



Gyuris Bence

BP Menetrend

Windows 8.1 Universal App

Önálló laboratórium 2

Dokumentáció

konzulens

Albert István

2015 Ősz

Tartalom

1. Bevezetés.....	3
2. Követelmények.....	4
3. Alkalmazás architektúrája	5
3.1. Architektúra migrációja.....	6
4. Az asztali alkalmazás felülete	7
4.1 DefaultPage	8
4.2. ListPage.....	9
4.3. RoutePage	10
4.4. StopPage.....	11
4.5. PlanningPage	12
5. A telefonos alkalmazás felülete	13
5.2 Navigációs struktúra	14
5.3 Oldalak új kinézete	15
6. Térképek.....	16
7. Összegzés.....	19

1. Bevezetés

A feladatom a szakdolgozatom (<https://diplomaterv.vik.bme.hu/hu/Theses/Budapesti-menetrend-alkalmazas-keszítése>) továbbfejlesztése volt. A szakdolgozatom témájaként elkészítettem egy budapesti menetrend vastagkliens alkalmazást Windows Phone 8.0-ás platformra. A félév során, az Önálló laboratórium 2 tárgy keretében, az alkalmazás asztali számítógépre és tabletre optimalizált változatát kellett elkészítenem. Ezt egy Windows 8.1-es Universal App-ban kellett megvalósítanom.

Ez a cél három fő részre osztotta a munkámat:

- Migrálnom kellett a Windows Phone 8.0-ás projektet Windows 8.1 Universal App projektbe. Át kellett írnom a megváltoztatott API hívásokat, a nem támogatott függőségeket le kellett cserélnem.
- Át kellett alakítanom a szoftver architektúráját úgy, hogy elkülönüljenek a megosztott komponensek a telefon vagy asztali alkalmazás specifikus komponensektől. A cél az volt, hogy az eredeti alkalmazást felhasználva minél kevesebb új kódot kelljen írnom az asztali kliens megírása közben.
- El kellett készítenem az asztali/tabletes felületet, igazodva az itteni követelményekhez, megvalósítva a szükséges funkcionalitást és megjelenítését.

Ezen dokumentum során leírom, hogy hogyan oldottam meg ezeket a problémákat. Az elkészült szoftver megtalálható a dokumentáció mellett forráskód szinten a GlobalCityTransit nevű mappában, Visual Studio 2013-mas solution-ként, valamint bináris szinten .appxupload kiterjesztésű fájlokban.

A teszteléshez egy legalább Windows Phone 8.1-es OS-sel rendelkező developer unlocked készülék vagy emulátor és egy legalább Windows 8.1-es OS-sel rendelkező számítógép vagy tablet szükséges, valamint legalább Visual Studio 2013-mas fejlesztőkörnyezet.

2. Követelmények

Az alkalmazás feladata különböző használati esetekben segítenie a felhasználóját a közlekedése során, így kielégítve a felhasználói igényeket. Az alkalmazásnak internetelérés nélkül, Windows 8.1-es platformokon kell futnia. Az alkalmazásnak ki kell használnia az adott platform (asztali/mobil) előnyeit és figyelembe kell vennie annak korlátait. Az alkalmazásnak lehetőleg minél kevesebb felhasználói interakcióval kell minél nagyobb felhasználói elégedettséget elérnie, azaz magas fokú intelligenciára és önálló döntéshozatalra van szüksége, valamint biztosítania kell a testreszabhatóságot.

Az alkalmazás követelményeit néhány tipikus felhasználói eseten keresztül lehet a legjobban szemléltetni:

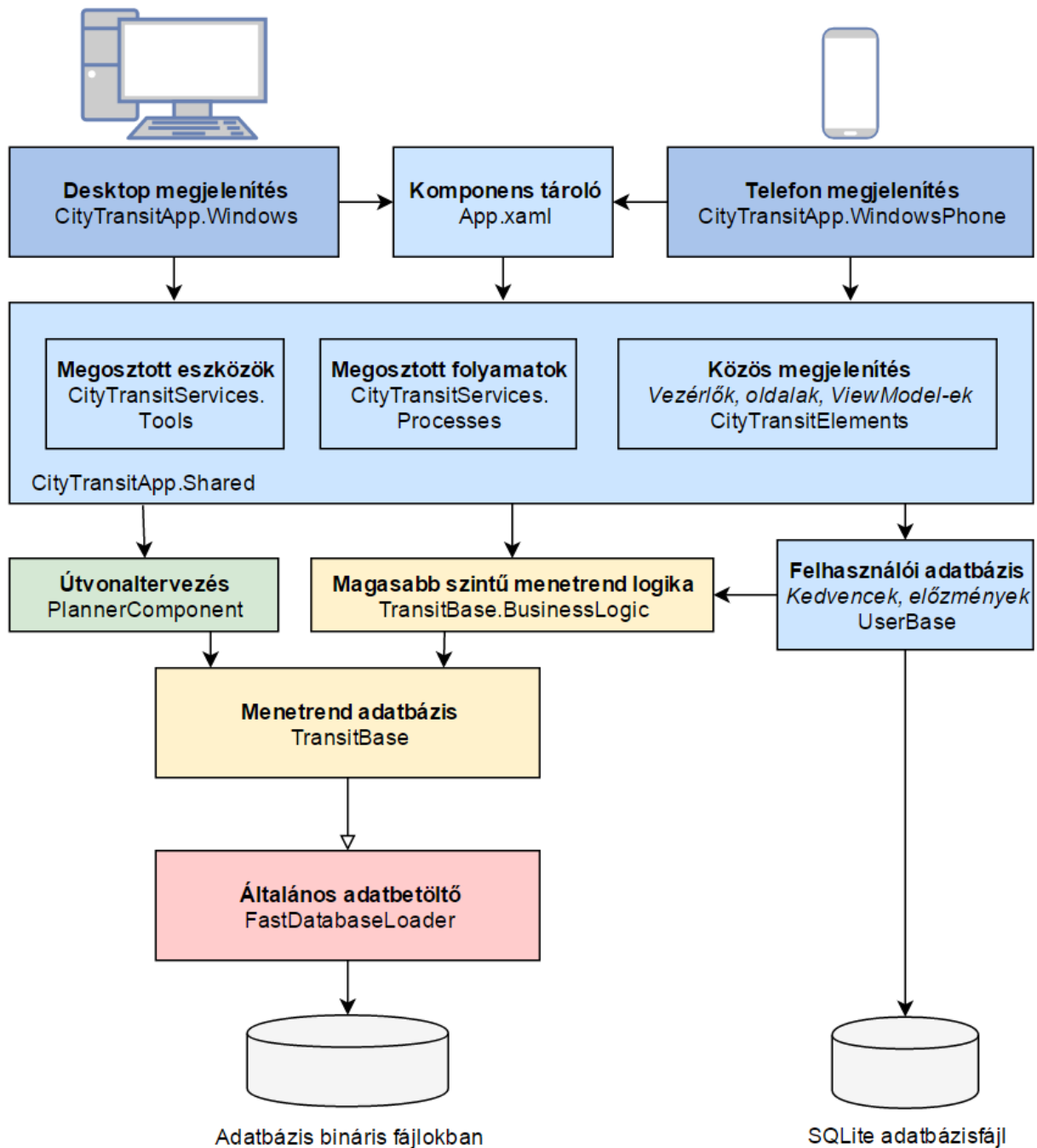
- A felhasználó el szeretne jutni valahova, kíváncsi az oda vezető legjobb útra.
- A felhasználó el szeretne jutni valahova egy időpontra, szeretné tudni, hogy mikor kell elindulnia, és hogyan fog oda eljutni.
- A felhasználó a rendszeresen használt járatainak a következő indulási idejeit szeretné tudni.
- A felhasználó tudni szeretné, hogy a járata egy megállójában milyen átszállási lehetőségei vannak.
- A felhasználó látni szeretné a kiválasztott megállóban, a kiválasztott járat, kiválasztott napon lévő menetrendjét és egyéb információkat.
- A felhasználó az utazását és az érintett megállókat szeretné térképen is látni a könnyebb tájékozódás érdekében.

