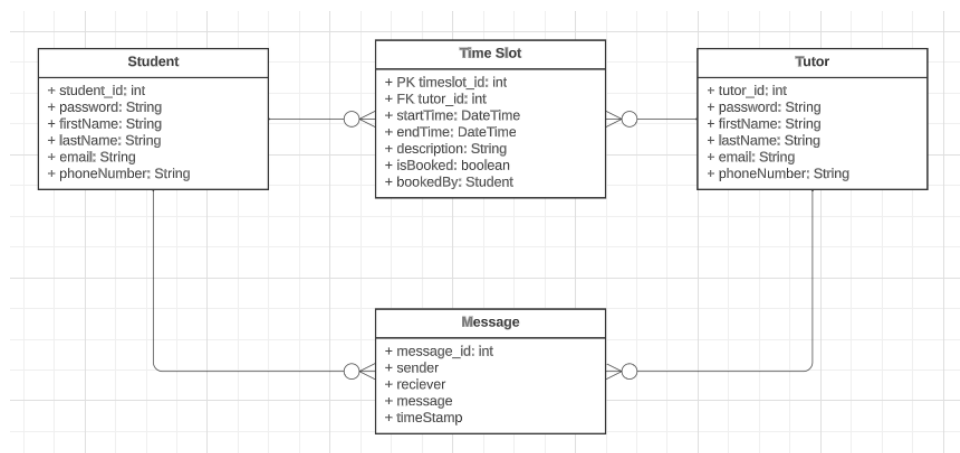


Database struktur



Time Slot vil være timene som er plottet inn i kalenderen, en Tutor (lærings assistent) vil kunne opprette null eller flere Time Slots, og en student vil kunne booke null eller flere timeslots.

Denne funksjonaliteten vil være hovedfokuset, hvis gruppen har nok tid vil melding tjeneste bli implementert. Det samme vil gjelde email notifikasjon og mulighet for å bytte språk på nettsiden mellom engelsk og norsk.

«Klassebiblotek»

Gruppen bestemte seg for å bruke laravell rammeverket i prosjektet. Begrunnelsen er for at laravel er et relevant rammeverk som er mye i bruk og siden gruppen består av tre medlemmer vil det være mulig for gruppen å sette seg inn i rammeverket og lage et fullstendig system. Samtidig har vi kjennskap fra denne fremgangsmåten fra tidligere som gjør det overkommelig og ta i bruk et slikt rammeverk.

-Laravell rammeverk

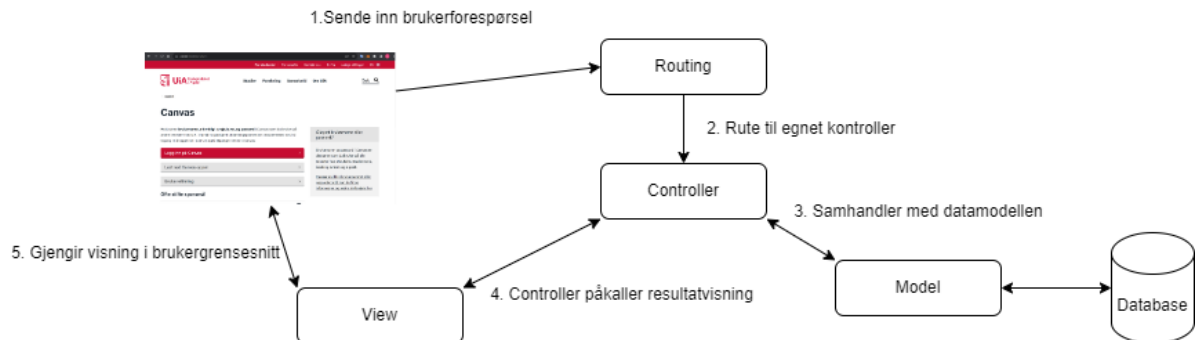
Ved å følge Larvavell kan vi dra nytte av MVC arkitekturen, noe som gjør det lettere for oss å organisere kode og separere logikk fra presentasjon. Her kan vi også ta å interagere med databasen med objektorientert syntaks for å håndtrere bookingdata, brukerinformasjon osv. Laravell gjør også at vi kan definere måter å rute på, samtidig som det byr på brukerautentisering for å sikre tillatelser.

