

1. Beadandó feladat dokumentáció

Készítette:

Pánczél Róbert

EY2P3W

Feladat:

Készítsünk egy adminisztrációs felületet, amely képes az egyes szervizek által nyújtott szolgáltatásokat, és a szolgáltatásokra történő időpontfoglalásokat kezelni.

A webes felületen listázhatók az egyes műhelyek, és időpont foglalható az egyes szolgáltatásokra.

- Minden műhely az alábbi szolgáltatás közül tudjon valamennyit: Abroncs csere, klímátöltés, defekt javítás, futómű állítás.
- A műhelyek ABC sorrendben jelenjenek meg.
- Legyen lehetőség a listában cím vagy név megadásával keresni.

Az ügyfél regisztrálhat a felületen, az alábbi adatok megadásával: név, email, jelszó, jármű típusa (szöveges), jármű rendszáma (3 betű – 3 szám).

- Ha valamelyik adat nem kerül kitöltésre, vagy nem megfelelő a formátuma, akkor azt hibaüzenetben jelezni kell a felhasználó számára

Az ügyfél bejelentkezhetsz a felületre a regisztrációkor megadott emaillel és jelszóval.

Bejelentkezés után egy adott műhelyt és annak egy szolgáltatását választva megtekinthetők a szabad időpontok egy hétre előre .

- Az egyszerűség kedvéért minden műhely 8:00-kor nyit és 17:00-kor zár, illetve minden időpont hossza fél óra.
- Azok az időpontok, amelyekre már történt foglalás ne jelenjenek meg a felületen
- A felületen lapozással legyenek elérhetők még egy hétre előre az elérhető időpontok.

A szabad időpontok közül választva az ügyfél lefoglalhatja az adott időpontot. A foglalás véglegesítéséhez legyen megerősítés szükséges.

A felületen bejelentkezés után jelenjenek meg az ügyfél által lefoglalt időpontok az alábbi adatokkal:

- Melyik szervíz
- Mettől – Meddig (Nap, óra, perc)
- Milyen szolgáltatásra

Elemzés:

A weboldalt úgy kell felkészíteni, hogy egyszerre több felhasználót is ki tudjon szolgálni. A feladatot az ASP.NET Core MVC keretrendszerén belül valósítom meg.

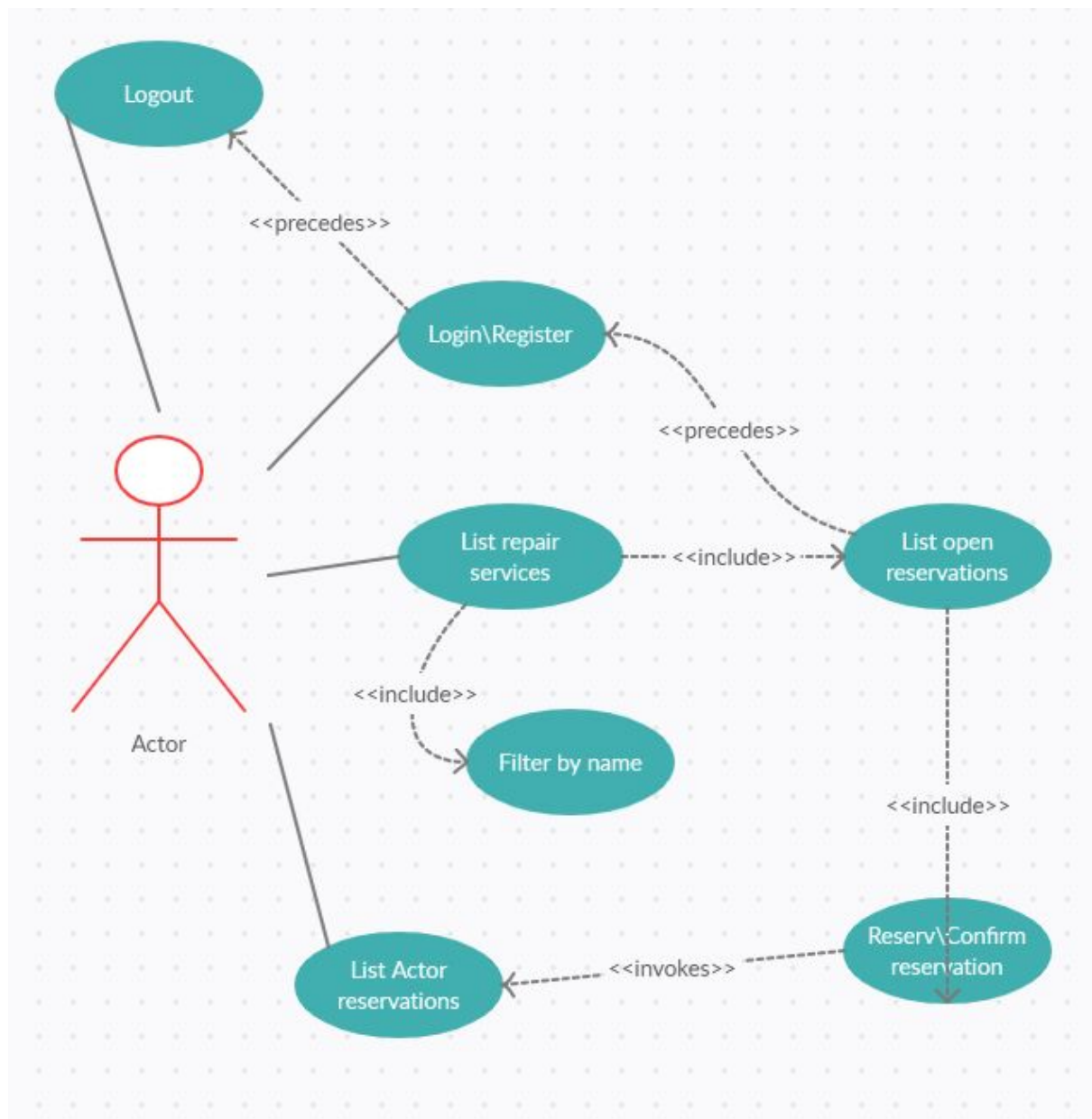
Mivel szükség van arra, hogy a felhasználók be tudjanak jelentkezni, ezért törekedem a felhasználók biztonságos adatainak a tárolására és kezelésére. Ezt a szerveren keresztül

