IT2810-Webutvikling

Gruppe 42
Maria Soleim
Tollef Jørgensen
Erlend Skarpnes
Håvard Løkensgard

November 7, 2017

1 Applikasjonen

Discover HS er en webapplikasjon utviklet med *Angular 4*. Hensikten er å gi en enkel, intuitiv og minimalistisk opplevelse som gjør det enkelt for Hearthstone-interesserte å finne informasjon om Hearthstone-kort.
Applikasjonen vil bli bygd opp av flere forskjellige komponenter med tanke på best practic ved *Angular4*. Se appendix for flowchart over hovedkomponentene. Dette oppfyller kravspesifikasjon 1. *(Appendix: Kravspesifikasjoner, punkt 1.)*

2 Data

Dataen vi skal bruke henter vi fra en egen database. Dataen vil bli hentet ut som JSON-objekter som inneholder informasjon om kortene. Utgangspunktet for dataen er Hearthstone sin egen database. Vi henter ut informasjonen om kortene ved å bruke API'en til Hearthstone.

3 Planlagt implementasjon

- Mulighet for å bytte mellom faner *Cards*, *Profile* og *Log in/out*
- Mulighet for å søke etter kort ved navn
- Mulighet for å sortere kort med checkboxer:
  - Class
  - Type
  - Rarity
- Mulighet for å sortere kort med sliders:
  - Attack
  - Health
  - Mana
- Bytte mellom to forskjellige view ved bruk av en toggle knapp *CardView*:
  - *CardView:* Viser alle kortene ved bilder, laster inn flere kort ved å scrolle nedover siden (se figur 1).
  - *ListView:* Viser en liste over alle kortene. Ved å trykke på et navn vil listen utvide seg og vise frem det kortet samt informasjon om kortet (se figur 2).
- Lage en egen bruker
- Logge inn og ut
- Legge til et favorittkort

- Testing av kode og funksjonalitet
- God og kontinuerlig dokumentasjon av prosjektet

4 Løsning

4.1 GUI

Vi har valgt å ta i bruk biblioteket *ng bootstrap 4*. Grunnen til dette er at det er enkelt og godt utvikliet bibliotek. Under er det lagt ved 3 designforslag av siden, endringer ved designet vil skje før det endelige produktet er ferdig.

