

## Feladatleírás

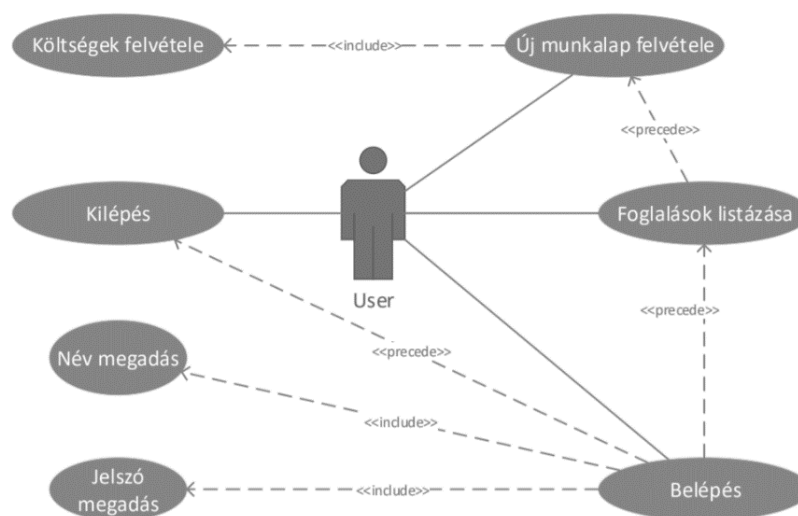
### Autószerviz

2. részfeladat: a szerelők a munkalapokat az asztali grafikus felületen keresztül adminisztrálják.

- A szerelő bejelentkezhets (azonosító és jelszó megadásával) a programba. Sikeres bejelentkezést követően látja a rá vonatkozó foglalásokat (az aktuális dátumtól), illetve kijelentkezhets.
- Bejelentkezést követően listázódnak (a munkalappal nem rendelkező) foglalások (időpont, foglaló neve, kategória). Egy foglalást kiválasztva új munkalap nyitható (a partner adatai és az időpont automatikusan áttöltődnek).
- A munkalapra tetszőleges számú munkafolyamat, illetve alkatrész vihető fel kiválasztva az előre megadott adatok alapján. Ezeket törölni is lehet a felvitel után. A munkalapon látható a végösszeg, amely az egyes tételek hozzáadásával/törlésével változik. Végül a szerelő véglegesítheti a munkalapot (ehhez a program kérjen megerősítést).

### Elemzés:

- A feladatot asztali részét egy háromablakos alkalmazásként Windows Presentation Foundation grafikus felülettel valósítjuk meg.
- A webszolgáltatást egy ASP.NET Core WebApi adja.
- A szoftver indítása után szükséges a felhasználónév és jelszó megadása.
- Ha sikerült a bejelentkezés, listázódnak a szerelő, munkalappal még nem rendelkező foglalásai.
- Egy foglalást kiválasztva munkalap készíthető hozzá.
- Ekkor egy új ablak nyílik meg, ahol hozzáadhatóak az egyes költségek.
- A munkalap ezután megerősítést követően menthető. Ezután az ablak eltűnik és újra listázódnak a foglalások.