hajtom végre az adatbázis segítségével. Egyes funkciókat csak bejelentkezés után érhető el a felhasználó.

A weboldal kezdőlapján, ha a felhasználó bejelentkezett megjelenítem az ő általa lefoglalt időpontokat, azt hogy hova foglalta le és milyen típusú javításról van szó. A fejlécben elérhető a műhelyek listája. Erre kattinva, megjelenítem azokat, minden adatukkal együtt. Itt a felhasználónak lehetősége van a műhelyek szolgáltatásai közül választani. Ahhoz hogy valamelyik lehetőséget ki tudja választani bejelentkezésre van szükség előtte.

Ezután, a választott műhely és javítási típus összes lehetséges időpontját kilistázza a program egy hétre előre.

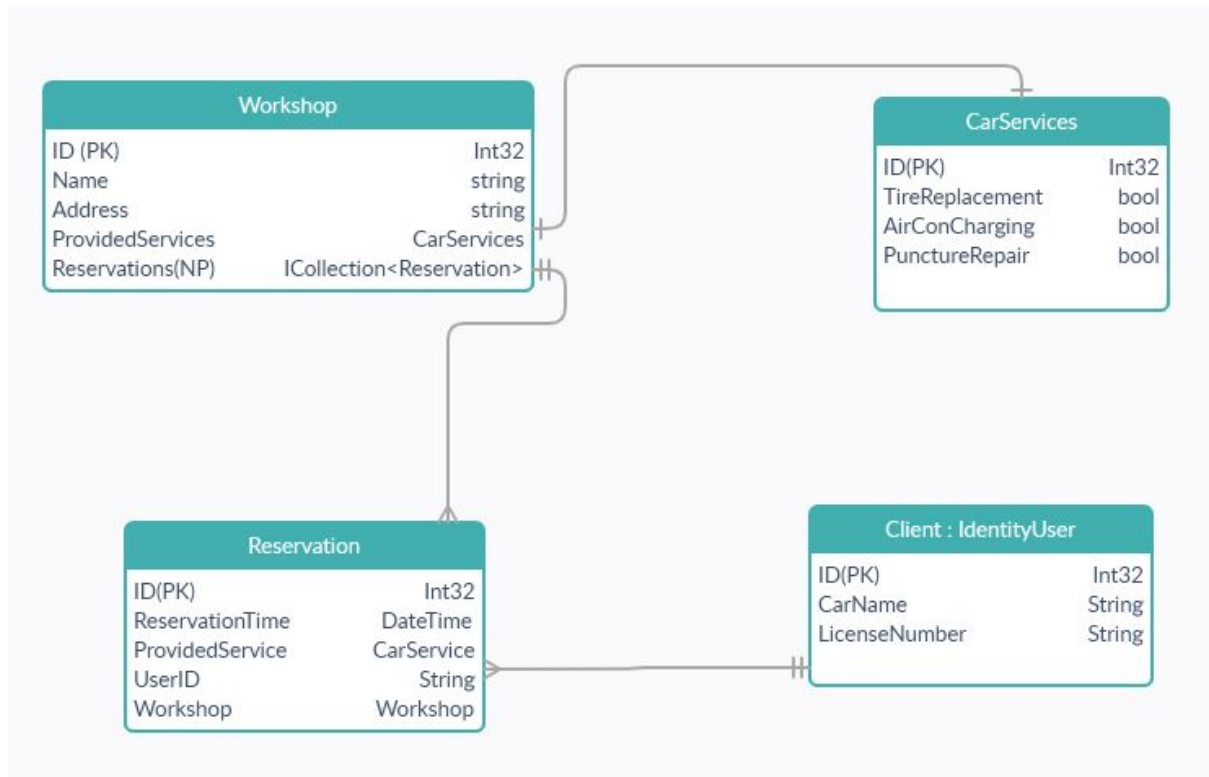
Ha a felhasználó kiválasztott, a számára megfelelő időpontot, akkor egy megerősítés oldalra irányítjuk át, ahol újra ellenőrizheti hogy megfelelő helyet és időpontot foglalt-e le. Igazolás után, elmentjük az új időpontot a Reservations táblába majd a felhasználót átirányítjuk a kezdőlapra.