Tipikus használati esetek mobil készülékeken:

- A felhasználó a megállóban vagy annak közelében van és kíváncsi arra, hogy mennyit kell várnia a következő járatra (járatokra), amivel utazni szeretne.

3. Alkalmazás architektúrája

Az alkalmazás új architektúráját a következő ábra szemlélteti. A színekkel jelöltem a szerelvényeket (dll) amelyekben az adott komponenst megvalósítottam.



Az ábrán a nyilak komponensek közötti függőségeket jelölnek. A komponens tároló egy UserBase, TransitBase és egy PlannerComponent-et hoz létre induláskor, majd tárolja el később. A megjelenítés ezen keresztül tudja elérni az alsóbb komponenseket.

3.1. Architektúra migrációja

Az architektúrában látható komponensek nagy része már készen volt az alkalmazás előző verziójában, csak újra kellett rendezni a kódot. A következő táblázatban foglaltam össze, hogy melyik komponensben milyen módosításokat kellett végrehajtanom.

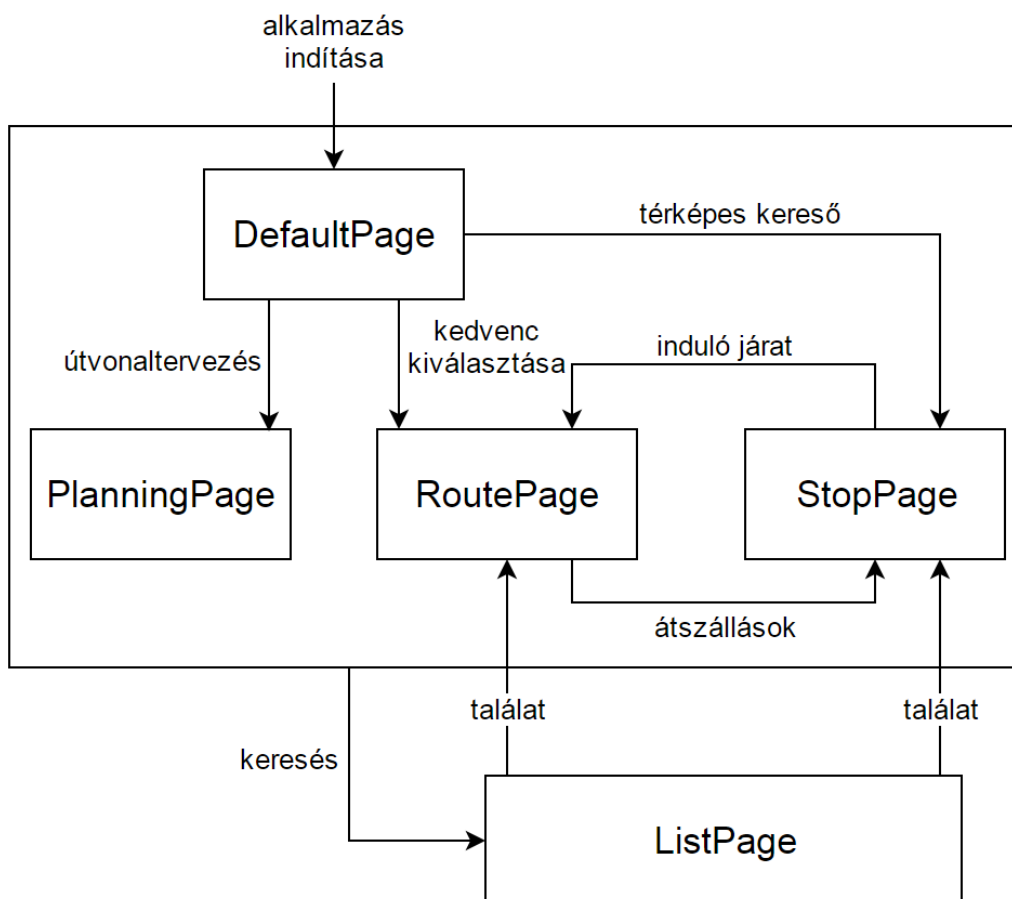
CityTransitApp. WindowsPhone	A tartalma az eredeti alkalmazás oldalainak megjelenítése. Az oldalak XAML kódját módosítanom kellett az új platform változásai miatt. Az oldalak logikáját kiszerveztem, hogy újrahasznosíthatóak legyenek.
CityTransitApp. Windows	Teljesen új komponens, az eredeti oldalak logikáját (ViewModel-ek) használja, csak a megjelenítést kellett megvalósítanom.
CityTransitServices. Tools	Az eredeti alkalmazás eszközeit helyeztem ide (pozíció, aszinkron hívások, felhasználói becslések, beállítások). Sok mindent át kellett írni a megújult API miatt (pl. http kérés, GPS lekérdezés, kicsomagolás, LocalSettings).
CityTransitServices. Processes	Új komponens, összetettebb üzleti folyamatokat szerveztem ide (pl. adatbázis letöltés/betöltés, frissítés keresés).
CityTransitElements	A tartalma az eredeti alkalmazás saját vezérlői (pl. kedvenc panel, megálló panel), ezek újrahasznosíthatók az asztali alkalmazásban. Ide szerveztem ki az oldalak logikáját (ViewModel osztályok), amely az eredeti oldalak állapotát vezérli. Ezekhez írtam új megjelenítést az asztali alkalmazásban.
PlannerComponent	Az útvonaltervezés algoritmus, nem változtattam rajta semmit.
TransitBase	A menetrend adatbázis (fogalmi szint), nem változtattam rajta semmit. Új interfészt írtam a táblák és a menetrend logika (TransitBase.BusinessLogic) eléréséhez. Az eredeti alkalmazás Storage mappáját hoztam át ebbe.
TransitBase. BusinessLogic	Ide szerveztem a menetrend magasabb szintű műveleteit (menetrend, következő indulások) változtatások nélkül. Az eredeti alkalmazás Model mappáját hoztam ide.
FastDatabaseLoader	Általános adatbetöltő, nem változtattam a működésén semmit. Mivel a reflexió API változott, kénytelen voltam kézzel megírni a hiányzó függvényeket (ReflectionExtension.cs).
UserBase	Az eredeti alkalmazás Dynamics mappáját hoztam át. Sajnos kénytelen voltam a nagyját újra írni, mert „Linq to SQL”-t használtam adattárolásra, ez már a Windows 8.1-ben nem támogatott, így az SQLite keretrendszert kellett használnom. Ehhez bizonyos lekérdezéseket és entitás attribútumokat át kellett írnom. Muszáj volt a megjelenítéssel egy szerelvényben elhelyezni, mert a szükséges függőségeket külön DLL-ben nem tudtam elérhetővé tenni.

