

## Kvar att göra 2020-03-23

### I Programmet

#### Menu

- ShowMenu() // Visar välkomst text
- Login() //Text CheckIn Menyval , användaren väljer detta menyval, får sedan skriv sitt namn
  - ShowShips() //Om man lyckas visas ShowShips, som använder returnerad Dictionary från Starport.
  - Användaren väljer skepp via att skriva in menyval tex 1, 2, 3, 4
  - Metod som bekräftar incheck, tar tillbaka användaren till menyn för att logga in igen.
- Logout() Text CheckOut, Menyval //användaren väljer detta menyval
  - Ett kvitto visas från metod i Starport
  - Någon loop om användaren vill checka in något nytt fordon

#### ValidateUser

- Tar in boolmetod GetAsyncTraveler() från StarWarsApi-klassen, om true så finns användaren i databasen, om returnerar false finns den ej. Något meddelande om man lyckas eller inte

#### Starport

- Dictionary<string, int> //Tar in metoden GetAsyncShipData från StarWarsApi-Klassen, gör om den returnerade listan till en dictionary som finns i Starport.
- Efter tagit in listan, randomiserar antal 1-10 för varje skeppskategori och lägger in i Dictionaryns value, för varje Key.
- Metod för att returnera Dictionaryn när den är klar med antal parkeringsplatser
- Metod för att dra av antal parkeringsplatser med -1 om någon checkar in
- Metod för att "lägga tillbaka" parkeringsplater med +1 om någon checkar ut
- Metod för att beräkna parkerings-tariff efter utcheckning. Tariffen är baserad på skeppstyp som väljs, samt hur lång tid man parkerat
- Metod som visar något slags fint formaterat kvitto () som används i Menyn efter Checkout

### Databasrelaterat

- Skapa Connections för databas som skapar upp 2 tables
- Skapa SaveEntry till databasen-metod när en användare checkar in.
- Skapa CreateReceipt till databasen, när en användare checkat ut och sparar kvittot som visas för användaren på skärmen.

- One to Many-förhållande mellan användare/incheckning -> kvitton. En användare kan ha flera kvitton, men ett kvitto kan inte ha flera användare.