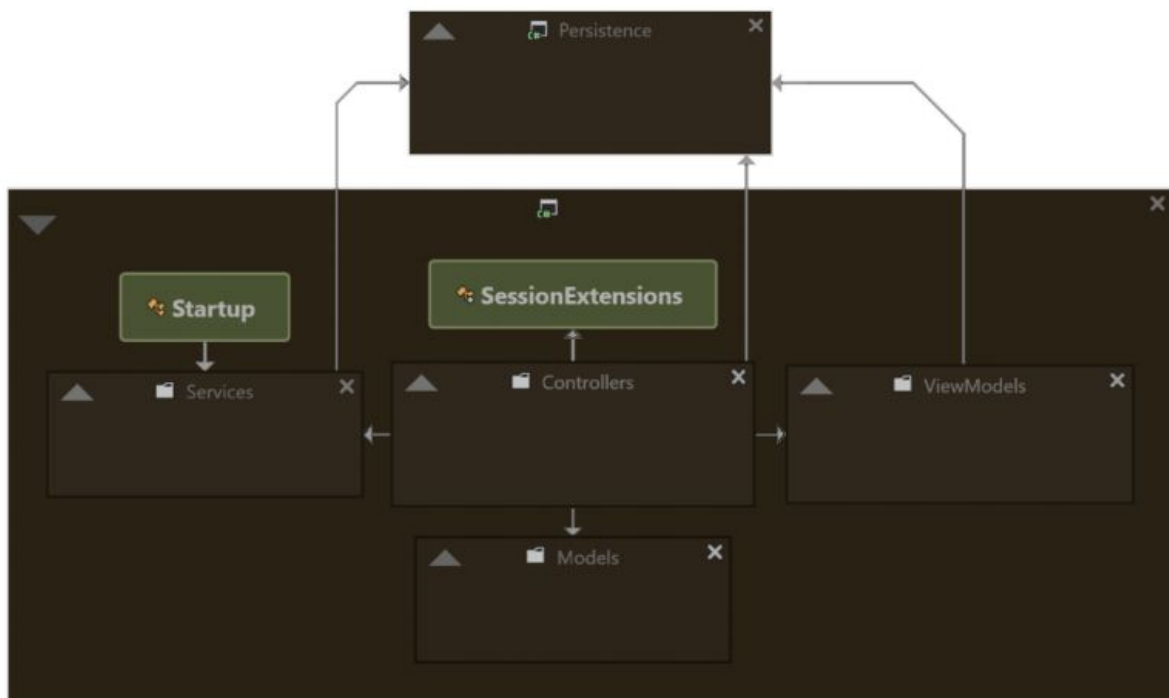
Felhasználói esetek diagramja

Tervezés:

Adatbázis szerkezete:



A programot négyrétegű architektúrában valósítom meg. A megjelenítésért a Views, a modell a Model, a nézetmodell a kettőt köti össze ViewModel, míg a perzisztencia a Data-ban helyezkedik el.



Csomag szerkezet

Nézet:

Az asp.net mvc keretrendszerben a nézet egy felhasználói felületet definiál. A nézet, megjeleníti a model adatait a felhasználó számára és lehetővé teszi, hogy ezeket módosítsa. A nézet az asp.net mvc-ben HTML-ből, CSS-ből és a Razor syntax-ből tevődik össze és teszi lehetővé, hogy könnyen tudjon a modell a kontrollerrel kommunikálni. A nézetben található a Clients nézet, mely a felhasználó bejelentkeztetéséért és regisztrációjáért felelős. A Home nézet, a default kezdőlapnak leírását tartalmazza. A Reservations Index nézete, megjeleníti az egy autós bolthoz tartozó időpontokat, míg a Create az foglalás megerősítésére szolgál. A Workshops nézet feladata, kilistázni a autó javító szervízeket.

