

Overvejelser til projekt:

Business Case:

Ved Mathias' Andelsforening, har der førhen været gratis parkering. Desværre udnyttede fremmede der ikke boede i andelsforeningen dette, hvilket førte til fyldte parkeringspladser og dette resulterede i, at foreningen valgte at indføre parkeringsvagter.

Beboere i andelsforeningen kan via foreningsformanden registrere sin nummerplade, så de ikke modtager bøder. Desuden skal gæster af beboeren vedlægge en parkeringsbillet i forruden. Denne process er oldschool og kræver også kontakt til foreningsformanden når man løber tør for parkeringsbillerter.

Andelsforeningen har brug for en løsning der sørger for, at beboere kan administrere parkering på en nemmere måde, da man får udleveret få parkeringsbillerter ad gangen fra formanden.

Kravspecifikation:

Andelsforeningen ønsker et program, hvor man som admin kan oprette nuværende samt nye beboere i et system, hvor beboerne kan oprette en virtuel parkeringsbillet til deres gæster. Denne billet skal være aktiv i 24 timer, hvorefter man som beboer skal kunne genoprette billetten, hvis ønsket.

Selve systemet, skal være bygget således, at alle kan benytte det, unge som gamle. Man skal som beboer kunne have sin egen personlige konto, hvor man også kan se en liste af alle de registrerede gæste nummerplader, her skal man også kunne genaktivere gamle registrerede nummerplader.

Funktionelle Krav for hele projektet:

- Bruger Login
- Admin Login
- Bruger funktionalitet til at oprette gæsteparkering i systemet.
- Bruger funktionalitet til at se alle nummerplader oprettet i brugerens navn.
- Admin funktion til oprettelse af beboere i foreningen - således at ingen andre end beboere kan blive oprettet.
- Admin skal kunne oprette beboers nummerplade permanent.
- Parkeringsvagt skal kunne scanne nummerplade og tjekke status på gæst eller beboer

Definition af funktionel krav til én essentiel funktion for vores business case

- Bruger funktionalitet til at oprette gæsteparkering i systemet.

Funktionelle krav:

- Bruger Login
- Bruger funktionalitet til at oprette gæsteparkering i systemet.

User Stories

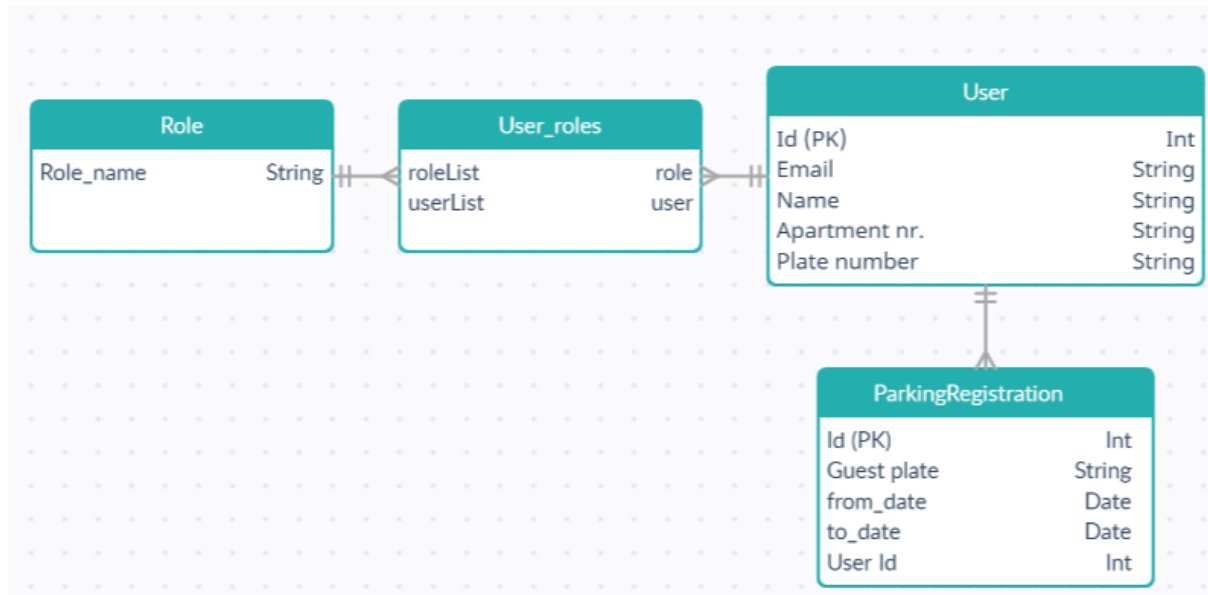
- As a User or an Admin I want to be able to Log in, so that I can access the application.
- As a User I want to be able to register my guest's car into the application, so that they won't get a fine from parking at my apartment.
- As a User I want to be able to see a list of previously registered license plates, so that I can see a saved list of who I have earlier registered.
- As a User I want to be able to update a saved license plate with one click, so that I don't have to re-register the plate again.
- As an Admin only I want to be able to register new residents as Users on the application so that no one else can create new users for safety measures.
- As an Admin I want to be able to register the residents license plate so that i can control who is in the system.
- As a Traffic Warden I want to be able to scan the license plate so that I can check whether or not the parking is validated.