A következő fejezetekben az új vagy módosított komponenseket fogom dokumentálni. Az eredeti appból változtatás nélkül átvett komponensekre helyhiány miatt nem térek ki, ezek elérhetőek a diplomaterv portálon a szakdolgozatomban (pl. útvonaltervezés algoritmus, menetrend generálása).

4. Az asztali alkalmazás felülete

Az alkalmazás asztali megjelenítésre optimalizált verziójának teljesen új felületet írtam. Itt sokkal nagyobb képernyőfelület áll rendelkezésre, így egy oldalon több információt is meg tudok jeleníteni, ezért több oldalt összevontam a telefonos verzióhoz képest. Mivel a hálózat segítségével meghatározott pozíció nem túl pontos, a közeli megállók funkciót nem implementáltam, így jelentősen egyszerűsödött a felület.

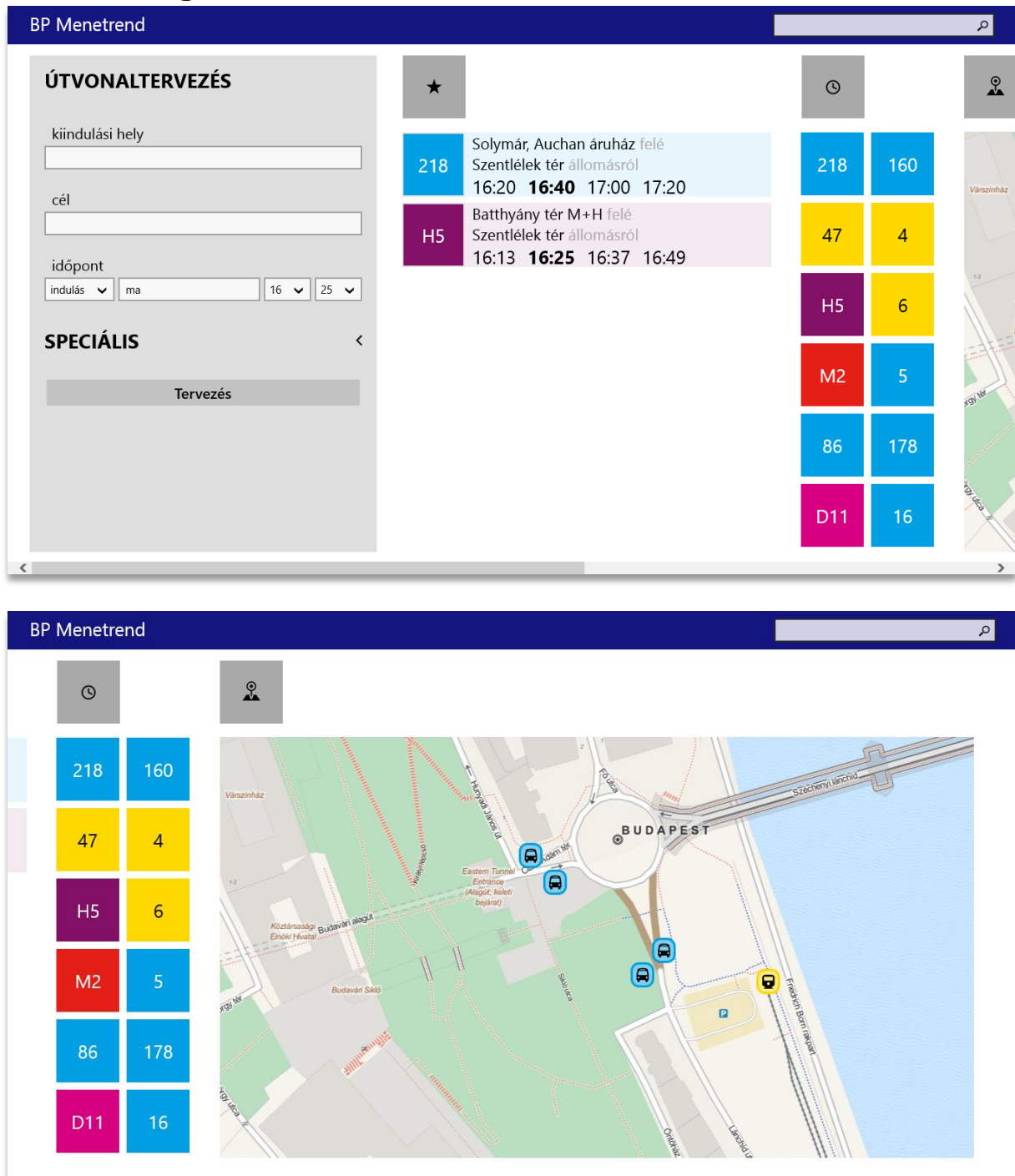
Az új felület navigációs struktúrája a következő ábrán látható.



