

Haladó .NET Programozás

Tömegközlekedés-követő rendszer

Rendszerterv

A program komponensei, és azok felelősei:

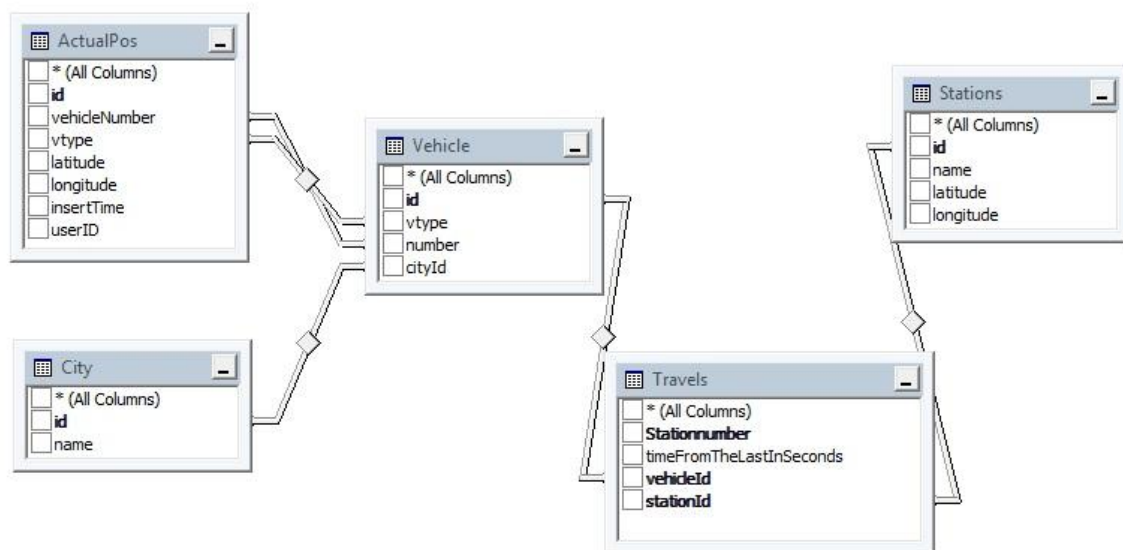
- MS SQL Server adatbázis: közös munka
- WCF Service-ek Gergő
- Windows Phone kliens: Gábor
- Webes felület: Gergő

A Program specifikációja, felépítése:

1. Adatbázis

Az alkalmazás mögött egy MS SQL Server adatbázis van. Itt tárolunk minden információt, a járatokat, azok megállóit, a felhasználók aktuális pozícióit stb.

Az adatbázistáblák felépítése a következő ábrán látható:



1. ábra Az adatbázis szerkezete

Táblák adatai:

- ActualPos: Itt tároljuk a felhasználók aktuális pozícióit. Ennek segítségével aktualizáljuk a megállók közti várható menetidőt. Minden felhasználó kap egy saját id-t és ezzel tudja az új pozíciókat a WCF service-en keresztül beszúrni, és ezen id alapján különböztetjük meg a felhasználókat.
- City: Városok listája
- Vehicle: Egy adott járat információit tároljuk itt
- Stations: Megállók adatai. Külön oszlopot tartunk fenn a megálló nevére, így, ha az adott megállót átnevezik, akkor csak egyetlen mezőt kell átírnunk, és máris érvényesek az új elnevezések
- Travels: Adott meneteket tartunk számon ebben a táblában