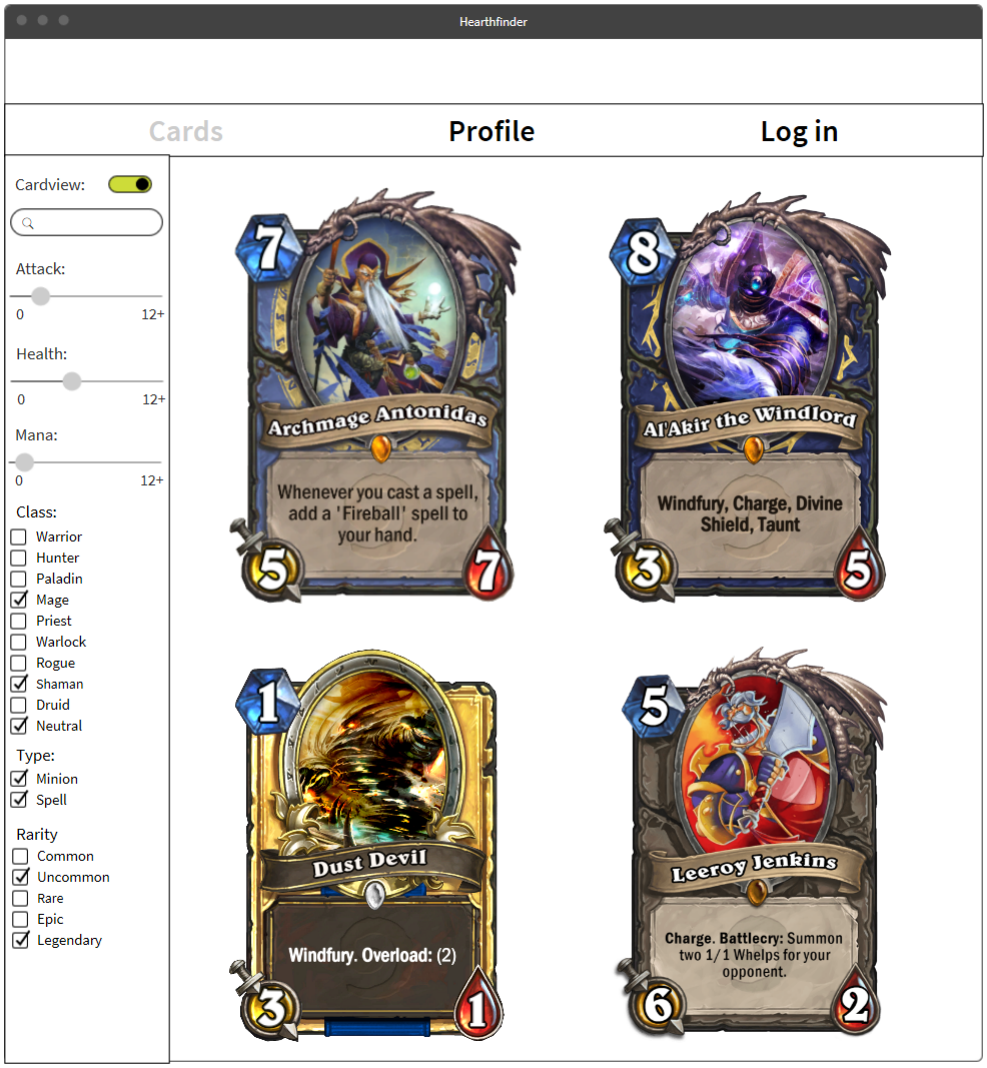

Figure 1: Mockup for Card view page. En alternativ visning av kortene. *(Appendix: Kravspesifikasjon, punkt 10.)*

Figure 2: Mockup for List view page. Listebasert visninger over alle kortene, med få detaljer. Ved å trykke på navnet får man opp mer informasjon og bilde av kortet. Dette løser kravspesifikasjon nummer fire. *(Appendix: Kravspesifikasjon, punkt 4.)* Det er mulig å sortere hvilke kort som blir vist ved å bruke checkboxene og sliderne. *(Appendix: Kravspesifikasjon, punkt 5 og 6.)* Lite visningen skal være implementert med endless-scroll.*(Appendix: Kravspesifikasjon, punkt 7.*)