Az alkalmazás minden oldalán található egy fejléc, amelyben van egy keresőfelület, ezen keresztül a keresőoldal bármely oldalról elérhető. A DefaultPage-en kívül minden oldalról lehet visszalépni az előző oldalra. Az alkalmazás keretét a **MainPage** valósítja meg, itt található a fejléc és a ContentFrame, amibe a navigációs struktúrán felsorolt oldalak fognak kerülni.

Az egyes oldalak az **MVVM** mintát követik, ahol csak a View részeket kellett újraírnom, a ViewModel-ek és a Model-ek, majdnem ugyanazok, mint amiket a telefonos alkalmazásban használtam, így ezeket a félév során nem kellett megírnom, csak a minta szerinti osztályokba szerveznem.

4.1 DefaultPage



Ez az alkalmazás kezdőoldala. Innen indítható útvonaltervezés, láthatjuk a kedvenceinket, előzményeinket (nemrég bezárt lapok) és egy térképes megálló keresőt.

A megjelenített adatokat a MainViewModel osztály vezérli, ugyanezt használtam a telefonos app kezdőoldalán. A mögöttes logikában csak az egyes elemekre kattintások eseménykezelőit implementáltam (navigációk, adatbevitel).

Az oldalaknál általában a tartalom vízszintesen terjed, a Windows Store-os appok stílusának megfelelően, azaz a teljes oldalt vízszintes görgetéssel lehet megnézni.

4.2. ListPage

The screenshot shows the 'BP Menetrend' app interface. At the top, there's a search bar with 'közvágó' entered. Below the search bar, a summary bar shows 'Keresési eredmények: "közvágó"' followed by icons and counts: a location pin (x 3), a train (x 3), a tram (x 2), and a bus (x 3). The results are organized into three columns: 'Megállók' (Stops), 'Járatok' (Routes), and 'H6' (H6 line). The 'Megállók' column lists 'Közvágóhíd' (H6, H7, 1, 2, 24, 23, 23E, 54, 55, 179,...), 'Közvágóhíd tér' (124, 125, 170, 270, 950), and 'Közvágóhíd utca' (124, 224). The 'Járatok' column lists routes 2, 24, 179, 901, 937, and H6 with their destinations. The 'H6' column lists routes H6 with destinations Tököl and Közvágóhíd H, and H6 with destinations Dunaharaszti külső and Közvágóhíd H.

Megállók	Járatok	H6
Közvágóhíd H6, H7, 1, 2, 24, 23, 23E, 54, 55, 179,...	2 → Közvágóhíd H → Jászai Mari tér	H6 → Tököl → Közvágóhíd H
Közvágóhíd tér 124, 125, 170, 270, 950	24 → Közvágóhíd H → Keleti pályaudvar M	H6 → Dunaharaszti külső → Közvágóhíd H
Közvágóhíd utca 124, 224	179 → Közvágóhíd H → Csepel, Szent Imre tér	
	901 → Bécsi út / Vörösvári út → Közvágóhíd H	
	937 → Bécsi út / Vörösvári út	
	H6 → Ráckeve → Közvágóhíd H	

Ha a fejlécben található kereső dobozba írunk szöveget, akkor ide navigál. A telefonos apphoz hasonlóan a szöveg alapján talált megállókat és járatokat listázza ki. A keresési találatokra kattintáskor a hozzájuk tartozó RoutePage vagy StopPage-re jutunk.

A keresési találat paneleket és a találatok számát összegző dobozokat, egy az egyben a telefonos app felületéről vettem át (RoutePanel, StopPanel, ResultCounter).

ViewModel-nek a ListPageViewModel-t hoztam létre, ezt az eredeti SearchViewModel-ből örököltem. Ugyanazt tudja, csak az eredményeket két külön listába (megállók/járatok) szedi szét (így terjesztettem ki a megjelenítésnek megfelelően a ViewModel-t). Az eredeti SearchViewModel volt a felelős a keresés végrehajtásáért a szöveg alapján.

4.3. RoutePage

BP Menetrend

←

⌕

H5

Batthyány tér M+H felé

▼

Szentlélek tér megállóból

▼

2015. december 16.

▼

Indulási időpontok

	33	36	43	46	53	56
8	03	08	13	19	25	32
	47	55				
9	04	14	24	34	44	54
10	04	14	24	34	44	54
11	04	14	24	34	44	54
12	04	14	24	34	44	54
13	04	14	24	34	42	49
14	04	12	19	27	34	42
	57					
15	04	12	18	25	31	37
	49	55				
16	01	07	13	19	25	31
	43	49	55			
17	01	07	13	19	25	31
	43	49	56			
18	04	12	19	27	34	44
19	04	14	24	34	44	54
20	04	14	24	34	44	54
21	04	24	34	54		
22	04	24	34	54		
23	04	24	34			

Kiválasztott járat megállói

16:30	•	Szentendre
16:32	•	Pannóniatelep
16:37	•	Pomáz
16:40	•	Szentistvántelep
16:42	•	Budakalász, Lenfő
16:44	•	Budakalász
16:48	•	Békásmegyér
16:51	•	Csillaghegy
16:53	•	Rómaifürdő
16:55	•	Aquincum
16:57	•	Kaszásdűlő
16:59	•	Filatorigát
17:01	•	Szentlélek tér
17:02	•	Tímár utca
17:04	•	Szépülgői út
17:06	•	Margit híd, budai hídfő
17:09	•	Batthyány tér

Ezen az oldalon jelenítem meg egy adott járatral kapcsolatos információkat. Az eredeti telefonos app TimetablePage, RoutePage és TripPage oldalait vontam itt össze (valamint a hozzájuk tartozó térképet). A fejlécben kiválasztható az irány, a megálló és a nap, amely menetrendjére kíváncsiak vagyunk és itt menthetjük az oldalt a kedvencek közé. Bal oldalt található a menetrend (TimetablePage), itt kiválaszthatunk egy indulási időpontot. A kiválasztott indulási időpont járműveinek megállóit középen találjuk, ha itt egy megállóra kattintunk az ottani átszállások oldalára jutunk. Jobb oldalon a járatot látjuk térképen is a könnyebb tájékozódás érdekében.

Az oldal ViewModel-je az új RoutePageViewModel osztály. Ez a TimetableViewModel-t és a TripViewModel-t fogja össze, ezek eredetileg külön oldalakon jelentek meg.

Az menetrend és a járat megjelenítését újraírtam, de a térképes nézet vezérlőjét közvetlenül át tudtam venni a telefonos app-ból (TripMapController). Ez az osztály felelős a térkép beállításáért (elemek elhelyezése, nézőpont beállítása) a paramétereknek megfelelően.