2. WCF Service

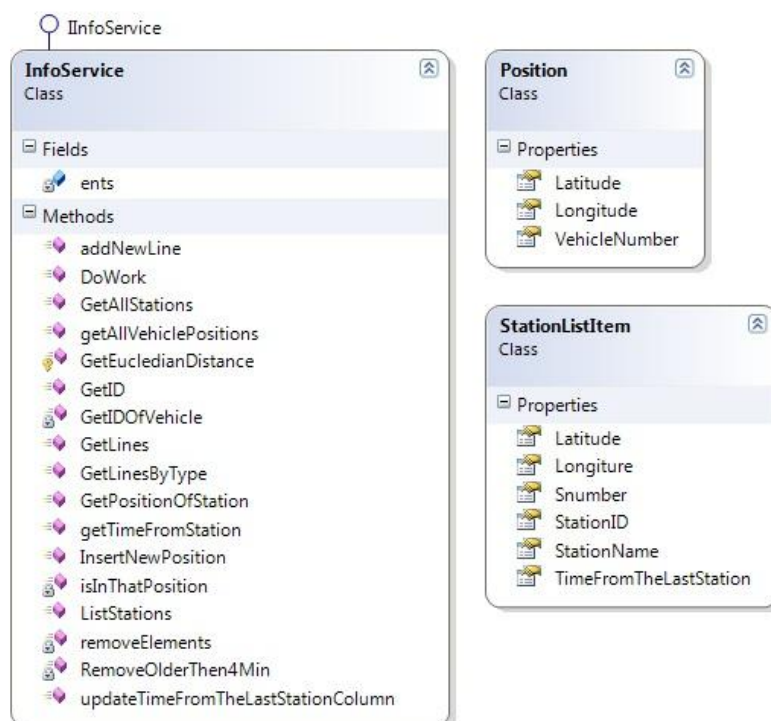
Az üzleti logika megvalósításához Windows Communication Foundation-t használunk. Az adatbázist is ezen keresztül érjük el. Az táblák objektumokká képzéséhez Entity Framework-öt használunk.

A szolgáltatásokat az InfoService osztály következő metódusain keresztül biztosítjuk (Be és Kimeneti paraméterek részletezése):

- `void InsertNewPosition(int VNumber, string Vtype, string userid, float lat, float lon)`
Új pozíció beszúrása
 - `int` Vnumber: Az adott járat száma (pl. 6)
 - `string` Vtype: Az adott járat típusa (pl. „V”, mint Villamos)
 - `string` userid: Az adott felhasználó id-ja (ezt a GetID metódussal lehet generáltatni)
 - `float` lat, `float` lon: Az adott pozíció szélességi és hosszúsági koordinátái
- `List<StationListItem> ListStations(int vNumber, string Vtype)`
Egy adott járat megállóinak listáját adja vissza. A StationListItem osztály egy adott megálló adatait tartalmazza (részletek a következő szekcióban lévő UML diagramon)
 - `int` Vnumber: Az adott járat száma (pl. 6)
 - `string` Vtype: Az adott járat típusa (pl. „V”, mint Villamos)
- `String GetID(float lat, float lon)`
Ez a metódus adja a felhasználók ID-ját. Minden egyes trackeles kezdetekor szükséges meghívni.
 - `float` lat, `float` lon: Az adott pozíció szélességi és hosszúsági koordinátái
- `int getTimeFromStation(int vNumber, string vType, float lat, float lon)`
Egy adott megállóhoz megmondja mp-ben, hogy mennyi idő múlva fog odaérni a következő járat.
 - `int` Vnumber: Az adott járat száma (pl. 6)
 - `string` Vtype: Az adott járat típusa (pl. „V”, mint Villamos)
 - `float` lat, `float` lon: Az felhasználó pozíciójának szélességi és hosszúsági koordinátái
- `List<Position> getAllVehiclePositions()`
Az adatbázisban lévő összes járat pozíciójának listáját adja vissza. Fontos, hogy minden felhasználónál csak a legutóbb beszúrt pozíciókat listázzuk.

- `List<int> GetLines()`
Az adatbázisban lévő összes járat számának a listáját adja vissza.
- `List<int> GetLinesByType(String type);`
Egy típushoz tartozó összes járatot adja vissza
 - `String type`: A típus, amihez tartozó járatokat szeretnénk visszakapni
- `List<string> GetAllStations();`
Az adatbázisban lévő összes megállót adja vissza
- `Position GetPositionOfStation(string StationName);`
Egy megálló koordinátáit adja vissza
 - `string StationName`: A megálló neve
- `void addNewLine(List<StationListItem> stations, int VNumber);`
Egy új járat beszúrására szolgáló függvény
 - `List<StationListItem> stations`: A megállók listája
 - `int Vnumber`: A járat száma

A Service osztálydiagramja a következőképp néz ki:



2. ábra A WCF Service osztály UML diagramja

Az `InfoService` mellett van egy `TestInfoService` nevű WCF service is. Erről részletesebb leírás a tesztelés szekcióban olvasható.