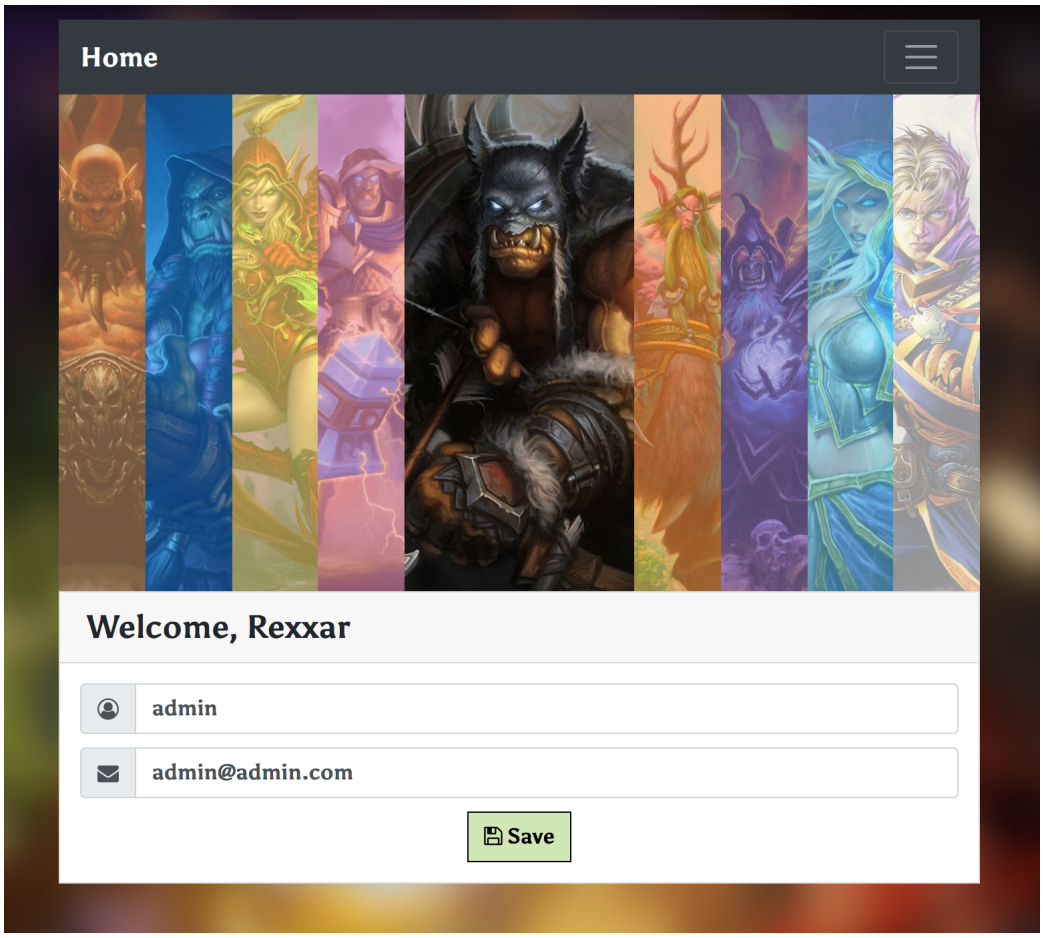

Figure 3: Mockup for Profile page (*iterasjon 4*). Brukeren skal kunne logge seg av og på. Informasjonen om brukeren blir lagret i databasen. Aktiviteten som blir lagret er hvilken *Hero* man velger som bruker. Den blir også highlighted hver gang man går inn.*(Appendix: Kravspesifikasjon, punkt 3, 8 og 9.)*

4.2 Database

Vi har valgt å bruke *MongoDB* som løsning for databasen. *(Appendix: Kravspesifikasjon, punkt 2.)* Den inneholder informasjon om alle Hearthstone kort, slik at vi enkelt kan hente ut informasjonen vi vil ha. Bildene til de forskjellige kortene blir hentet fra en annen server, dette er for å unngå at databasen tar for stor plass. Ved å bruke checkboxene og sliderne vil det bli gjort spørringer mot databasen. *(Appendix: Kravspesifikasjon, punkt 3.)*

