**Vizuális programozás beadandó**

**4.2-es feladat dokumentáció**

***WPF alapú telefonkönyv rendszer fejlesztése MS SQL Server Express adatbázissal***

**1. Bevezetés**

A jelenlegi dokumentáció célja bemutatni a 4.2-es számú beadandó gyakorlati feladat teljes megvalósítását, amelynek során egy Windows Presentation Foundation (WPF) technológián alapuló, grafikus felhasználói felülettel rendelkező alkalmazás készült. A rendszer egy telefonkönyvet valósít meg, amely lehetővé teszi kapcsolatok és hozzájuk tartozó telefonszámok kezelését, tárolását, módosítását és törlését.

A megoldás során különös figyelmet fordítottam az adatszerkezet kialakítására, a relációs adatbázis helyes felépítésére, az Entity Framework modell generálására, a WPF struktúra és eseménykezelés szabályaira, valamint az esztétikus, modern dizájn és felhasználóbarát működés kialakítására.

**2. A feladat célja és elvárásai**

A feladat keretein belül az alábbi elvárásokat kellett teljesíteni:

* Készíteni kellett egy **MS SQL Server Express** alapú adatbázist, legalább két táblával.
* A táblák között egy **egy-a-többhöz kapcsolatot** kellett létrehozni (pl. egy kontakthoz több telefonszám tartozhat).
* A kliensoldali megjelenítés **WPF** technológiával történt.
* Az alkalmazásnak biztosítania kellett a **CRUD** (Create, Read, Update, Delete) funkcionalitást:
  + új kapcsolatok rögzítése,
  + kapcsolatok listázása,
  + adatok módosítása, törlése,
  + telefonszámok kezelése kontaktonként.

A programhoz tartozó összes vezérlő és logika a kliensoldalon történt, nem használtam külön szerveroldali feldolgozást.

**3. Adatbázis struktúra és modell**

Az adatbázis neve: **Telefonkonyv**

A struktúra két fő táblát tartalmaz:

**Kontaktok tábla**

ID INT (PK, Identity)

Nev NVARCHAR(100)

Email NVARCHAR(100)

**Telefonszamok tábla**

ID INT (PK, Identity)