A felület reszponzív, ha nem elég széles az oldal (túl kicsi a kijelző), akkor a térkép csak jobbra görgetéskor látható, így nem lesz túl keskeny a menetrend, a megálló lista vagy a térkép. Azonban, az oldal mindig kitölti a rendelkezésre álló helyet.

4.4. StopPage

BP Menetrend petőfi híd

← Petőfi híd, budai hídfő

Induló járatok

Járat	Idő	Megálló
154	17:07 +0'	Neumann János utca
4	17:08 +7'	Széll Kálmán tér M
6	17:08 +7'	Móricz Zsigmond körtér M
212	17:08 +7'	Svábhegy
4	17:10 +3'	Újbuda-központ M
6	17:10 +3'	Széll Kálmán tér M
212	17:12 +5'	Boráros tér H
154	17:19 +72'	Gazdagréti tér
153	17:21 +74'	Neumann János utca
153	17:34 +27'	Gazdagréti tér

Jelenleg nem közlekedő járatok

Járat	Megálló
918	Kelenföld vasútállomás M
918	Óbuda, Bogdáni út

Térképen

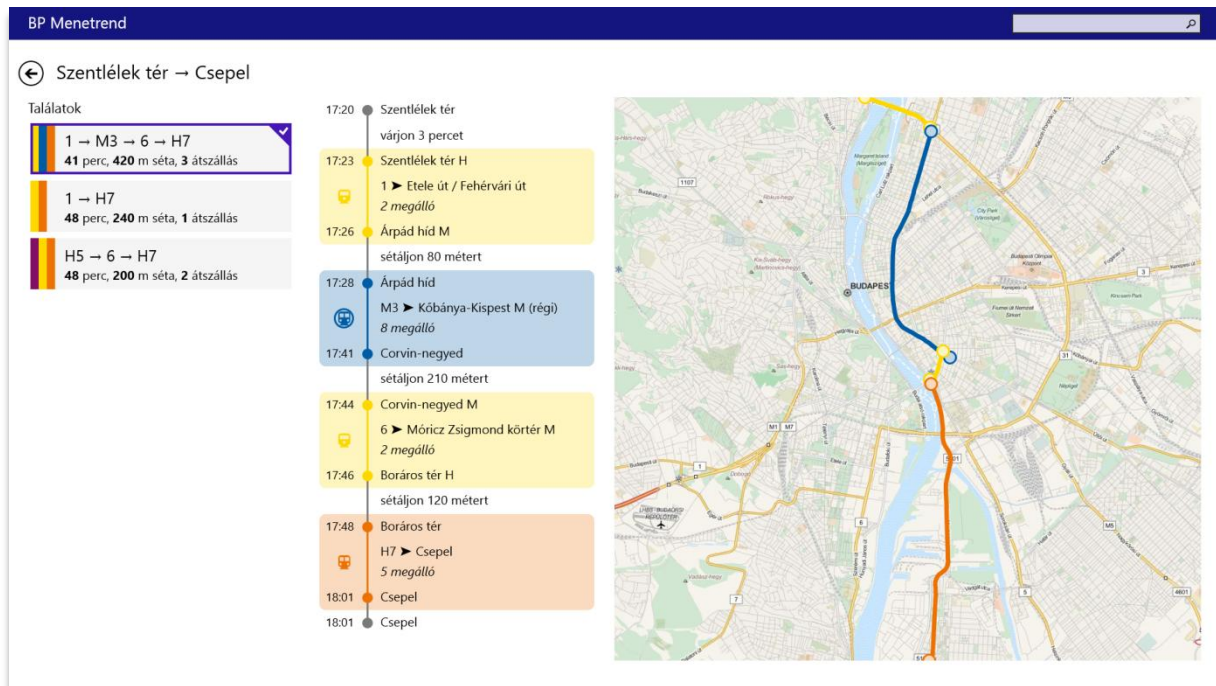
Petőfi híd, budai hídfő
indulások 17:07 után
4 Széll Kálmán tér M 1'
6 Széll Kálmán tér M 3'

Ez az oldal az eredeti app StopPage-ének és a hozzá tartozó térképes oldalnak felel meg. Láthatóak a megállóból induló járatok (hány perc múlva indulnak és átszállás esetén milyen messze vannak). Következő oszlopban a jelenleg nem közlekedő járatok láthatóak (pl. éjszakai járatok nappal). A következő oszlopban a közeli megállók listáját láthatjuk (a képernyőképen nincs ilyen). Az utolsó oszlopban a megálló térképes nézete látható, itt az adott megállót kiválasztva láthatjuk az onnan induló járatokat.

ViewModel-je a StopPageViewModel, amit az eredeti StopViewModel-ből örököltettem. Ez szét tudja szedni az érkező járatokat két listába, valamint tartalmazza a közeli megállók listáját. Az eredeti osztály számítja ki a megállóba érkező járatok indulási idejeit, ezt már nem kellett külön megírnom.

A megjelenítést itt is újra kellett írnom (teljesen máshogy néznek ki a listák, mint telefonon). A térkép vezérlőjét itt is egy az egyben át tudtam venni (a vezérlő rakja a térképre a megálló ikonokat és állítja be a nézőpontot). A mögöttes kódban az eseménykezelőket és a ViewModel osztályt írtam meg.

4.5. PlanningPage



Ez az oldal jelenik meg, ha a kezdőoldalon útvonaltervezést indítunk. Összevontam tartalmazza az eredeti alkalmazás PlanningTab-jának listáját, a PlanDetailsPage oldalt és a hozzá tartozó térképet, tehát 3 különböző szintet jelenít meg egy oldalon. A bal oldalon látható a tervezés eredményeként megjelent különböző útvonalak (alternatív útvonalak időre vagy séta távra vagy átszállások számára optimalizálva). Középen látható a kiválasztott útvonal részletes terve, szakaszokra bontva. Jobb oldalt látható az útvonal térképes nézeten, az pontokat kiválasztva megjelennek az átszállások részletei.

Az oldal ViewModel-je a PlanningViewModel (a telefonos app útvonaltervezés listájában használok ezt). Ez tartalmaz WayModel-eket, ezek egy úthoz tartalmazó modellek (ezeket a telefonos app PlanDetailsPage oldalán használtam). Ez a VM felelős az útvonaltervezés indításáért a paraméterek szerint, és annak eredményének értelmezéséért.

A megjelenítésben itt újat nem kellett írnom. A listákat a telefonos app hozzájuk tartozó oldalainak listájainak újrafelhasználásával kaptam. A térkép vezérlője a PlanMapController, amit egy az egyben átvettem. Ez felelős a nézet beállításáért, az útvonal kirajzolásáért, a pontok és szövegdobozok elhelyezéséért.