-jQuery

Ved bruk av jQuery kan vi enkelt lage interaktive funksjoner som kalenderwidgets for å velge datoer, tidsluker, animasjoner osv. Vi har også muligheten for å implementere Ajax for å hente, vise og sende data til serveren uten å laste siden på nytt.

-Bootstrap

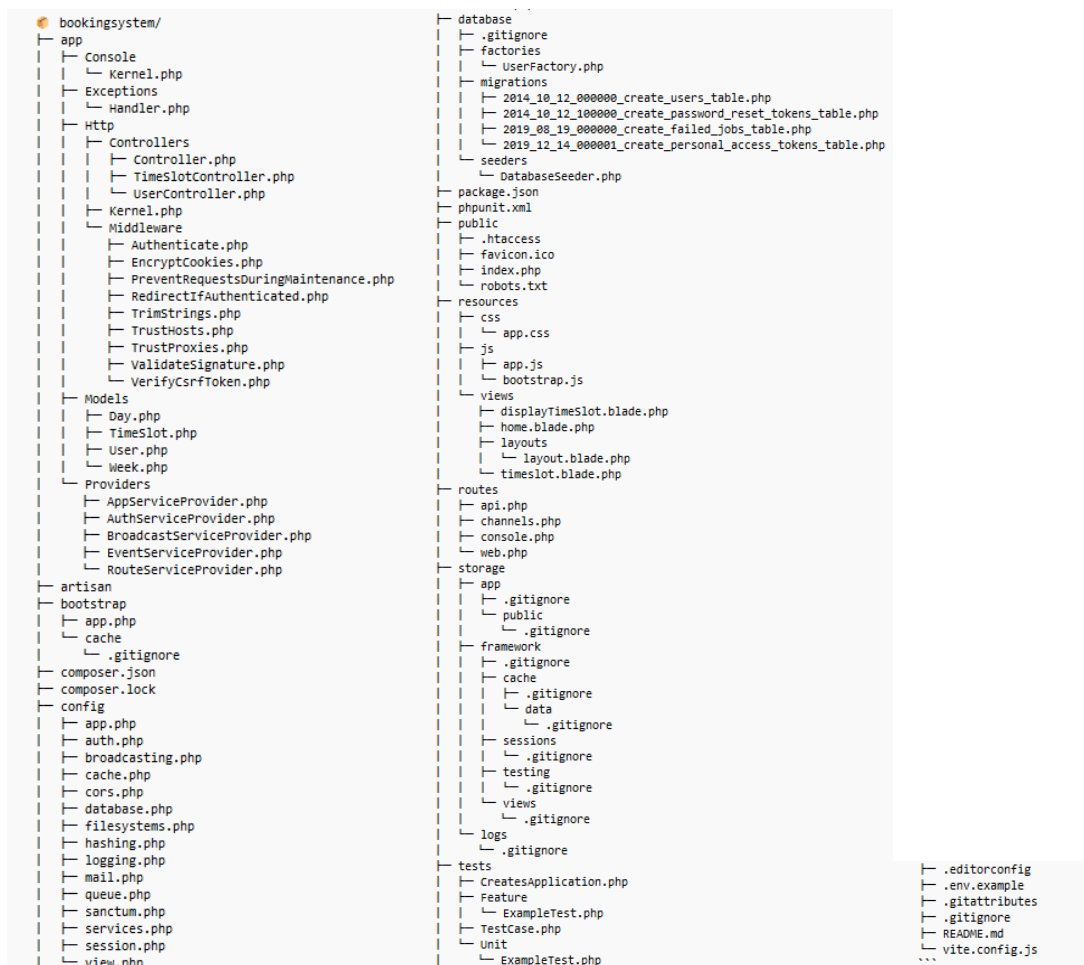
Dette er et frontend rammeverk som lar oss lage responsivt design og ta nytte fra det. Dette har mange forhåndsbygde komponenter som knapper og varlser for å bygge et profesjonelt brukergrensesnitt.



