibliotek, endre visning av elementer, velge en film/serie som sett, samt slette elementer fra sin liste. Velges filmen som sett, vil den bli flyttet over til «*Library*».

## Search

Search er alltid tilgjengelig øverst i navbaren. Search gir brukeren muligheten til å søke opp filmer eller serier, som så kan legges til i «*Library*» og «*Watchlist*».

#### User

User inneholder profilen til brukeren. Her har brukeren mulighet til endre kontoinnstillinger.

## 3 Moduler

### 3.1 Frontend

Frontenden er skrevet i *Angular 4*. Denne applikasjonen er kompilert/transpilert før den serves via backenden. Angular applikasjonen gjør all sin kommunikasjon for å hente data fra vårt eget og eksterne API-er gjennom backenden vår via http-requests.

#### 3.1.1 UI-Kit 3

Ved valg av UI-kit ble det bestemt å bruke et lightweight, modulært UI-bibliotek kalt *UI-Kit 3*. Dette er for å gi siden et pent visuelt preg som er lett å implementere, samt for å gjøre siden brukervennlig for brukeren uten å påvirke hastigheten.

### 3.2 Backend

Det brukes *NodeJS* med *Express*, for å håndtere data og HTTP-requests. All data som brukes av applikasjonen går først gjennom backend, der den blir sortert og filtrert, figur 4. Dette er for å håndtere all dataflyt sentralt slik at autorisering blir kontrollert riktig.

#### 3.2.1 Express

For å håndtere routing og HTTP requests brukes rammeverket *Express*. Dette rammeverket er modulært, og kan konfigureres på forskjellige måter. Applikasjonen er delt opp i flere filer, og har et eget system som gjør det enkelt å legge til utvidelser, der for eksempel API-er er enkle og forståelige. *Express* gir mulighet til å dele opp routingen på en hierarkisk måte. Dette gjør det lett å lage egen middleware, til for eksempel å sjekke at brukeren er logget inn og hva de er autorisert til å gjøre. Backend holder styr på de absolutte adgangene i applikasjonen.

#### 3.2.2 Database

For å lagre bruker og annen data, brukes *Mongodb* med *mongoose ODM*-en. For brukerautentisering brukes *Google+* sitt api gjennom *Oauth2*. Dette lar brukere enkelt logge inn med en eksisterende bruker uten å oppi sensitiv informasjon direkte til oss. Siden det gjøres på denne måten trenger ikke brukere å lage en ny bruker på nettstedet, men kan bruke sin egen *Google*-konto. Det er også tryggere, da det ikke håndteres passord. Det lagres dermed kun informasjon som

navn, epost og bilde. Det lagres fortsatt som en bruker i vår database, dette er for å kunne koble *Google+*-brukeren med vår egen data. For filmer og tv-serier brukes *The Movie Database* sitt API. Dette API-et lar oss hente ut den dataen som trengs, som kan kobles med objekter i egen database, der det bare trengs å lagre ID-en til filmen eller serien. Deretter hentes resten av informasjonen fra *TMDb*-apiet.