5. A telefonos alkalmazás felülete

A telefonos felület logikája az eredetihez képest nem változott (kicsit szépítettem a kinézetén). A MapPage-t bővítettem két új nézettel, már a járatokhoz és a megállókhöz is tartoznak térképes nézetek (az útvonaltervezés mellett).

Az XAML nyelv változásai miatt helyeken át kellett írnom a .xaml fájlokat (pl. lokalizációs sztringek, WindowsPhoneToolkit hiánya, méretek frissítése), de a megjelenítést jelentősebben nem kellett megváltoztatnom.

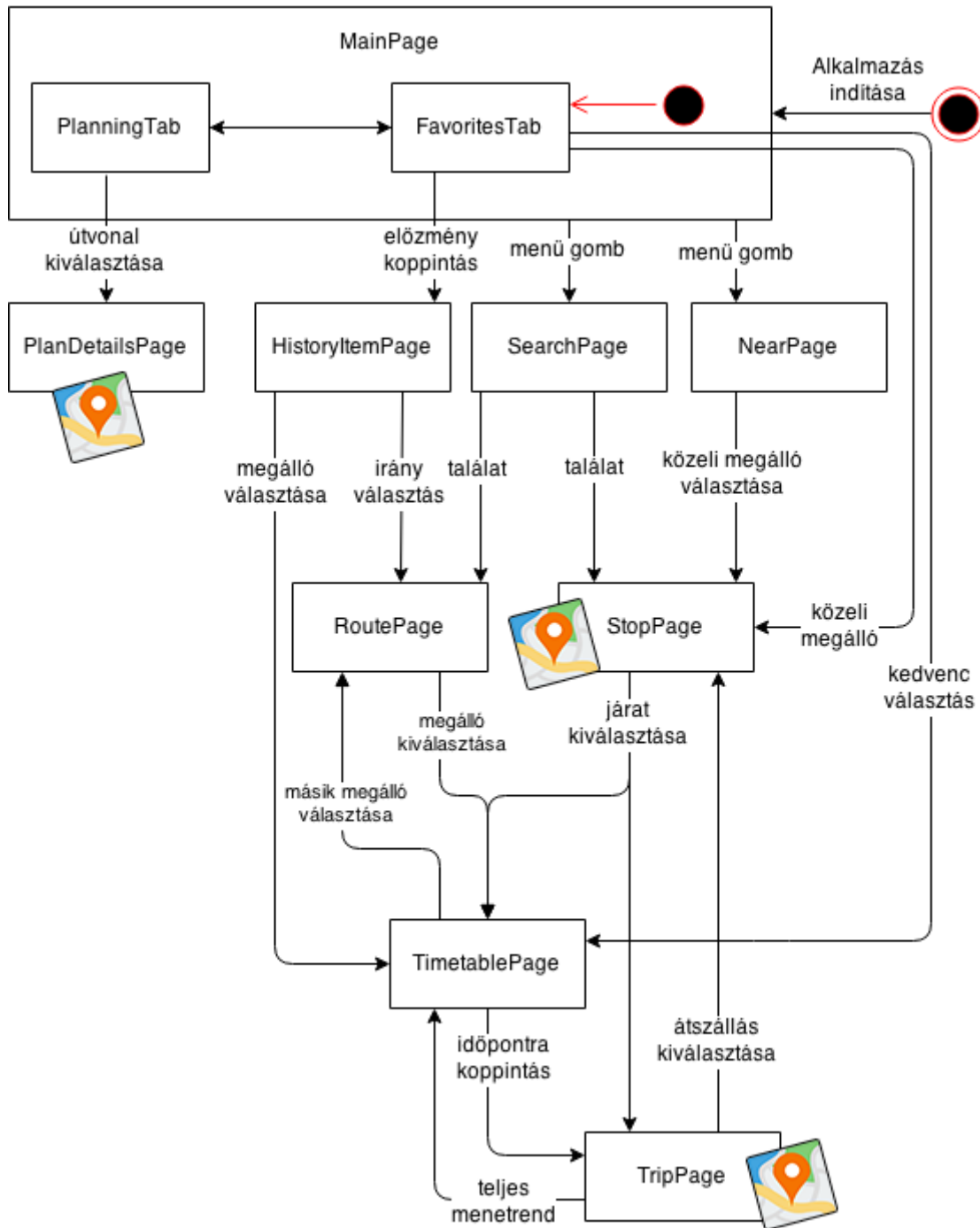
Be kellett vezetnem az **MVVM** mintát, mert WP8.1-ben csak így tudom visszaállítani az oldal tartalmát a vissza gomb megnyomása után (ha elnavigálok egy oldalról egyből megszűnik a tartalma). Ez hasznos volt, mert a ViewModel osztályokat újra tudtam hasznosítani az asztali felületnél. A minta bevezetéséhez minden oldalon a logikát és az oldalak állapotát ki kellett szerveznem a hozzájuk tartozó ViewModel osztályokba, úgy hogy az oldalak mögöttes kódjában ne maradjon az oldal vezérlőit manipuláló kód, minden ilyen a DataContext-nek beállított ViewModel csináljon. A ViewModel-ek a **CityTransitApp.Shared/CityTransitElements/ViewModels** mappában találhatóak.

Az oldalak közötti navigációt is meg kellett változtatnom. Már nem Uri sztringek segítségével történik a navigáció, hanem a cél oldal típusának és egy paraméter objektumnak a megadásával. Így minden oldalhoz kellett írnom egy paraméter osztályt, amely objektumai az oldalak kezdőállapotát határozzák meg. Ezek az osztályok a **CityTransitApp.Shared/CityTransitElements/Parameters** mappában találhatóak.

A kedvencek élőcsempére való kitűzését ki kellett vennem, mert memória hiány miatt a háttérben futó szál mindig meghalt. Továbbfejlesztési lehetőség ennek a visszarakása, de ehhez a háttérszálat c++-ban kellene megírni, mert csak így tudunk elég takarékosan bánni a memóriával képgeneráláskor, azonban erre a félév során nem volt időm.