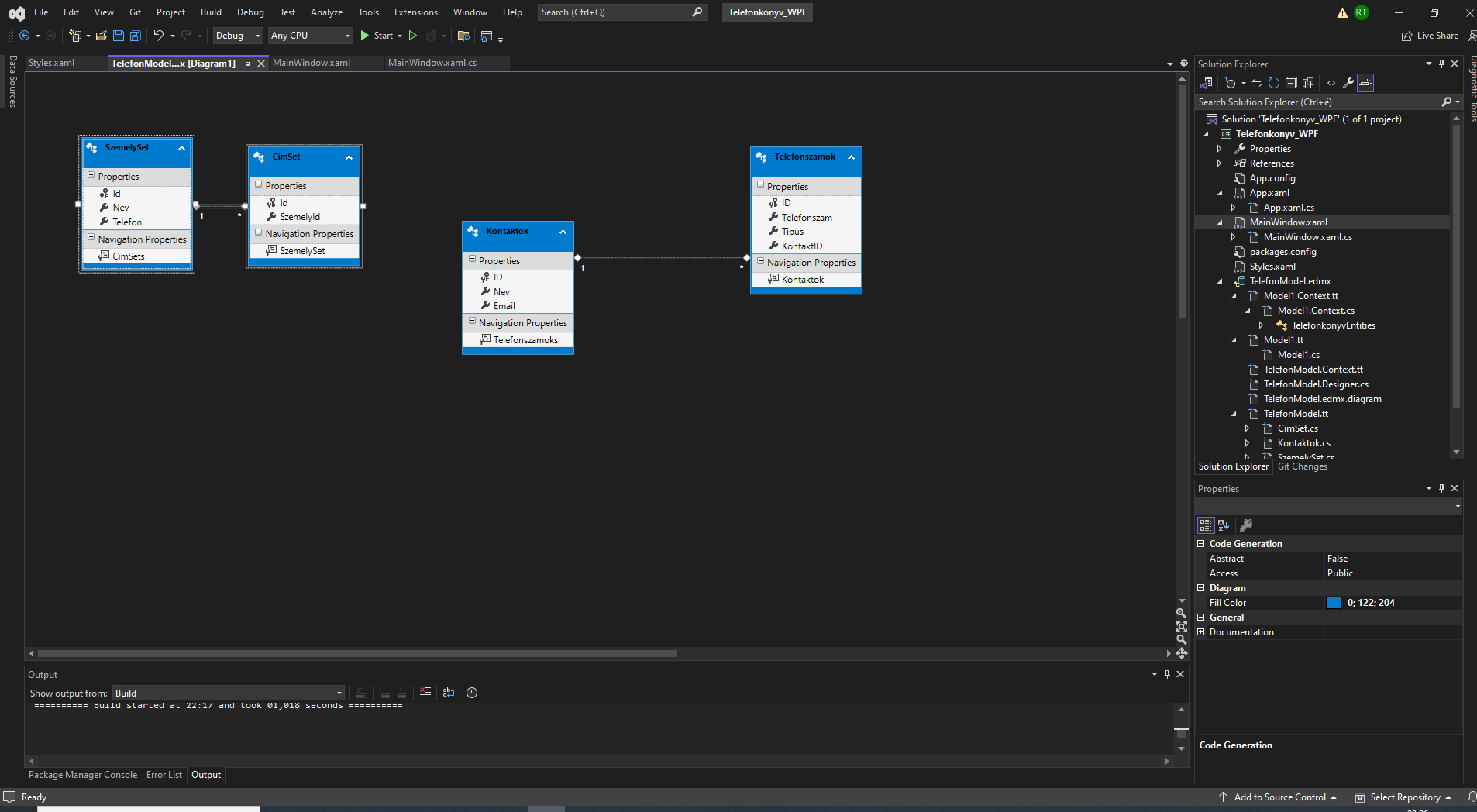
Telefonszam NVARCHAR(20)

Tipus NVARCHAR(20)

KontaktID INT (FK → Kontaktok.ID)

