

# Övning 14 - Garage 3.0

I den rafflande följetongen 'Garaget' fortsätter äventyret men en utbyggd databas och ytterligare funktionalitet. I ett försök att hålla oseriösa söndags förare ute stänger man portarna för icke-medlemmar. För att parkera i garaget måste föraren finnas registrerad i applikationen. För att inte tappa alla kunder så är medlemskapet gratis, och registreringen går snabbt och enkelt. När vi ändå skriver om databasen passar vi på att skapa en egen, normaliserad, tabell för fordonstyper (ersätter den Enum ni använde i 2.0 versionen).

## Relationsdatabas

Läs igenom övrig uppgiftsbeskrivning och starta sedan arbetet med att rita ett ER-diagram.

*ER-diagrammet skall godkännas innan ni startar en implementation.*

## Funktionalitet / Krav

- Medlemmar skall kunna registreras och lagras i databasen.
- När en medlem registreras ska personnummer anges.
- Personnumret ska valideras att det är i ett korrekt format.
- Förnamn får inte vara samma som efternamn
- Enbart ett konto per medlem tillåts.
- Ett fordon har endast en ägare.
- En sökbar översiktsvy för registrerade medlemmar. Där vi även ska kunna se hur många fordon varje medlem har registrerade. Från översiktsvyn ska vi kunna navigera till ägaren och se alla fordonen.
- Denna översiktsvy ska sorteras enligt instruktion längre ned i övningsbeskrivningar.
- När ett fordon parkeras skall det kopplas till en registrerad medlem
- Endast medlemmar över 18 år ska kunna parkera ett fordon
- När ett fordon parkeras kopplas det till sin fordonstyp.
- Det ska gå att lägga till nya fordonstyper.
- En *översiktsvy* för parkerade fordon som innehåller: Ägare, Medlemskap, Fordonstyp, RegNum och ParkTid som minimum.
- Sökfunktion för fordonstyp och registreringsnummer i översiktsvyn.
- Kvittot ska ha information om vilken medlem som parkerat.
- Kvittot ska även hålla information om hur mycket vi sparar vid eventuellt pro medlemskapet.
- Övrig funktionalitet från Garage 2.0 del 1 och 2 (se tidigare uppgiftsbeskrivningar)

## Medlemskap

Pro uppnås enligt:

- Automatiskt vid registrering och gäller i 30 dagar. Därefter Member
- Alla över 65 erhåller 2 års pro medlemskap.
- Ovanstående börjar löpa direkt vid registreringen. Sista dagen gäller pro ända fram till midnatt.

## Prissättning

Det har även kommit in nya krav på prissättningen. Det är först en grundavgift som tas ut när fordonet parkeras. Samt olika taxor beroende på vilket fordon som parkerats. Sen har vi även medlemskap nivån som påverkar prissättningen.

Fordon som tar upp en plats	grundavgift + timpris * tid
Fordon som tar upp två platser	grundavgift * 1.3 + timpris * 1.4 * tid
Fordon som tar upp tre eller fler	grundavgift * 1.6 + timpris * 1.5 * tid

Vid Pro medlemskap minskas grundavgiften med 10% för varje plats medlemmen tar upp i garaget när ett fordon checkas ut. Maximalt kan 40% rabatt uppnås på grundpriset. För större fordon rabatteras även den löpande kostnaden med 10% med Pro medlemskapet.

Om pro medlemskapet upphör under en parkerings period beräknas priset fram till dess att det upphör enligt Pros prisplan resterande enligt ordinarie taxa.

## Sortering av medlems översiktsvy

Denna vy ska sorteras enligt följande regler.

- Sorteras på medlemmens förnamn.
- Sorteras endast på de två första tecknen i namnet
- Om två namn har samma första två tecken så behåller vi ordningen på dessa (stable sort)
- Sorteringen är case sensitive enligt ASCII (versaler sorteras före gemener)

## Taktik

Vi kommer att lägga till flera nya vyer och det är en god idé att ta några minuter för att tänka igenom applikationens flöde och navigation. Hur passar vyerna ihop? Vart borde man göra vad? Hur kommer applikationen att användas osv.

Garage 2.5 går alldeles utmärkt att bygga ovanpå er implementation av 2.0. Det går även bra att börja från scratch.

## Parprogrammering

Turas om att sitta vid tangentbordet och byt regelbundet t.ex. var 15e minut. Den som sitter vid tangentbordet skriver, medan de andra berättar vad denne skall skriva (*förare* och *navigatören*). Detta är ett vedertaget arbetssätt för två programmerare på en dator och väl utfört är det effektivare än om de skulle arbetat på separata datorer - extra sant i projekt som inte lämpar sig väl för parallell utveckling.

## Extra

- När fordon checkas ut tas de ej bort ur databasen. När en medlem kommer och ska parkera kan man välja från den medlemmens registrerade fordon.
- Implementera paginering tex max 20 träffar per sida.
- Implementera att användaren kan välja hur många träffar varje sida ska innehålla

- Bygg en ViewComponent som visar information om lediga och upptagna platser.
- Bygg en Taghelper som visar en ikon när ett fordon är parkerat och en annan när det är utcheckat
- Använd er av en service för att generera en dropdown i parkera och edit.
- Läs in hur många platser garaget har från config fil.
- När man trycker på en medlem får man fram en totalsumma på alla dess parkerade fordons totalkostnad.
- Hitta på kvalificeringsregler för att uppnå pro igen. Tex parkerat en viss mängd över en viss tid osv
- Möjlighet att lägga till fler typer av medlemskap tex Gold.
- Möjlighet att kunna boka parkeringsplats för fordon i förväg.
- Söka på första lediga tid för ett specifikt fordon osv...

**Tips:** Efter att ni skrivit era nya modeller och migrerat till databasen är det en god idé att scaffolda en ny fordonscontroller med ett annat namn (t.ex. Fordon2Controller) för att använda som referens när ni bygger ut era gamla vyer med dropdowns och relationer.