5.2 Navigációs struktúra

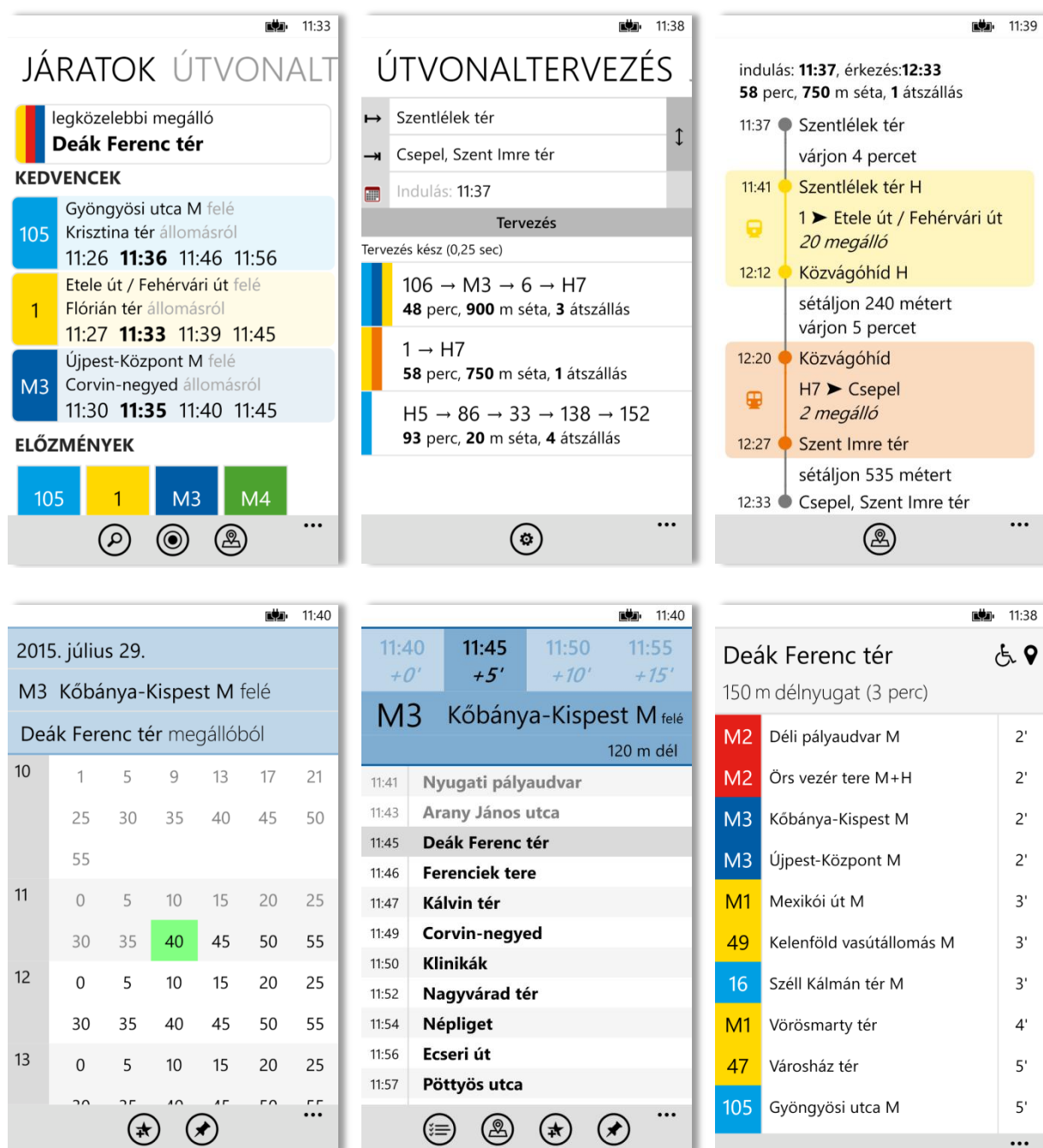
A telefonra optimalizált alkalmazás navigációs struktúrája a következő ábrán látható.



Az ábrán térképpel jelöltem, hogy melyik oldalakhoz tartozik térképes nézet. Ezeket a nézeteket a MapPage-ben valósítottam meg. A térképes nézetekről ugyanoda lehet navigálni, mint az őket tartalmazó oldalakról.

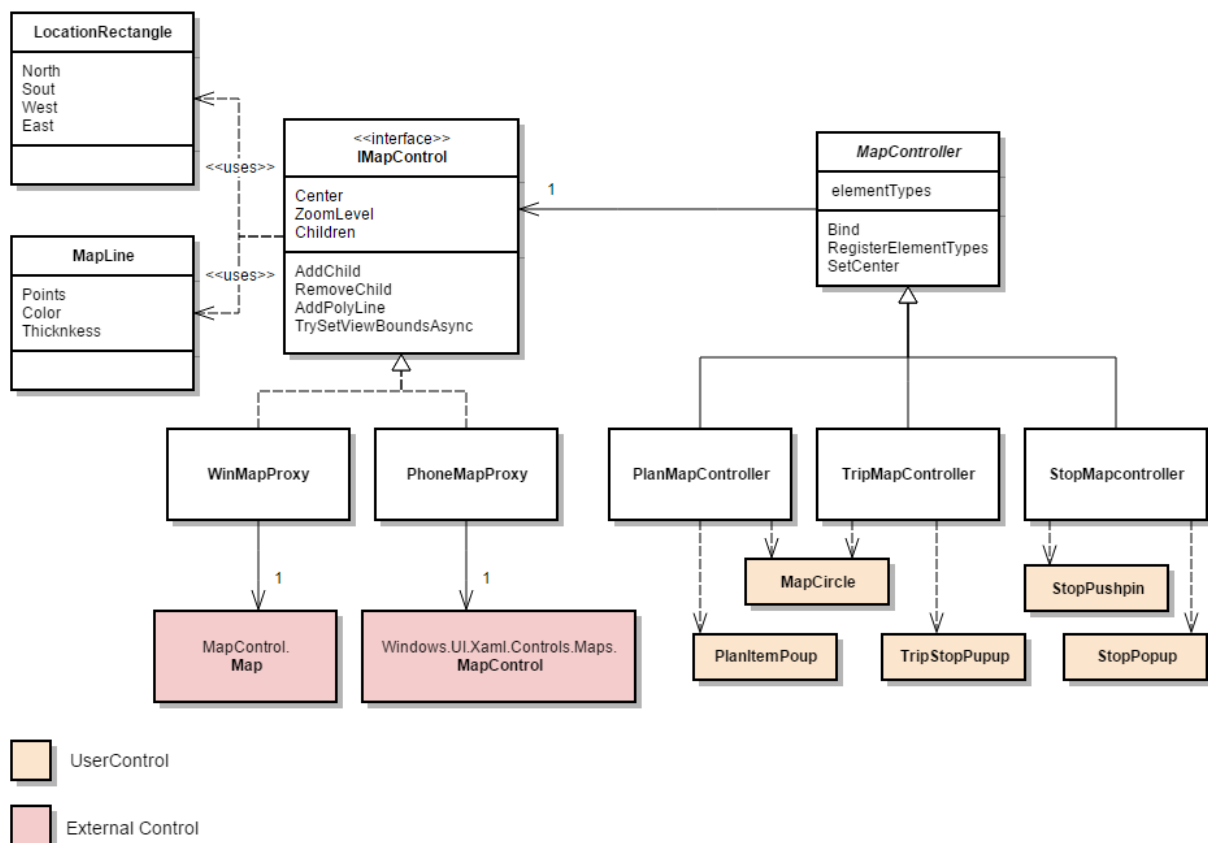
5.3 Oldalak új kinézete

Az oldalak új egyszerűsített kinézete a következő képernyőképeken látszik.