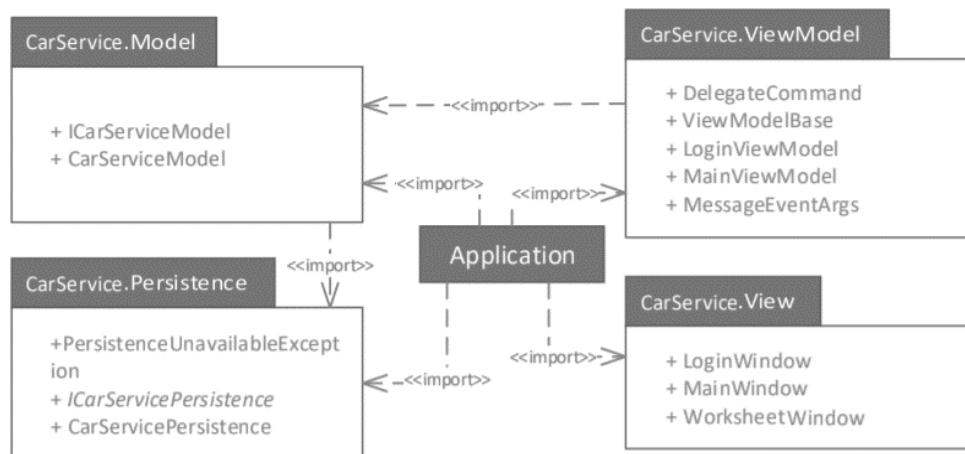


Felhasználói esetek diagramja

# Tervezés:

## 1. Programszerkezet:

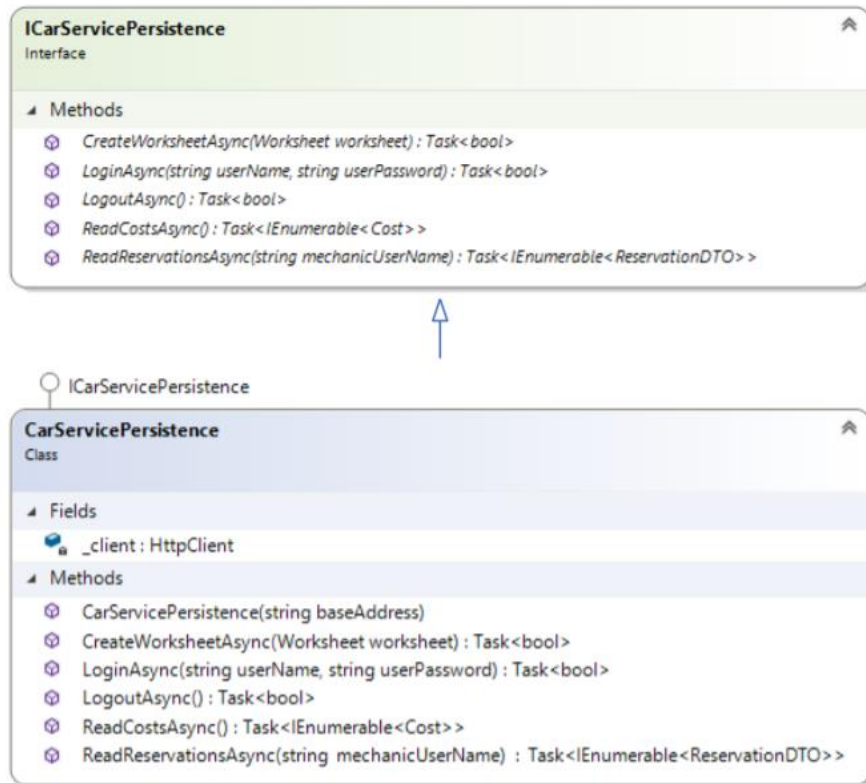
- a. A programot MVVM architektúrában valósítjuk meg. A megjelenítés a CarService.View, a modell a CarService.Model, a nézetmodell a CarService.ViewModel, míg a perzisztencia a CarService.Persistence névtérben helyezkedik el. A program környezetét az alkalmazás osztály (App) végzi, amely példányosítja a modellt, a nézetmodellt és a nézetet, biztosítja a kommunikációt, valamint felügyeli az adatkezelést.