3. WP7 Alkalmazás

A mobil alkalmazáson keresztül pozíciókat küldhetnek, és lekérhetik a legközelebbi megállóba érkező szerelvény hátralevő becsült idejét.

4. Webes Felület

Mivel a Windows Phone eszközök kijelzője elég kicsi, ezért úgy gondoltuk, hogy hasznos lenne, ha a felhasználók nagyobb kijelzős eszközökkel egy webes felületen további információkat kaphatnának a közlekedés aktuális állapotáról.

Ezt az igényt hivatott kielégíteni a PTTWebSite oldal.

Az adatok kinyeréséhez a WP7 apphoz hasonlóan az előző pontban ismertetett InfoService-t használjuk.

Az oldal az ASP.NET MVC framework segítségével készült. Ennek oka az, hogy az MVC Pattern remekül illeszkedik az alkalmazásunk felépítéséhez, hiszen a modellünk az InfoService WCF Service-el már adott volt, így gyakorlatilag csak a megjelenítést, és az oldal struktúráját kellett kitalálni és megírni.

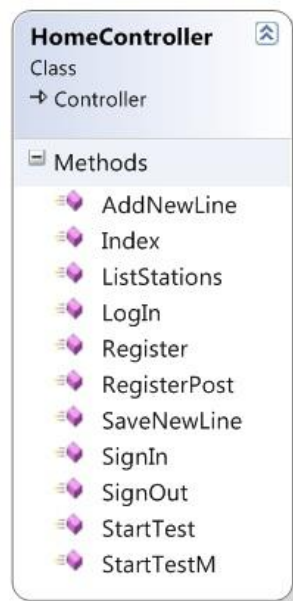
Az oldal funkciói a következők:

- Minden felhasználó látja egy Bing Maps térképen az összes járat aktuális állapotát
- Ki lehet rajzoltatni egy térképre a megállók listáját
- A bemutatás kedvéért bárki egy gombnyomással elindíthat egy tesztet, amivel egy villamos vonalán történő utazást követhetünk nyomon a térképen
- A felhasználók regisztrálhatnak az oldalra
- Regisztrált felhasználók saját járatokat tölthetnek fel az adatbázisba (megállóadatokkal együtt)

A felhasználók kezeléséhez az ASP.NET Membership funkciót használjuk. A belépéshez szükséges adatokat a saját adatbázisunkban tartjuk számon (tehát nem használunk lokális adatbázist), amit a aspnet_regsql.exe segítségével konfiguráltunk.

Az oldalról gyakran hívjuk asszinkron módon a WCF service-t, ehhez JQuery-t használunk.

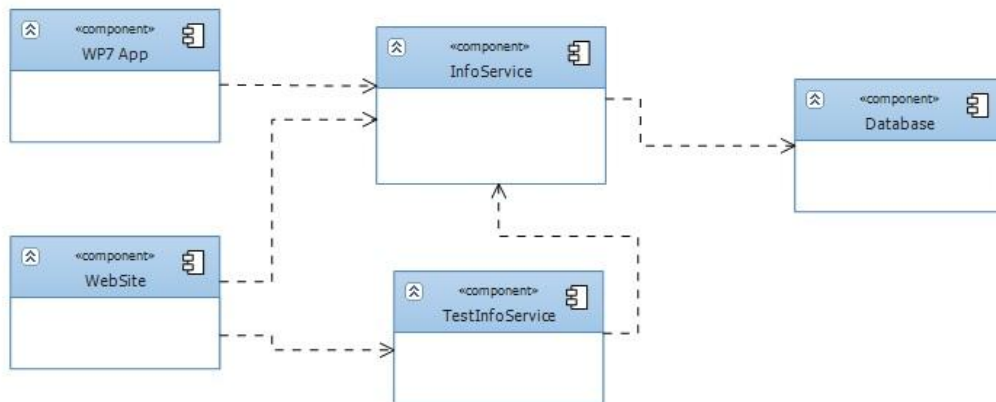
Az MVC alkalmazás Controllerjének UML osztálydiagramja a következő módon néz ki:



3. ábra A Controller metódusai

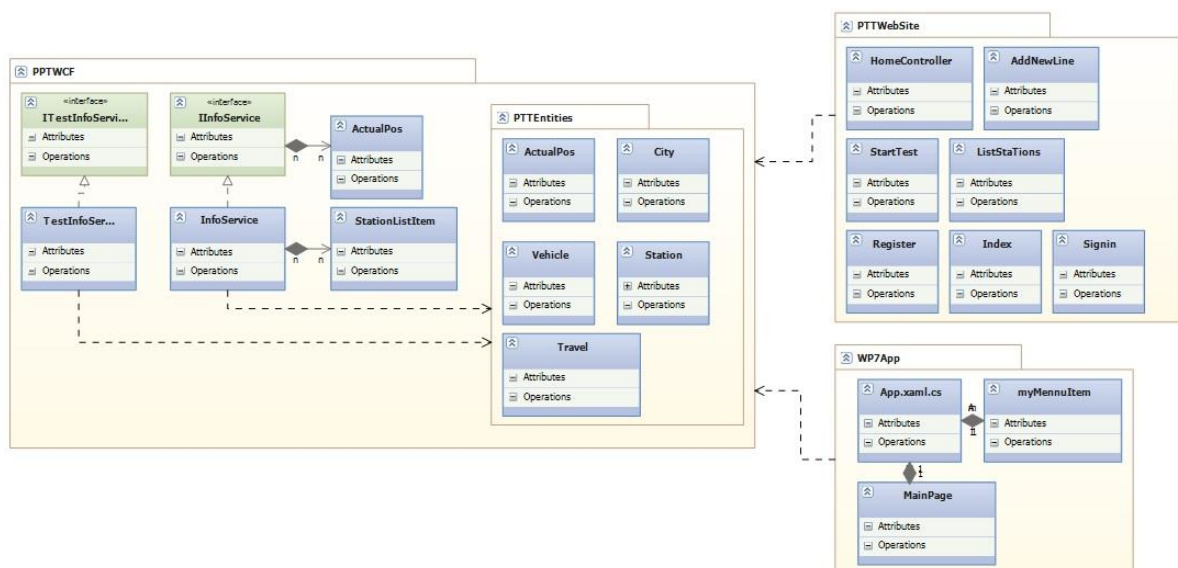
A Rendszer felépítése:

A következő ábrán a rendszer felépítését látjuk, és azt, hogy az egyes komponensek hogyan kapcsolódnak egymáshoz.



