Hovedopgave

Mikkel Bendix Munk
mikk2385@edu.eal.dk

Christoffer Frydkjær Kristensen
chri635r@edu.eal.dk

4. november 2015

Indhold

Gruppekontrakt

- Vi arbejder efter SCRUM
- Sprint varighed: 14 dage
- Mødetid: senest kl. 8.00
- Daily standup: kl. 8.15
- Arbejdsugen er 37 timer inkl. frokost
- Kodekoordinering, gør opmærksom hvis man tager en ny opgave
- Spørg om hjælp
- Frihed under ansvar, det forventes af gruppens medlemmer at man overholder aftaler og gør alt hvad man kan for at overholde deadlines
- Hvis der bliver konflikter, skal vi turde tale om det og finde en løsning
- Hvis man skal gå før, gør man opmærksom på det til daily standup
- Hvis man er forhindret i at møde, kontaktes gruppen hurtigst muligt

Kvalitetssikring

Vi vil kvalitetssikre vores produkt ved at teste alt det der giver mening at teste, så vi sikrer at funktionaliteten fungerer efter hensigten. Derudover laver vi reviews på hinandens kode inden det bliver godkendt og merges til master branchen i git. For at sikre læsbarhed og konsistens i koden, følger vi nogle specifikke PHP kode standarder som er beskrevet i afsnittet omkring kode standarder. Efter hvert sprint holder vi et review møde hvor interessenter fra Ordbogen A/S vil deltage og komme med feedback. Så vi på den måde hele tiden sikrer os at det vi laver, er det Ordbogen gerne vil have. For at sikre kvaliteten i projektet yderligere vil vi bruge FURPS som kvalitetsmodel og hele tiden veje projektet op imod FURPS.

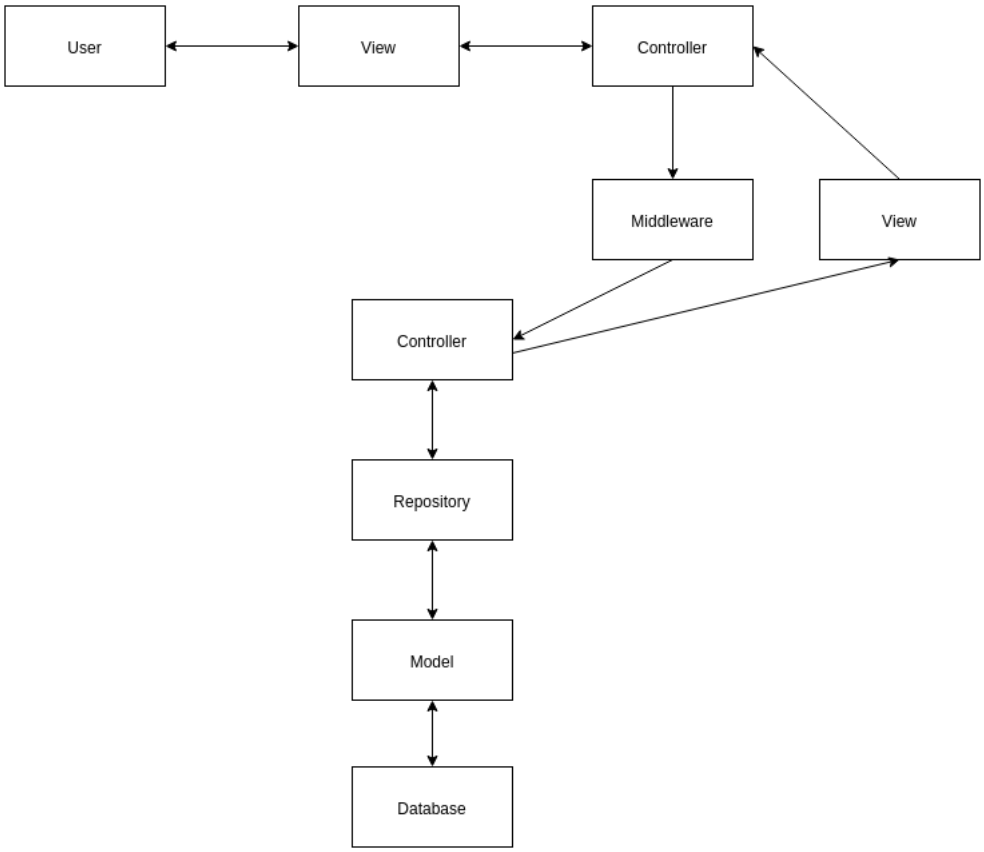
Produktbeskrivelse

Systemet kan læse, gemme og vise statistik på en præsentabel måde. Systemet er delt op i to subsystemer, hvoraf det ene tager sig af modtagelsen af data og det andet tager sig af visningen af den data. Grunden til vi har to subsystemer er at vi tager udgangspunkt i to typer af brugere, hhv. brugeren der indlæser data og brugeren der aflæser data. Vi har designet systemet efter brugerens behov, som har været:

- bla bla bla

Arkitektur

Vi har valgt at køre angularjs i frontend og derfra kalde RESTful APIs i backenden. I figur 1 kan man se hvordan det samlede system kommer til at se ud, hvor vores middleware/routing lag kalder backenden, som returnere et view, med de dataer, der skal bruges.



Figur 1: Vores valgte arkitektur for både frontend og backend