Fig 1: Discover page

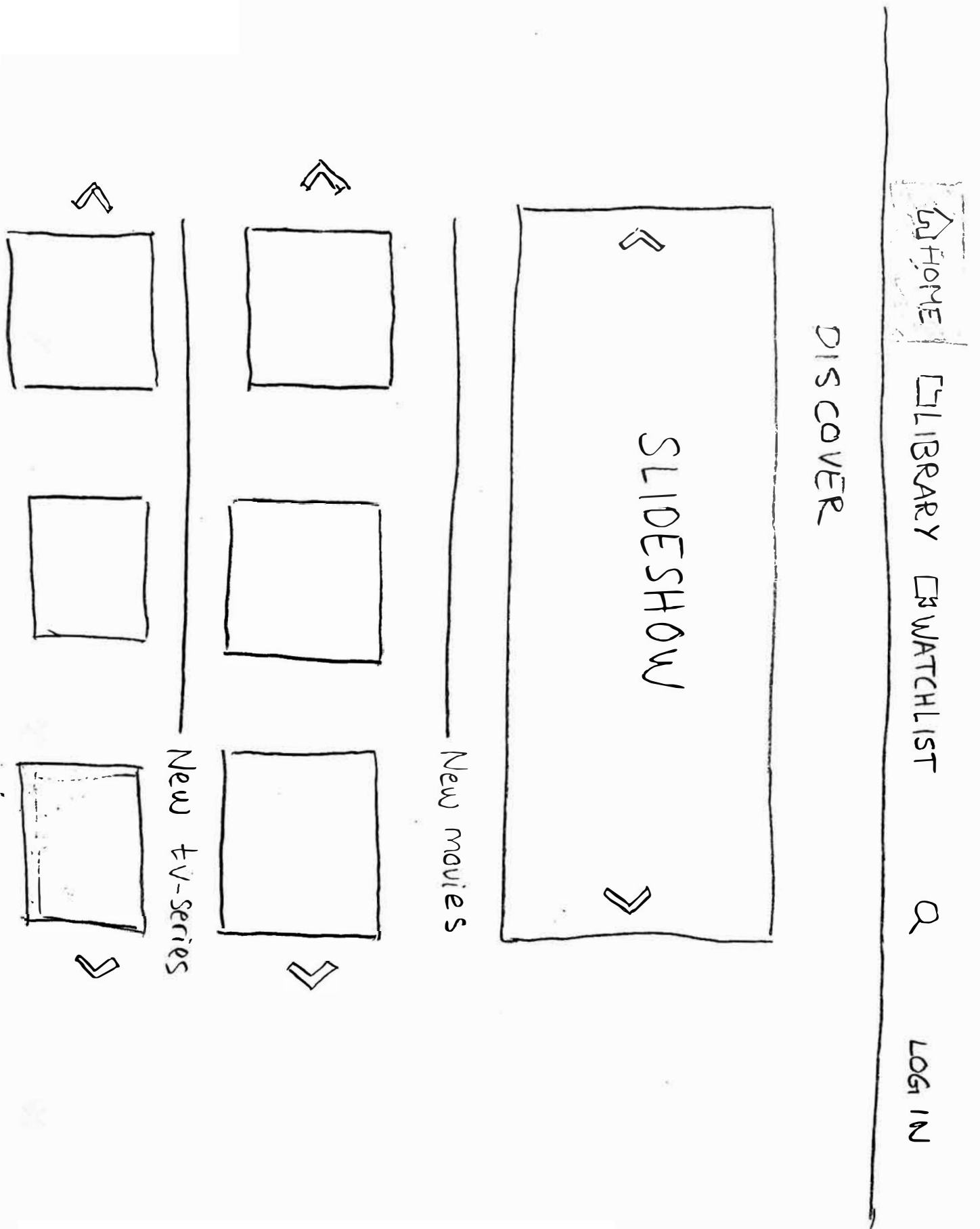


Fig 2: Library page

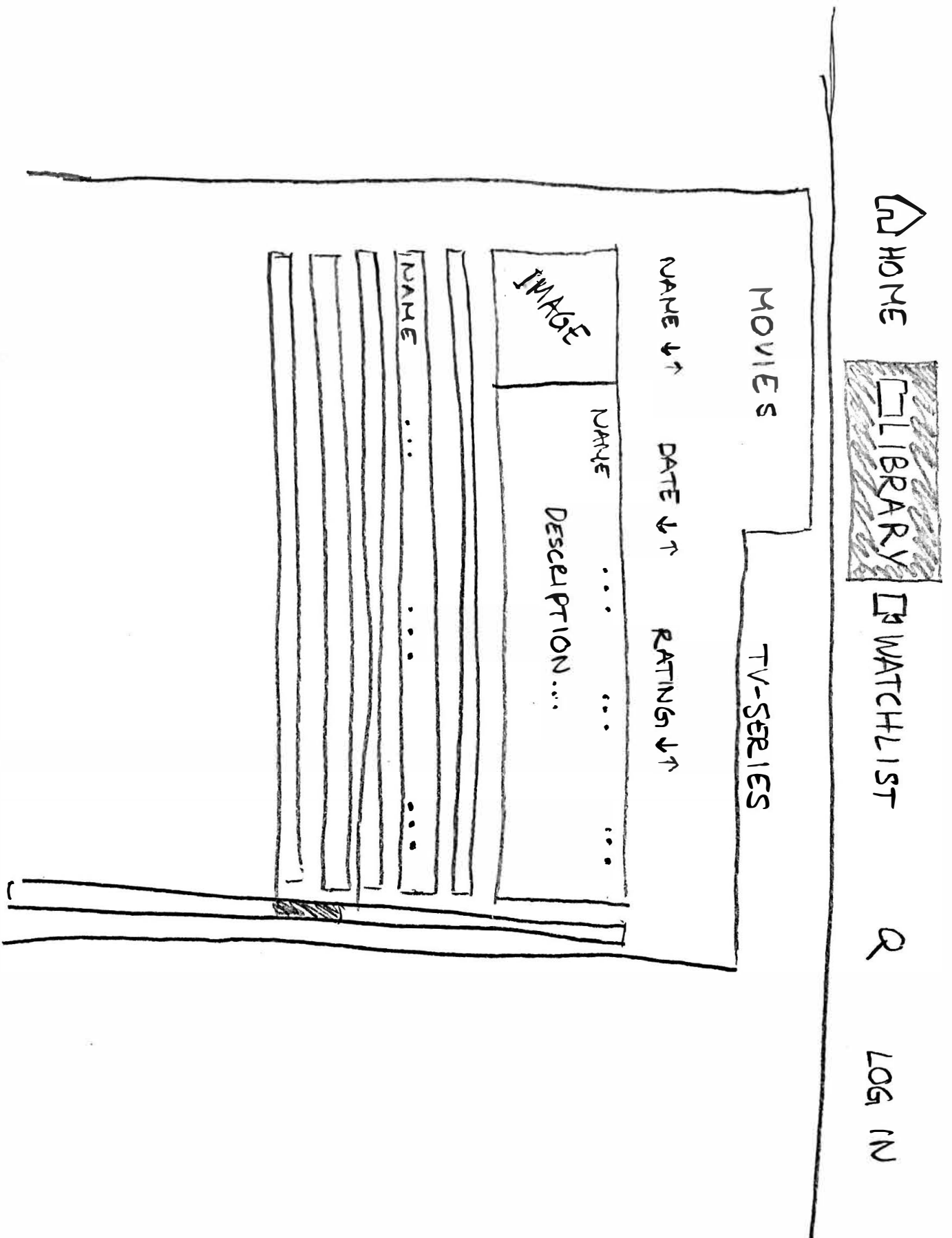






Fig 3: Search page

---

 HOME    LIBRARY    WATCHLIST   

---

IMAGE	NAME ADD WATCHLIST DESCRIP...				

Fig 4: Backend-arkitektur.