4. ábra A fő komponensek

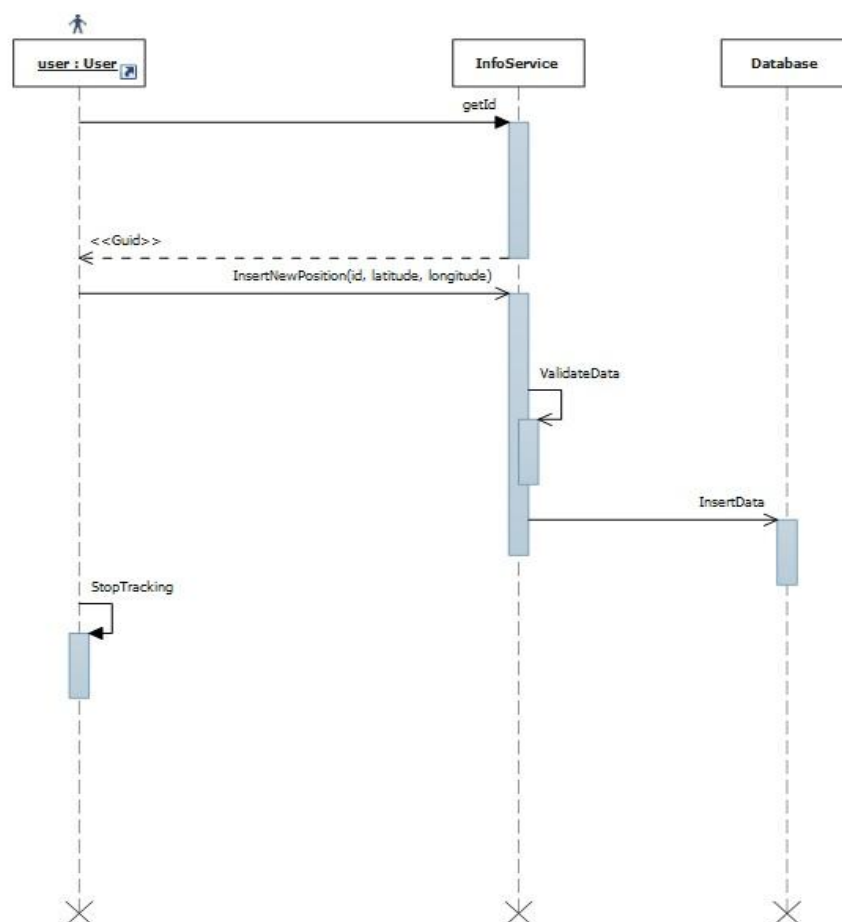
Az UML csomagdiagram pedig a következőképp néz ki:



5. ábra Csomagdiagram

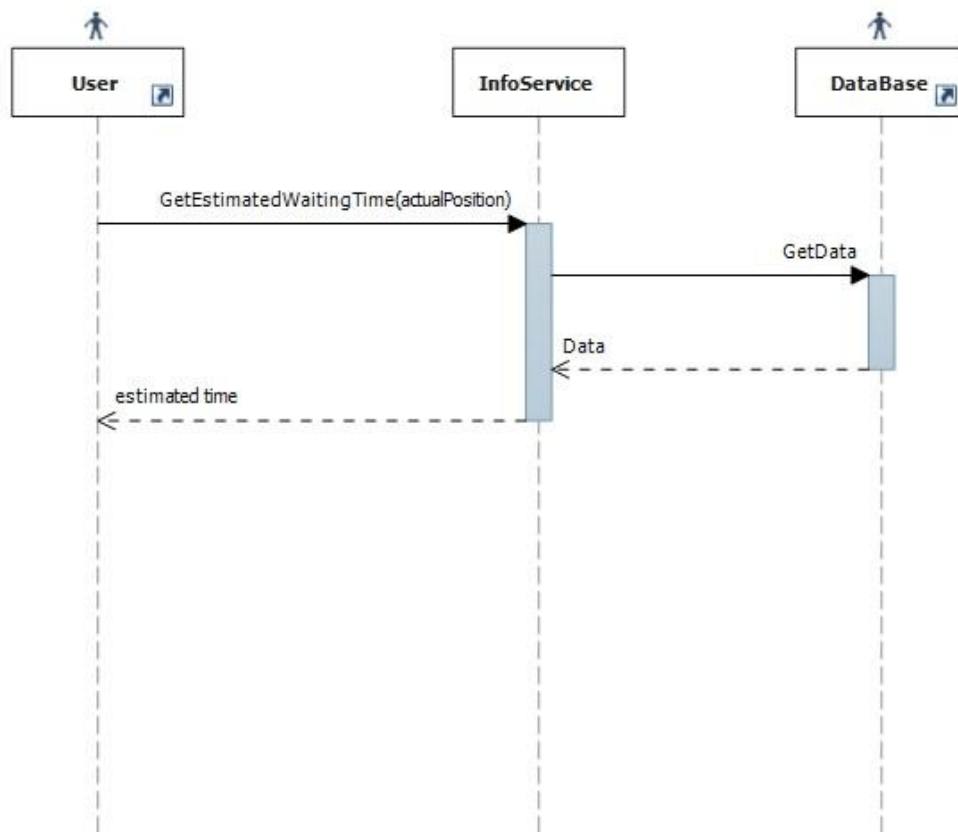
A rendszer működési folyamata, és állapotai:

Amikor a felhasználó felszáll egy járatra, akkor a telefonján beállítja, hogy trackelje az aktuális pozíciókat. Ehhez a telefon először egy ID-t kér az InfoService-től. Ha ez az ID megérkezett, akkor ezzel a telefon bizonyos időközönként elküldi az aktuális koordinátákat. Ha a koordináták átjutnak a validáción (pl. nincsenek túlságosan messze a járat útvonalától stb.), akkor a WCF Service Entity Framework segítségével beleírja az adatokat az adatbázisba.



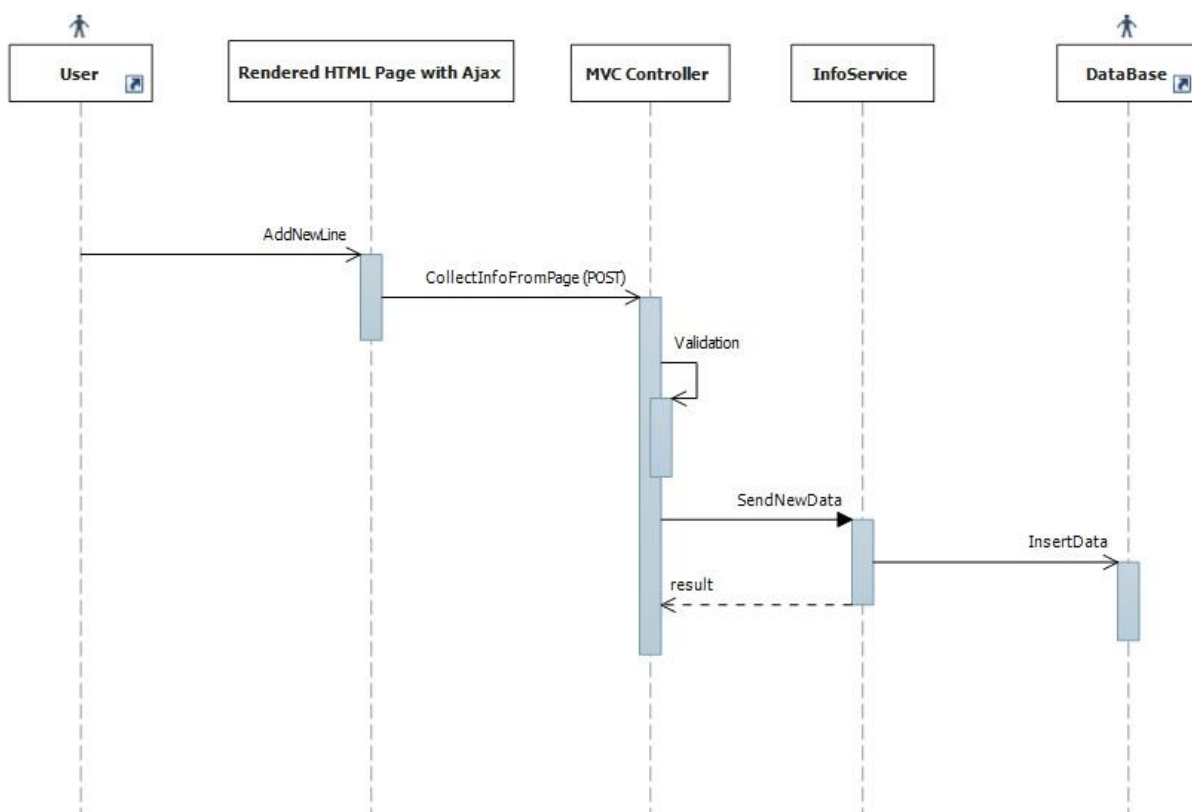
6. ábra Az utazás folyamata

A felhasználók telefonjaik segítségével nemcsak adatokat küldeni tudnak, hanem egy tetszőleges pontból meg tudják kérdezni a rendszertől, hogy a legközelebbi megállóba mennyi idő múlva érkezik a következő járat.



