

# Övning 11: Garage 2.0 - Återkomsten del 1.

Garaget kommer till webben i ASP.NET MVC! Då detta är den första MVC-baserade garageversionen så kommer vi att fokusera på gränssnitt och funktionalitet. Uppgiftsbeskrivningen är skriven i prioriteringsordning.

Då det här även är er första gruppuppgift kan det vara en bra ide att börja med att göra en planering vad som ska göras och i vilken ordning. Hur ni ska dela upp arbetsuppgifterna? När anses en implementation vara klar enligt er i gruppen. Har ni någon strategi hur ni ska jobba med github?

Viktigt nu när vi jobbar på distans att ni har en kod kommunikation. Utarbeta gärna egna förhållningsregler hur ni bäst ska kunna samarbeta.

Varje morgon ska ni ha ett **kort** möte max 15 min. Det går ut på att ni ska informera varandra om hur ni ligger till i gruppen. Var och en i tur och ordning berättar:

Vad har ni gjort sen sist?

Vad ska ni göra under dagen?

Är det något som hindrar er från att komma vidare?

## Singelmodell

I denna version vill vi bara använda en ensam modell. Alltså inte en modell för varje fordonstyp som vi gjorde i konsolversionen, utan endast en *klass*

ParkeratFordon/ParkedVehicle med properties som motsvarar:

fordonstyp, registreringsnummer, färg, märke, modell, antal hjul samt ankomsttid.

## Grundfunktionalitet

- Fordon skall kunna parkeras och hämtas ut.
- Fordon ska kunna editeras efter de har parkerats.
- Fordonets ankomsttid ska ej kunna editeras eller manipuleras.
- Vid parkering skall användaren kunna fylla i fordonets data.
- Det skall finnas en **översiktsvy** där alla parkerade fordon visas med **viss** grunddata Typ, RegNr, Ankomsttid, (Parkeradtid<sup>1</sup>).
- En **detaljvy** för varje fordon där **övrig information** visas, t.ex. antal hjul, märke, färg & modell.
- Registreringsnummer ska vara unikt.
- Vid uthämtning av fordon ska kvitto erbjudas.

---

<sup>1</sup> Bonusuppgift: Skriv ut hur länge fordonet har varit parkerat i garget

## Utseende

Applikationen skall ha ett enhetligt utseende, tydligt och lättnavigerat. Språk och instruktioner skall vara anpassade för användningen, exempelvis: vi skapar inte en bil - vi parkerar/checkar in bilen, vi skall inte heller radera en bil - vi "hämtar/checkar ut den" osv. Om ni väljer att göra applikationen på Svenska eller Engelska är valfritt.

## Input-kontroll

Vissa input vill vi inte att användaren skall kunna skriva hur de vill i fritext. Viss input kontroll ges av scaffoldingen (endast nummer i sifferfält osv), andra vill vi hantera manuellt.

## Fordonstyper

Fordonstyp skall gärna inte heller skrivas in manuellt, utan skall finnas som en *dropdownlista* som bara innehåller giltiga fordonstyper (I framtida projekt kommer vi dock lösa detta på annat sätt).

## Tidstämpel

En tidsstämpel för när fordonet parkerades (lagras i databasen tillsammans med övrig fordonsinformation) **ska inte** matas in manuellt utan skall genereras och sparas automatiskt när användaren checkar in ett fordon.

## Kvitto

I samband med utcheckning så har vissa kunder ett behov av ett kvitto för att redovisa hur länge de parkerat. Implementera en ny vy som visar fordonsinformationen, RegNr, in/ut-checkningstid, total parkerings period och pris automatiskt efter att en bil checkats ut. Kvittot ska använda sig av en **ViewModel** som model. Gör gärna kvittot utskriftsvänlig. Ni sätter ert eget pris.

## Filtrering och sortering

I takt med antalet parkerade fordon så växer behovet av en sökfunktion. Implementera en sökning för registreringsnummer.

Bonusuppgift:

Om tid finns så utöka sökningen till de andra fälten. Implementera även sorterings knappar ovanför kolumnerna i översiktsvyn så att användaren kan välja en kolumn att sortera fallande eller stigande. Ni väljer själva om ni sorterar i backend eller frontend.

## Feedback

Efter ett fordon parkerat, editerats eller liknande vill vi som användare få feedback på att det gått bra eller dåligt. Så att vi exempelvis slipper titta efter själva i listan att fordonet verkligen har uppdaterats.

Lycka till!