Figur: MVC Arkitektur

Filstruktur

Filstrukturen til prosjektet vil se ut som vist i bildet under:



Det meste som gruppen kommer til å samhandle med i rammeverket er kontrollerne, models, views og web.php. App/Models mappen inneholder klassene som er blueprint for objektene som blir hentet fra database, og andre objekter som blir laget, men som ikke er i database, eksempel Week og Day som bare eksisterer i webapplikasjonen. Gruppen baserer seg på MVC rammeverk. Alle filer som har med samhandling mellom det som vises på skjerm til bruker og database går igjennom kontroller filene, mens views er siden som vises til bruker, og models er de objektene som blir sendt fra views til databasen og metodene som er tilknyttet dem. Andre klasser som er nødvendig som Week og Day som ikke har noe med databasen å gjøre er også lagret i Models mappen.

Resources mappen inneholder views, js og css. Views mappen inneholder alle viewsene som er av typen .blade.php, det er disse filene som blir vist til brukerne og som brukerne kan samhandle med. JS og CSS mappene inneholder javascript og css filer.

Under er kort beskrivelse på filstrukturen til laravel rammeverket, ikke nødvendig å lese hvis du allerede kan det.

App mappen er der du finner kjerne koden til applikasjonen.

Console mappen inneholder filer som lar applikasjonen bruke console kommandoer.

Exeption mappen er hvor applikasjonen håndterer exceptions som kommer ifra applikasjonen.

Http mappen inneholder kontrollerne som håndterer data fra og til databasen.

Models er hvor klassene som er blueprints for objektene som blir hentet fra databasen er lokalisert, her har gruppen også lagt til et par klasser som ikke har noe med data fr databasen å gjøre, ny definisjon for gruppen er en plass for å lagre alle klasser som har noe med views å gjøre.

Providers inneholder falle service providersene til applikasjonen, for øyeblikket er det bare tjenester som komm med rammeverket som er der, men det kan bli lagt til flere hvis nødvendig.

Bootstrap mappen er hvor app.php filen ligger, denne filen er den første som kjøres og starter opp prosjektet. Innen i denne mappen ligger også en annen mappe som heter cache som inneholder filer som hjelper ved optimaliseringen av koden og inneholder service og route cahce filene.

Config mappen inneholder alle konfigureringsilene til applikasjonen.

Database mappen inneholder filer som håndterer migrering av data og factories mappen hvor filer som lagrer mock data til databasen kan bli lokalisert hvis gruppen bestemmer seg for å bruke denne funksjonaliteten.

Public inneholder index.php filen, som er filen som starter hele applikasjonen.

Resources mappen inneholder css filer, javascript filer og blade php filer, det som vises til brukeren og front end kode finner du i denne mappen.

Routes mappen inneholder filer som har med rute håndteringene til applikasjonen.

Storage mappen står for lagring av driftsdata. Dette inneholder filer som applikasjonen generere, som opplastede dokumenter og kompilerte maler. Den lagrer også systemlogger og midlertidige cache-filer.

Tests mappen er hvor alt som har med tester er lokalisert.

.env er en konfigureringsfil som lagrer miljøspesifik konfigurasjoner, f.eks api nøkler, variabler eller connection til database.

Composer.json er en fil som håndterer dependencies og eksterne bibliotek. Composer genererer også en composer.lock fil som låser de nåværende versjonene av dependensiene som blir brukt i prosjektet.