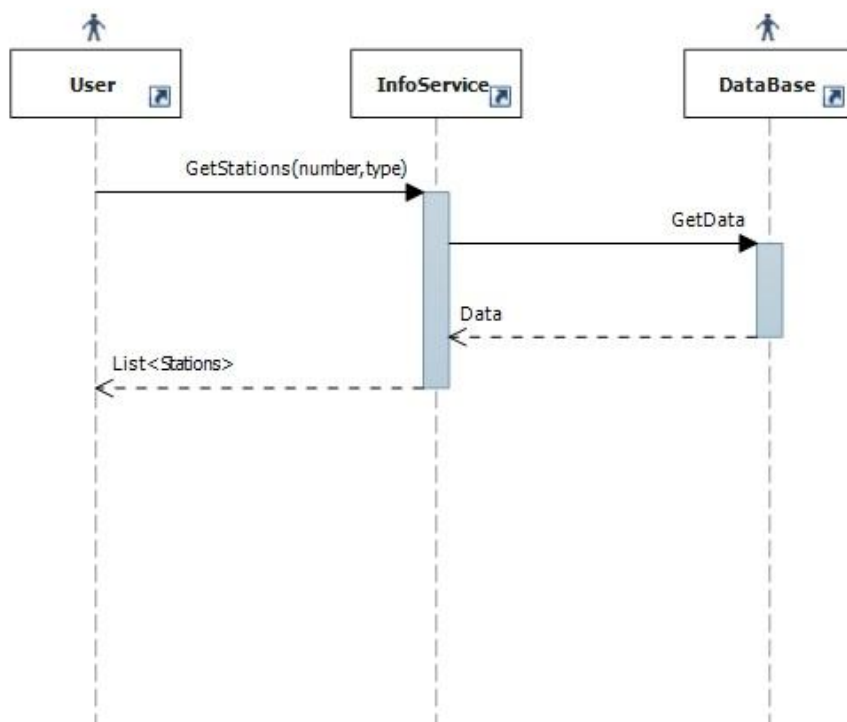
7. ábra Várakozási idő lekérése

A rendszerhez tartozó weboldalon a regisztrált és bejelentkezett felhasználók adhatnak új útvonalat az adatbázishoz. Ehhez először az ASP.NET MVC által kirenderelt html oldalon JQuery segítségével összeszerkeszthetik az útvonalat. Ha készen vannak, akkor a Save gombra kattintva HTTP POST segítségével elküldjük a szervernek az adatokat. Ezeket az adatokat a Controllerben feldolgozzuk, és ha a validáció sikeres (minden adatot megadott, a szélességi/hosszúsági számok valóban számok stb.), akkor az adatokat elküldjük az InfoServicenek, ami az adatbázisban letárolja az új információkat.