Appendix

Flowchart

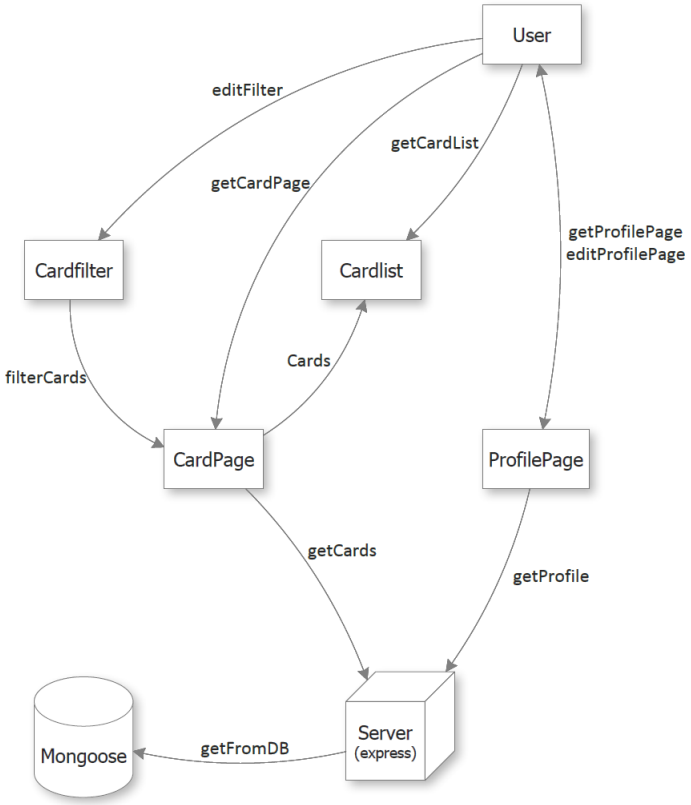

Figure 4: Flowchart over hovedkomponentene

Kravspesifikasjoner til prosjekt 4

1. Webapplikasjonen skal kjøres på gruppas virtuelle maskin og bruke node.js på serversiden, og skal være utviklet i Angular (bruk v2 eller v4, https://angular.io). Det er selvsagt greit å i tillegg bruke andre bibliotek eller løsninger som dere finner hensiktsmessig.

2. I webappliksjonen skal det inngå en backend database som kjøres på gruppas virtuelle maskin. Type database og hvordan denne brukes er opp til dere å bestemme, men grensesnittet til databasen skal være godt designet ihht. god praksis (bruk av REST ea).

3. Dere skal demonstrere både skriving og lesing til databasen fra webapplikasjonen inklusive en form for søk (i praksis dynamisk brukerdefinert utvalg av det som skal vises). Generelt er det mye artigere å jobbe med en datamengde som gir et realistisk inntrykk (eksempevis mulig å søke på forskjellige ting og få resultatsett som er forskjellige og har forskjellig antall). Bruk data dere finner på web, eller lag egne data.

4. Brukergrensensittet skal ha listebasert visning med få detaljer for hver enhet, og hvor målet er å vise brukeren hva som er i databasen eller hva som er resultatet av et søk. Brukeren skal ha mulighet til å se flere detaljer for hver enhet enten i et eget vindu, eller ved at listen enheten i lista har expand/collpase egenskap.

5. Den listebaserte visningen skal kunne sorteres på minimum to forskjellge egenskaper. Eksempel: etter at brukeren har fått returnert en liste etter et søk skal brukeren kunne bytte mellom forskjellige sorteringer.

6. Den listebaserte visningen skal kunne filtreres på minimum to forskjellge egenskaper. Eksempel: etter at brukeren har fått returnert en liste etter et søk skal brukeren kunne krysse av på en egenskap for å få begrenset antallet enheter i resultatsettet til kun de som har denne egenskapen.

7. Den listebaserte visningen skal ha dynamisk lasting av data. Eksempel: etter et søk vises de 10 første treffene, men flere lastes når brukeren scroller eller ved blaing i sider.

8. Webapplikasjonen skal ha "min side" funksjonalitet som i praksis betyr at en bruker skal kunne logge seg på og at det blir registrert noe fra brukerens søkeaktiviteten f.eks. hva brukeren har sett på tidligere eller søkene som brukeren har brukt.

9. Webapplisjonen må implementere "session"-håndtering (som du f.eks. trenger for å implementere dynamisk lasting, min side, og filtrering/sortering som skal fungere med sidevisning).

10. Webapplikasjonen skal ha et litt "fancy" alternativ visning av listen f.eks. visning på kart eller visuell grafisk fremstilling av data, ordsky ea.

11. Kode skal være testet og funksjonaliteten skal være godt utprøvd og feilfri.

12. Prosjektet skal være godt dokumentert, slik at det er lett å sette seg inn i for andre.