Teknologi stack:

(De nævnte teknologier er blevet valgt ud fra at det er de teknologier som er blevet brugt igennem vores uddannelse)

- Java
- ReactJS
- JavaScript
- CSS
- mySQL
- JPA & JPQL
- DigitalOcean som hosting service
- Github til versionsstyring

ER diagram:



Mockup til Mobil:

GuestParking [Sign up](#)

GuestParking [Logout](#)

GuestParking [Logout](#)

Name

Licens plate

Email address

Submit

Password

Login

Thank you for your registration!

Licens plate: *****
Date: 01/01/2022
Expires: 02/01/2022

Kodestandard:

Fokus på kode læsbarhed

Det er vigtigt for udvikler teamet at have korrekte kodnings konventioner og standarder når der skal udvikles et projekt. Dette vil hjælpe teamet til at bibeholde kodekvalitet og kan formindske tidsspild i forhold til, hvis nye udviklere skal forstå kompleksiteten bag koden.

- Generelt set skal koden være nem, at læse for udefrakommende udviklere. Gør derfor brug af funktionsnavne der giver mening.
- Brug af DRY(Don't Repeat Yourself) princippet, automatisér gentagne opgaver når nødvendigt - det samme stykke kode bør ikke gentages i et script.
- Funktioner skal ikke være for lange, idéelt skal en enkelt funktion kunne klare en enkelt opgave.

Test:

Til projektet vil vi benytte os af test driven development(TDD).

- Der vil blive lavet unit tests.
- Der vil blive lavet integration tests.
- Der vil blive lavet REST Assured.
- Hver test vil være selvstændig.
- Der vil også blive benyttet try/catch for at fange fejl i koden.

Sikkerhed:

Vi vil i form af sikkerhed benytte os af:

- RBAC (Role Based Access Control).
- Accountability (Log key events E.g. CRUD operations).
- Kryptere sensitive data.
- Følge relevante OWASP retningslinjer.

Beskrivelse af pipeline:

- Vores pipeline er til for at sørge for, at der ikke bliver pushed en opdatering op på systemet, hvor der ikke er kørt test på, hvis ikke der er kørt test på koden, kan vi ikke være sikre på, at der ikke er fejl i koden. Vores pipeline builder vores code, kører tests og sikkert deployer den nye version af systemet. Automatiserede pipelines forhindre manuelle fejl, og udbyder standardiseret feedback til udviklere.

Videreudvikling:

- Til videreudvikling, ville vi ændre funktionaliteten af admin funktionen og oprette en formands role istedet for, således vil der være en person fra hovedkontoret af der styrer det overordnet system, så der bliver tjekket op på, at formanden ikke udnytter sin stilling. Formanden vil fortsat skulle oprette tilflyttende beboere i systemet, samt deres nummerplade til permanent parkering.

- På nuværende tidspunkt, vil vi i vores system udelukkende skulle skrive nummerplade teksten ind manuelt og så sammenligne med data i databasen. Fremadrettet vil vi finde en løsning så vi faktisk ville kunne scanne / tage et billede af nummerpladen, konvertere teksten på billedet til en String og sammenligne med databasen for at finde valid status.

Hvordan vi vil færdiggøre projektet:

- Først og fremmest ville vi selvfølgelig kode hele projektet i stedet for at pseudokode det. Dette har vi valgt at gøre fordi, vi følte at det var det mest realistiske i forhold til at vi har en deadline i projektet, samt skole ved siden.
- Når det handler om programmering, er det et mega svært spørgsmål at besvare: "Hvordan eller hvornår er projektet færdigt?" Da der med al sandsynlighed altid vil være nye ting man kan implementere, eller at man finder ny teknologi, eller tilegner sig ny viden til, hvordan man kan definere hvornår et projekt er færdigt. For at kunne færdiggøre projektet ville vi begynde en iterativ process og udarbejde en mere detaljeret planning phase - aka. den indledende del i projektet - Dertil ville vi udarbejde en specifik projektplan over projektet. Dette er noget vi ikke har set, at vi kunne gøre, da vi har valgt at pseudokode projektet og ikke gå i gang med den reelle kode.