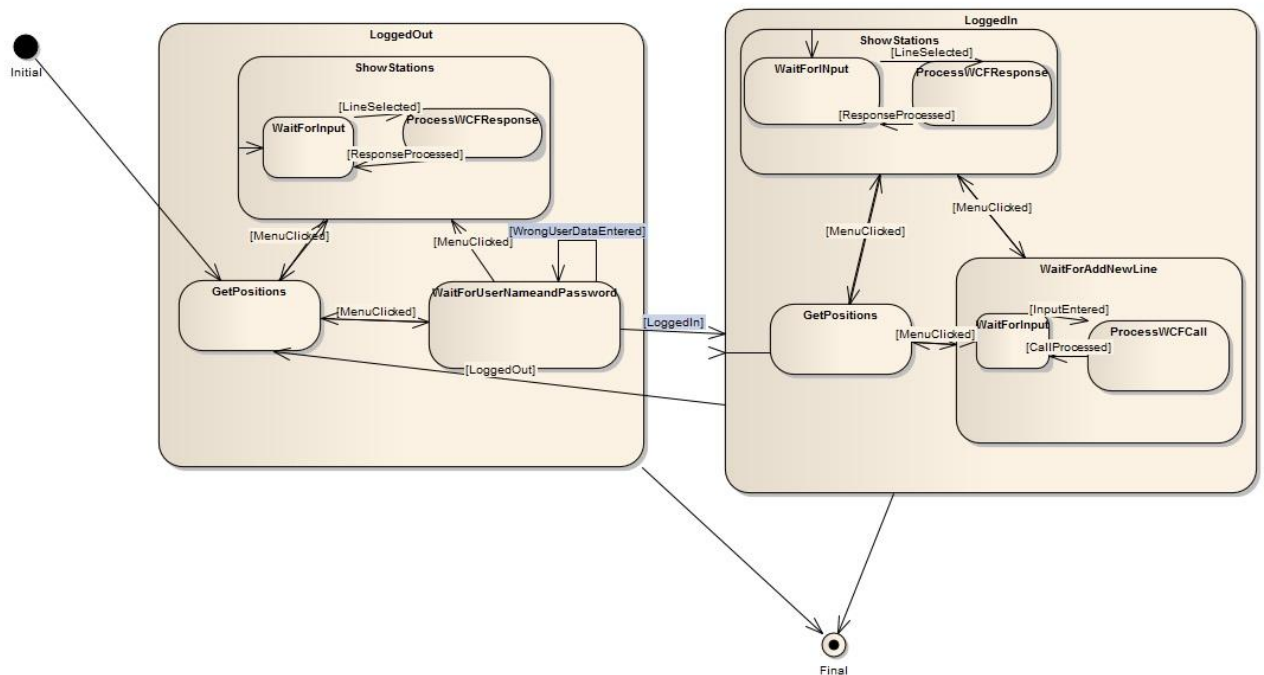
8. ábra Új járat hozzáadása

Továbbá a felhasználóknak lehetőségük van lekérni egy adott járathoz tartozó megállók listáját:



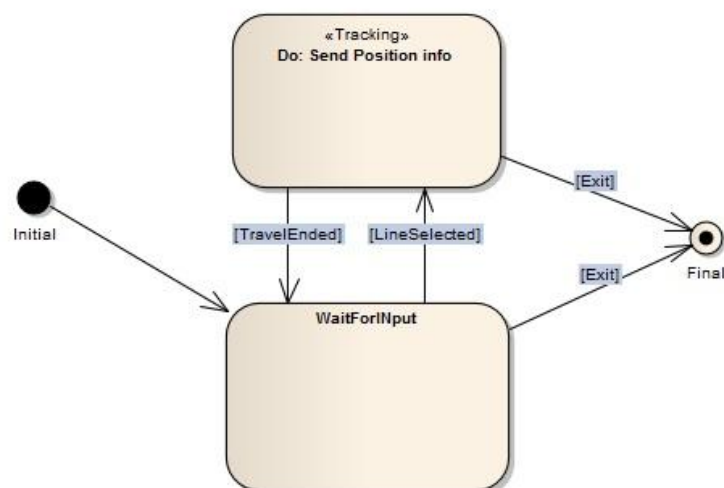
9. ábra Állomások listázása

A Weboldal elérhető mindenki számára, bárki lekérheti az aktuális közlekedési állapotokat, és a járatok megállóinak listáját. Azok a felhasználók, akik regisztráltak be is tudnak jelentkezni. Bejelentkezés után egy új menüpont jelenik meg, ahol további járatokat adhatunk az adatbázishoz.



10. ábra A Weboldal állapotdiagramja

A Windows Phone 7 alkalmazás alapvetően 2 állapotban lehet, vagy adatokat küld a szerver felé, azaz a felhasználó utazik egy villamoson, vagy pedig adatok bevitelére vár, és információkat jelenít meg.



11. ábra A WP7 alkalmazás állapotai