Nézetmodell:

A nézetmodell, az az adatmodell amit meg szeretnénk jeleníteni a felhasználó felé a nézetben. Ha például több modellt szeretnénk megjeleníteni egyetlen nézetben akkor a nézetmodellt kell átadnunk a nézetnek.

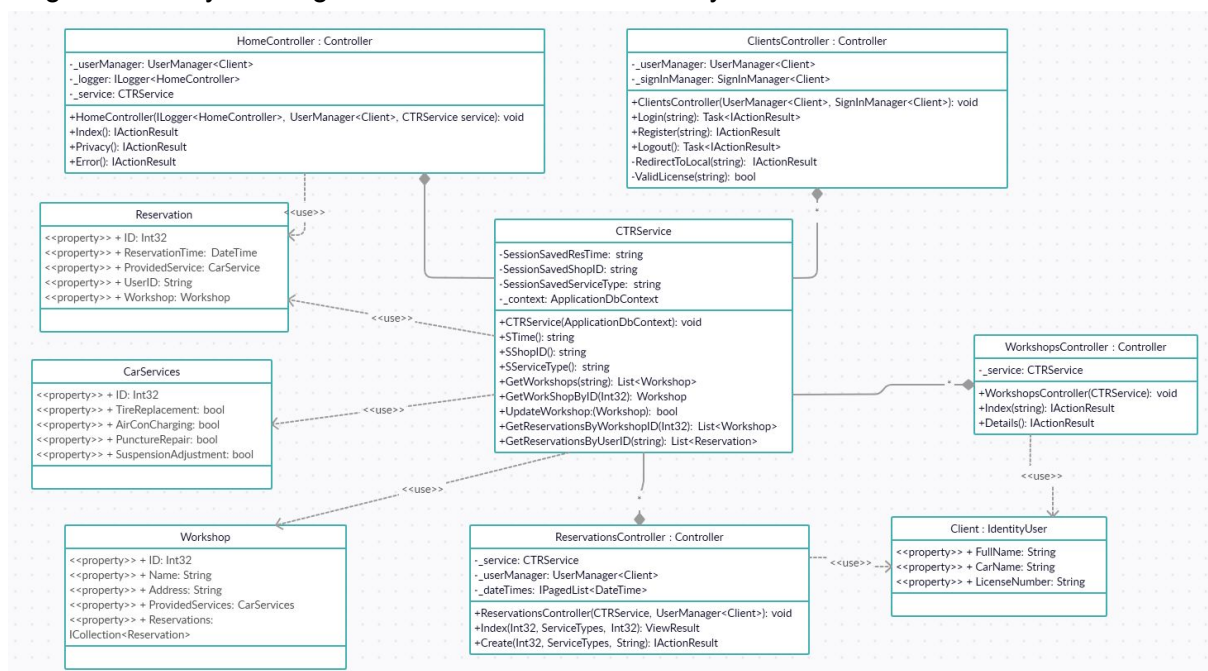
Modell:

Az mvc keretrendszerben, a modellben helyezkednek el az osztályok, azaz meghatározza az adatok formáját, elemeit és a biznisz logikához tartozó metódusokat. A modellben helyezkedik el a Client osztály amely az Identity osztályból van származtatva. Ennek köszönhetően a felhasználó adatai biztonságosan lesznek eltárolva. A Reservation osztály meghatározza a foglalások formáját. Minden foglaláshoz tartozik egy ID, a foglalás

időpontja, javítás típusa továbbá a felhasználó aki foglalta és a bolt ahol foglalva lett. A Carservices osztály meghatározza, hogy az egyes boltokhoz milyen javítási típus tartozik. A Workshop osztály, a műhelyek adatainak formáját határozza meg.

Perzisztencia:

Feladata az adatok biztonságos tárolása Code First módon. 4 fő csoportot tárolok: Workshop, Carservices, Reservation, Client. Mindegyik rendelkezik egyedi azonosítóval. A Workshop tartalmaz egy Reservations listát, tehát hozzárendeli a létező foglalásokat egy-egy szervízhez. Illetve, mindegyik szervízhez egy-egy Carservice-t rendel hozzá, amiből megtudható milyen szolgáltatások vannak az adott helyen.



Az alkalmazás osztálydiagramja