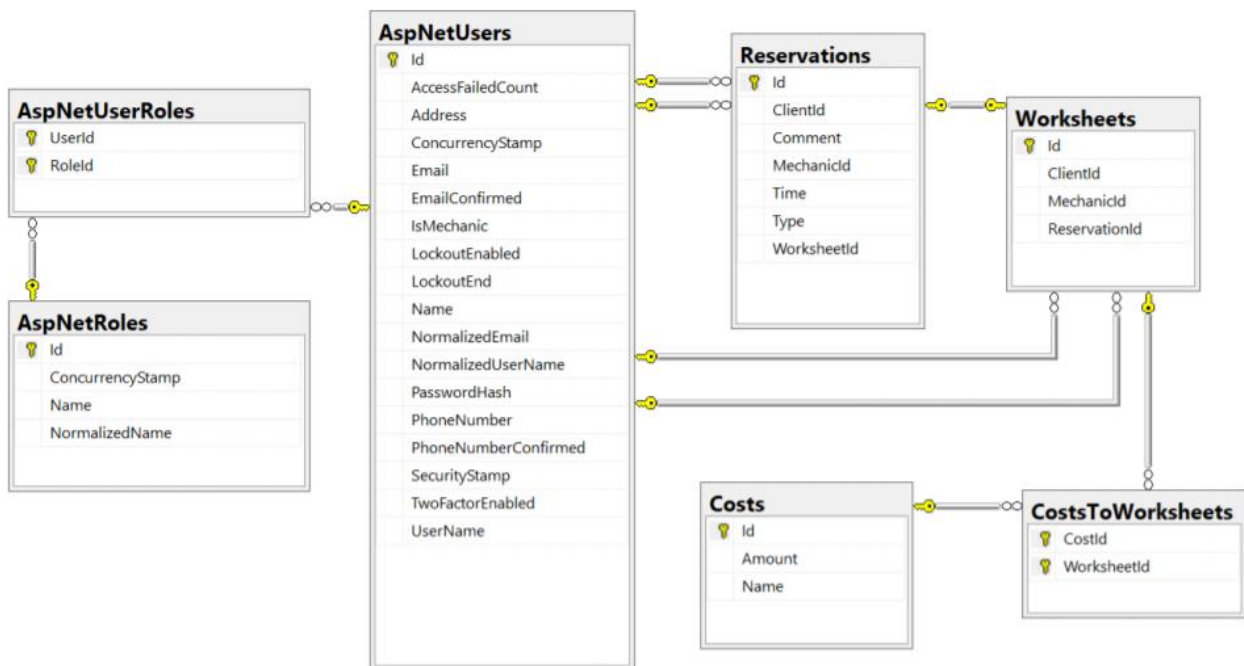
Program csomagszerkezete

## 2. Perzisztencia

- a. Az adatelérés feladata a webszolgáltatással és azon keresztül az adatbázissal való kommunikáció.
- b. Ezen feladatok végrehajtásának lehetőségét az ICarServicePersistence interfész adja meg: biztosítja a be- és kilépést, a foglalások és költségek lekérdezését valamint a munkalap mentését.
- c. Az előző pontban említett interfészt a CarServicePersistence osztály implementálja és valósítja meg. Ez az esetleges hibákat a PersistenceUnavailableException kivétellel jelzi.
- d. Az adatok egy lokális, MySQL adatbázisban tárolódnak.
- e. A tesztelés elősegítéséhez a rendszer automatikusan létrehoz néhány teszt felhasználót.