6. Térképek

Az alkalmazásban a megállókhoz, a járatokhoz és az útvonaltervekhez van szükség térképes nézetre. A probléma az, hogy Windows Phone és Windows környezetben más térkép vezérlők állnak rendelkezésre, de nem akartam kétszer megírni a térképes megjelenítés logikáját. Ehhez a Proxy tervezési mintát használtam fel, a konkrét térkép vezérlőket egy absztrakt proxy interfésszel fedtem el, amelyen keresztül a közös logika kezelni tudja a megadott térképet. A térképek kezelését a következő ábra szemlélteti:



A fenti osztályok többsége a CityTransitApp.Shared/CityTransitElements/Controllers/Maps mappában találhatóak. A WinMapProxy a CityTransitApp.Windows/Implementations, a PhoneMapProxy a CityTransitApp.WindowsPhone/Implementations mappában találhatóak. Az alsó két osztály, Map és MapControl külső osztályok. A PlanItemPoup, TripStopPoupup, StopPoupup, StopPushpin vezérlők a CityTransitApp.Shared/CityTransitElements/Page-Elements/MapMarkers mappában találhatóak.

Az osztályok rövid leírását a következő táblázatok tartalmazzák.

Általános elemek

IMapControl	Interfész, azokat a műveleteket valósítja meg, amelyek szükségesek a térkép kezelő logikához.
LocationRectangle	Segédosztály, a nézet beállítás paramétere. Földrajzi négyzet.
MapLine	Segédosztály vonal rajzolásához a térképen. Pontokból, színből és vastagságból áll.
MapController	Általános vezérlő, amely IMapControl-t implementáló térképek kezelését teszi lehetővé. Absztrakt osztály, LocationRectangle-t számító segédfüggvények találhatóak itt, valamint eseményben értesít térkép elemekre vagy térkép háttérre kattintásról.
PlanMapController	A Bind-ben kapott térképre rajzol tervezési útvonalat a kapott paraméter (Way) alapján. Figyeli a pontokra kattintást és megjeleníti a szükséges szövegdobozokat ekkor.
StopMapController	A Bind-ben kapott térképre rajzolja a megállókat (kiemeli a paraméterül kapott főbb megállókat). Ha megállóra kattintok megjeleníti az ahhoz tartozó információs panelt az érkezési idővel. Kifele zoomoláskor eltünteti a megállókat bizonyos távolság után, hogy ne telítsék meg a térképet. Visszazoom-oláskor újra megjeleníti őket. Mindig az éppen képen lévő terület megállót jeleníti meg.
TripMapController	A Bind-ben kapott térképre rajzol egy vonalat a járat útvonala mentén, valamint pontokat rajzol a megállóiba. A pontokra kattintáskor megjeleníti az adott megálló információs dobozát az érkezési idővel és az átszállásokkal.
MapCircle	Egy sima Grid, de saját típus kellett a regisztráláshoz.
PlanItemPopup	Útvonalterv átszállási pontjának információs panele.
TripStopPopup	Járat megálló pontjának információs panele.
StopPushpin	Megállót jelölő ikon.
StopPopup	Megálló ikonra kattintáskor felugró információs ablak. Az adott megállóból induló járatok listáját tartalmazza.

Windows Phone elemek

MapControl	A HERE Maps-es térkép vezérlő. Internetkapcsolat nélkül képes térképeket megjeleníteni a telefonon mentett térképek segítségével. A használata ingyenes, telefonos használatra tökéletes. Kibővíthető elemek és alakzatok hozzáadásával.
PhoneMapProxy	A MapControl műveleteit elfedő proxy osztály. Megvalósítja az IMapControl interfészt, így a MapControl-ok kezelni tudják az elfedett térképet.

Windows elemek

Map	3rd Party térképkezelő, a binárisa a projekt gyökerének dependencies mappájának MapControl.Winrt.dll fájljában található meg. OpenStreetMaps térképeket használ, ingyenesen lehet vele korlátlan számú kéréseket küldeni. Használatához internetkapcsolat szükséges. Kibővíthető elemek és alakzatok hozzáadásával. A projektről leírás a következő címen érhető el: https://xamlmapcontrol.codeplex.com/
WinMapProxy	A Map műveleteit elfedő proxy osztály. Megvalósítja az IMapControl interfészt, így a MapControl-ok kezelni tudják az elfedett térképet.

7. Összegzés

Munkám során specifikáltam a feladatot, megterveztem a szükséges architektúrális változásokat, majd implementáltam a menetrend alkalmazás asztali számítógépre és tabletre optimalizált változatát is egy Windows 8.1 Universal App formájában. A félév elején a szakdolgozatomból indultam ki, amely akkor még csak egy telefonos alkalmazásból állt, de most már használni lehet ezt az alkalmazást bármilyen legalább Windows 8.1-gyel rendelkező asztali számítógépen és tableten.

Az alkalmazás segítségével elérhetőek Budapest összes metró, HÉV, villamos és busz vonalának menetrend információi. Tájékozódhatunk járataink indulásairól és a célmegállóinkba érkezésének pontos időpontjairól. Átszállások között keresgélhetünk a kiválasztott megállóknak. Útvonalakat tervezhetünk indulás vagy érkezés szerint bármikor, bármely két megálló között. Mindehhez térképes felület segít minket a könnyebb tájékozódás érdekében.

Munkám során nagy mennyiségű tervezési és implementációs munkát végeztem, amelynek eredménye egy több platformon működő vastagkliens alkalmazás. Bár az alkalmazás nem tökéletes, előfordulhat hibás működés és vannak továbbfejlesztési lehetőségek is, de egy jól használható, egyszerű, könnyen továbbfejleszthető alkalmazást sikerült elkészítenem.