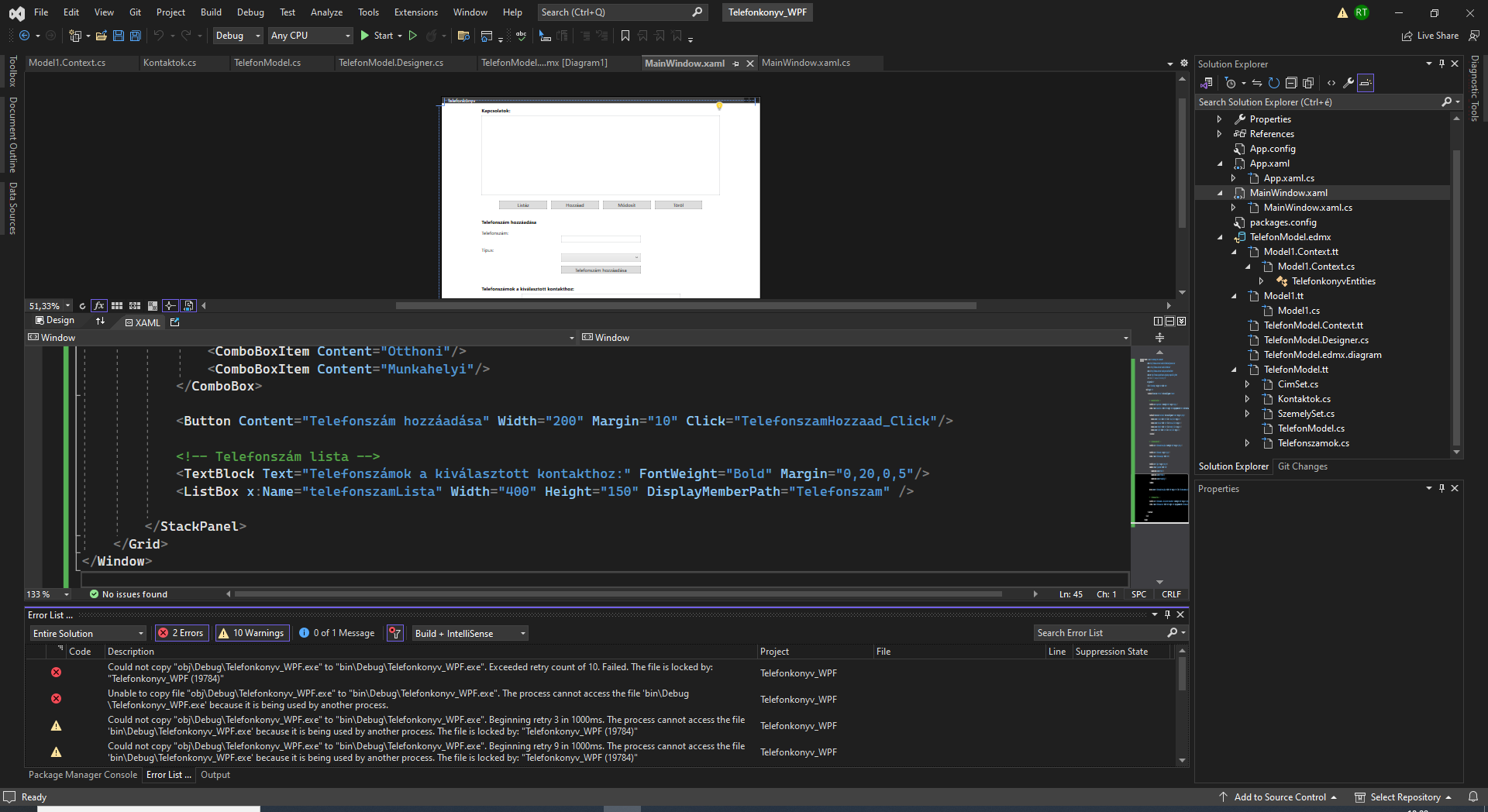
A két tábla között **egy-a-többhöz (1:N) kapcsolat** van: egy kontakt több telefonszámot is tárolhat. A kapcsolatot idegen kulcs (Foreign Key) biztosítja.

Az adatbázist SQL Server Management Studio segítségével hoztam létre. A táblák generálása után az edmx modellen keresztül építettem fel az Entity Framework kapcsolatot.



**4. Az alkalmazás funkcióinak bemutatása**

A program kezdetleges, elsődleges felépítése után….



Folytatólagosan egy **háromoszlopos Grid-elrendezésben** építettem fel, ahol:

* **Bal oldalon**: a kontaktok listája és műveleti gombjai
* **Középen**: kapcsolat létrehozási űrlap
* **Jobb oldalon**: telefonszámok kezelése (űrlap, lista, műveleti gombok)

**📋 Kontakt funkciók:**

* **Kapcsolat hozzáadása:** név és e-mail megadása után egy új rekord kerül be az adatbázisba.
* **Listázás:** az aktuálisan elérhető összes kontakt betöltése a ListBox-ba.
* **Módosítás:** a kijelölt rekord automatikusan frissíthető (jelenleg fix tesztadatokkal történik, fejleszthető).
* **Törlés:** az adott rekordot eltávolítja az adatbázisból, és frissíti a felületet.

**📱 Telefonszám funkciók:**

* **Hozzáadás:** a kiválasztott kapcsolathoz rögzít egy telefonszámot, a kiválasztott típussal (Mobil, Otthoni, Munkahelyi).
* **Módosítás:** a telefonszám kijelölésekor a mezők automatikusan betöltődnek, módosítás után a friss adat menthető.
* **Törlés:** a kiválasztott szám végleg törlődik.