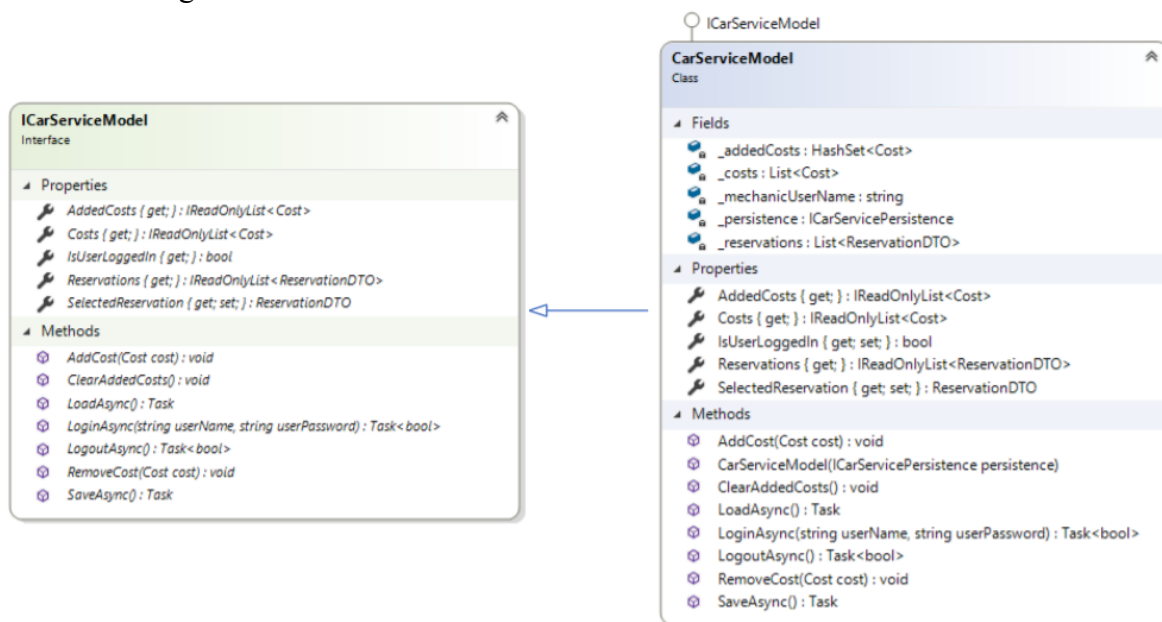
A perzisztencia osztálydiagramja



A perzisztencia egyedkapcsolati diagramja

### 3. Modell

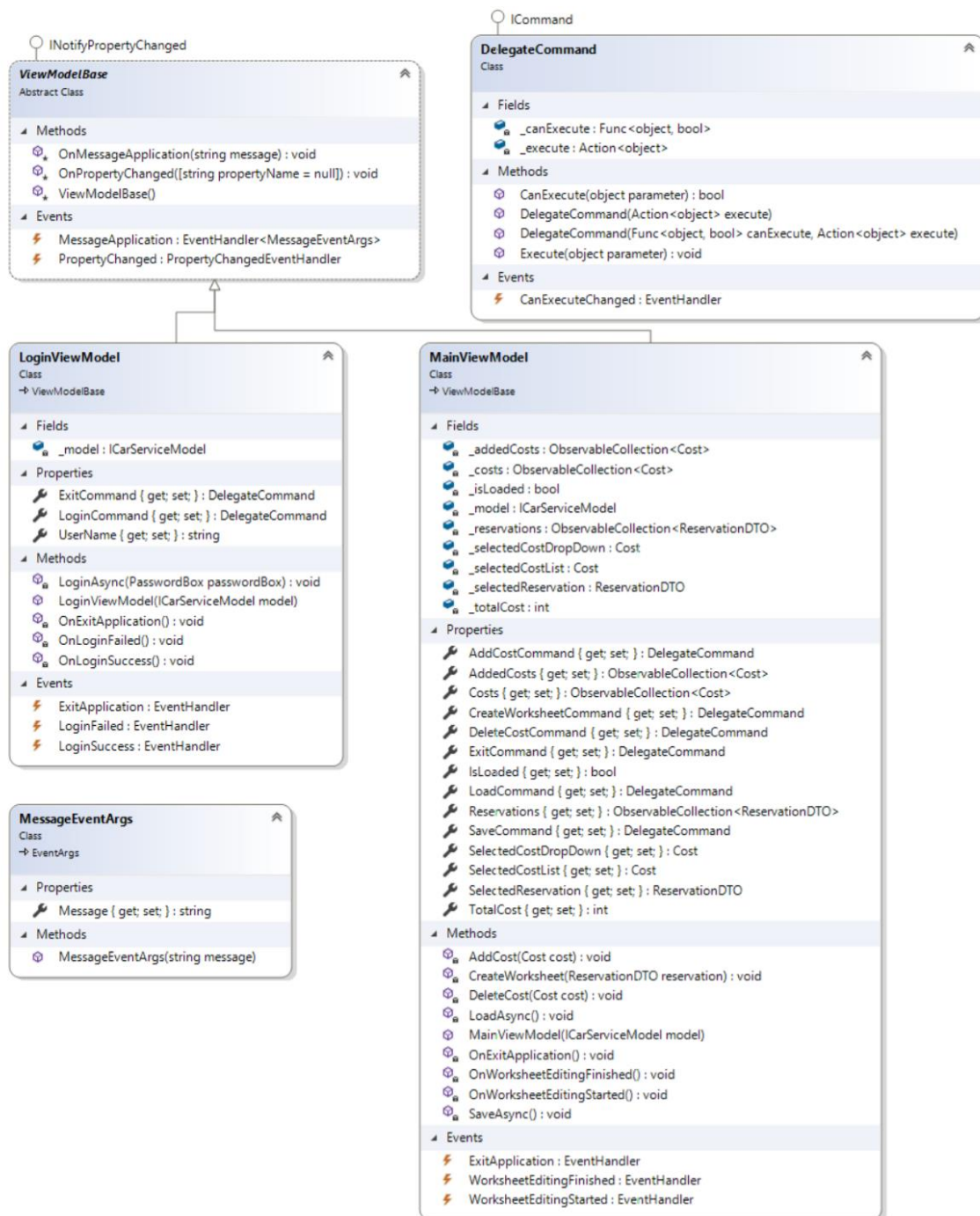
- A modell lényegi részét a CarServiceModel osztály valósítja meg, mely tárolja a foglalásokat, a bejelentkezett szerelő nevét, a munkalaphoz adott költségeket és a hozzáadható költségeket
- A függőségi befecskendezés használatához ki lett emelve belőle egy interfész, az ICarServiceModel.
- A modell lehetőséget biztosít költség hozzáadására, törlésére, a foglalások betöltésére, a be- illetve kijelentkezésre, valamint a kész munkalap elmentésére.
- A modell példányosításkor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad az adatok elérésére.



