Dokumentation

# Syfte och Överväganden

Syftet med denna webbapplikation är att kunna se över och hantera boter som har

förvärvats av användaren. Användaren av applikationen skall kunna hämta, acceptera

och betala, eller bestrida en given bot via applikationen.

Jag la vikt på att lära mig de verktyg, ramverk, tillvägagångssätt,

och dokumentation som skulle krävas under ett faktiskt arbete, då med att jag

såg att det var mer användbart för mig ifall jag blir anställd,

och mer användbart för er i evalueringsprocessen.

Detta dock gjorde att jag inte hade så mycket tid att implementera en slutförd

lösning. Min tidigare erfarenhet var att bara hitta första tillgängliga

lösningen, men såg inte detta som hållbart för denna uppgift givet kontexten.

# Design

## Databas

Jag började med att installera Lavarel Homestead, och fixade så att sidan

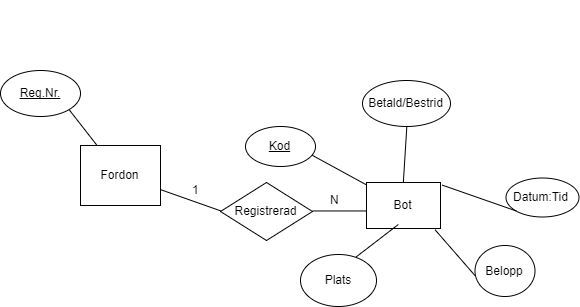
ovningsuppgift.app blev tillgänglig. Jag sedan bytte Lavarel version, och

läste på lite hur den användes. Sedan ändrade jag version på Lavarel, och

startade upp denna applikation. Jag sedan såg över hur jag skulle designa

databasen enligt ER-modellen, och design-mönstret på applikationen.

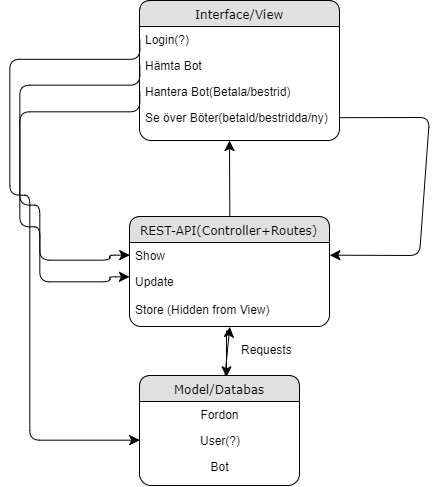
Såhär såg ER-modellen ut av Databasen Botdb:



*Ingen User krävs då botkoden fungerar som lösen. Ett fordon kan ha flera böter på sig men måste åtminstone ha en. Reg.nr ligger i relationsmodellen både i fordon och i bot som foreign key. Users och passwords\_reset tabellerna kommer med i lavarel/databasen men används inte i applikationen, så är därför inte med i datamodellen.*

Jag korrigerade sedan i env. Variabeln så att källkoden kunde kommunicera med databasen.

## High-level/Konceptuell design



# Metod

Här börjar jag med att använda artisan migrations kommandot, exempelvis för en bot:

php artisan make:migration create\_bot\_table

Detta gör jag inuti den virtuella maskinen (VirtualBox, via Homestead/Vagrant) och modiferar php-filerna. Detta ska facilitera ändring av andra arbetskamrater. Lavarel Schema är också inbyggt vilket gör att andra databaser skall kunna interagera med den.

Sedan matar jag in tabllerna i databasen med kommandot:

php artisan migrate

# Section 2