A modell osztálydiagramja

### 4. A nézetmodell

- A nézetmodell megvalósításához felhasználunk egy általános utasítás (DelegateCommand) osztályt.
- A nézetek a ViewModelBase absztrakt osztályból származnak le.
- A nézetmodell feladatainak javát a MainViewModel osztály látja el, amely parancsokat biztosít a fentebb már említett funkciók végrehajtásához, valamint a kilépéshez. A parancsokhoz eseményeket kötünk, amelyek a parancs lefutását jelzik a vezérlőnek. A nézetmodell tárolja a modell egy hivatkozását (model), de csupán információkat kér le tőle. Direkt nem avatkozik az üzleti logikába.
- A belépést a LoginViewModel nézetmodellek keresztül lehet végrehajtani. Rossz adatok, vagy elérhetetlen webszolgáltatás esetén hibát jelez.



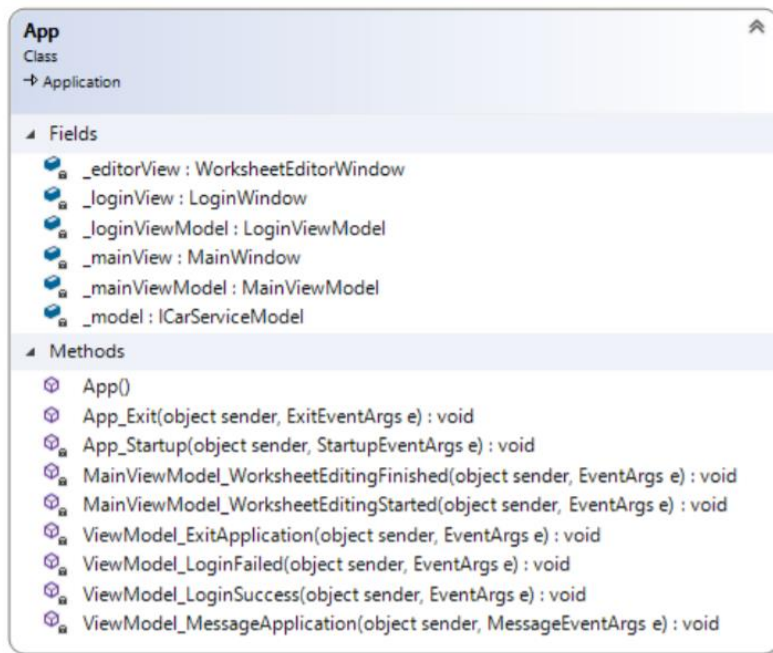
A nézetmodell osztálydiagramja

## 5. Nézetmodell

- A nézet három képernyőt tartalmaz, a MainWindow, a LoginWindow, valamint az EditWorksheetWindow osztályt.
- Minden adatot adatkötéssel kapcsolunk a felülethez, továbbá azon keresztül szabályozzuk a plusz információk megjelenítését is.
- A felületen elhelyeztünk egy menüsort a megfelelő elemekkel és a hozzájuk tartozó Commandokkal.
- Egy mentés megerősítéséhez egy felugró dialógusablakot használunk.

## 6. Környezet

- a. Az App osztály feladata az egyes rétegek példányosítása (App\_Startup), összekötése, a nézetmodell, valamint a modell egyes eseményeinek lekezelése, és ezáltal a modell, az adatkezelés, valamint a nézet szabályozása.



Az applikáció osztálydiagramja

## A Webszolgáltatás végpontjai és tesztelésük

### 1. api/costs

- a. HttpGet
- b. Visszaadja az adatbázis összes költségét, melyek hozzáadhatóak a munkalapokhoz.

### 2. api/reservations/{mechanicUserName}

- a. HttpGet
- b. Visszaadja egy szerelő összes, munkalappal még nem rendelkező foglalását.

### 3. api/Worksheets

- a. HttpPost
- b. Egy munkalap elmentésére használható.