A kapcsolatok kiválasztásakor az alkalmazás **automatikusan betölti a hozzájuk tartozó számokat** is.

**5. Felhasználói felület (UI) és dizájn**

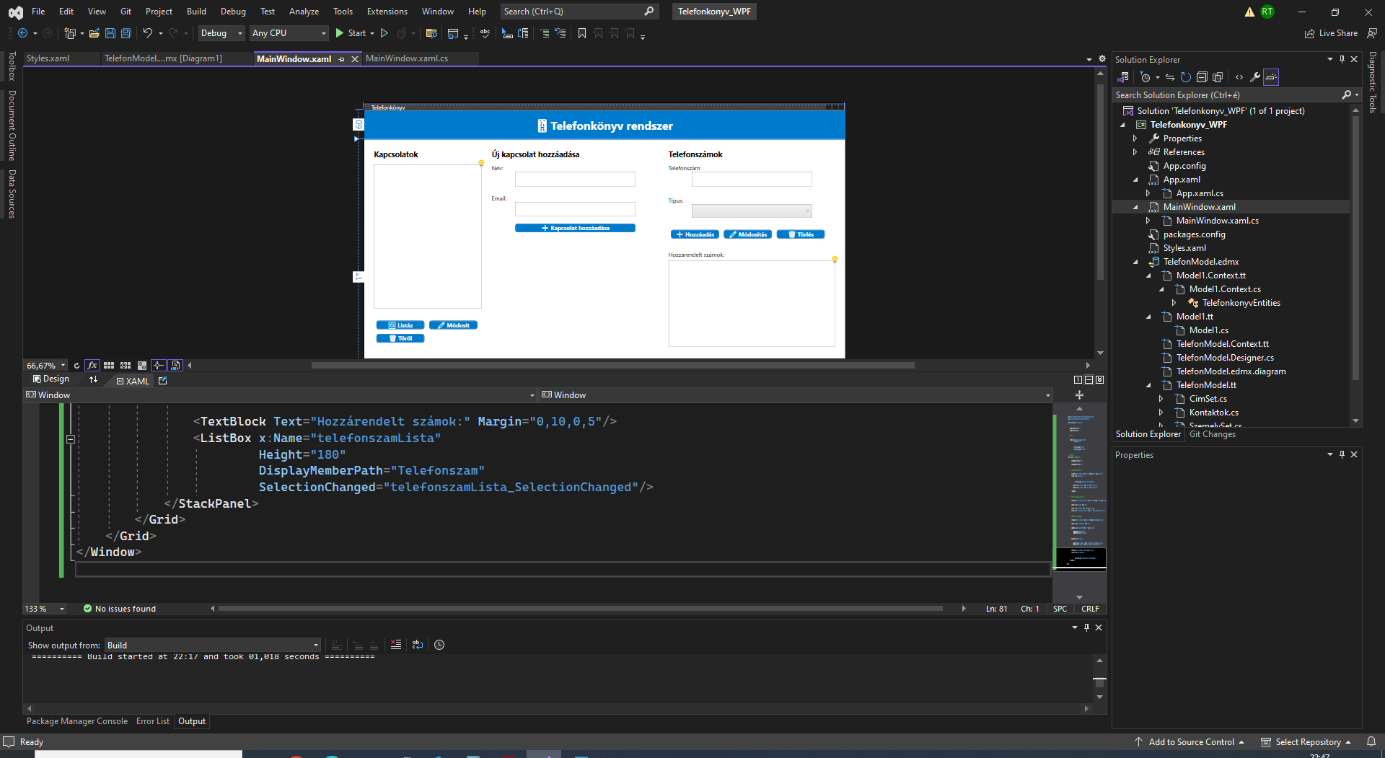
Az alkalmazás teljes felhasználói felületét a WPF Grid és StackPanel rendszere rendezi. A látványosabb felhasználói élmény érdekében az alábbi stílusmegoldásokat alkalmaztam:

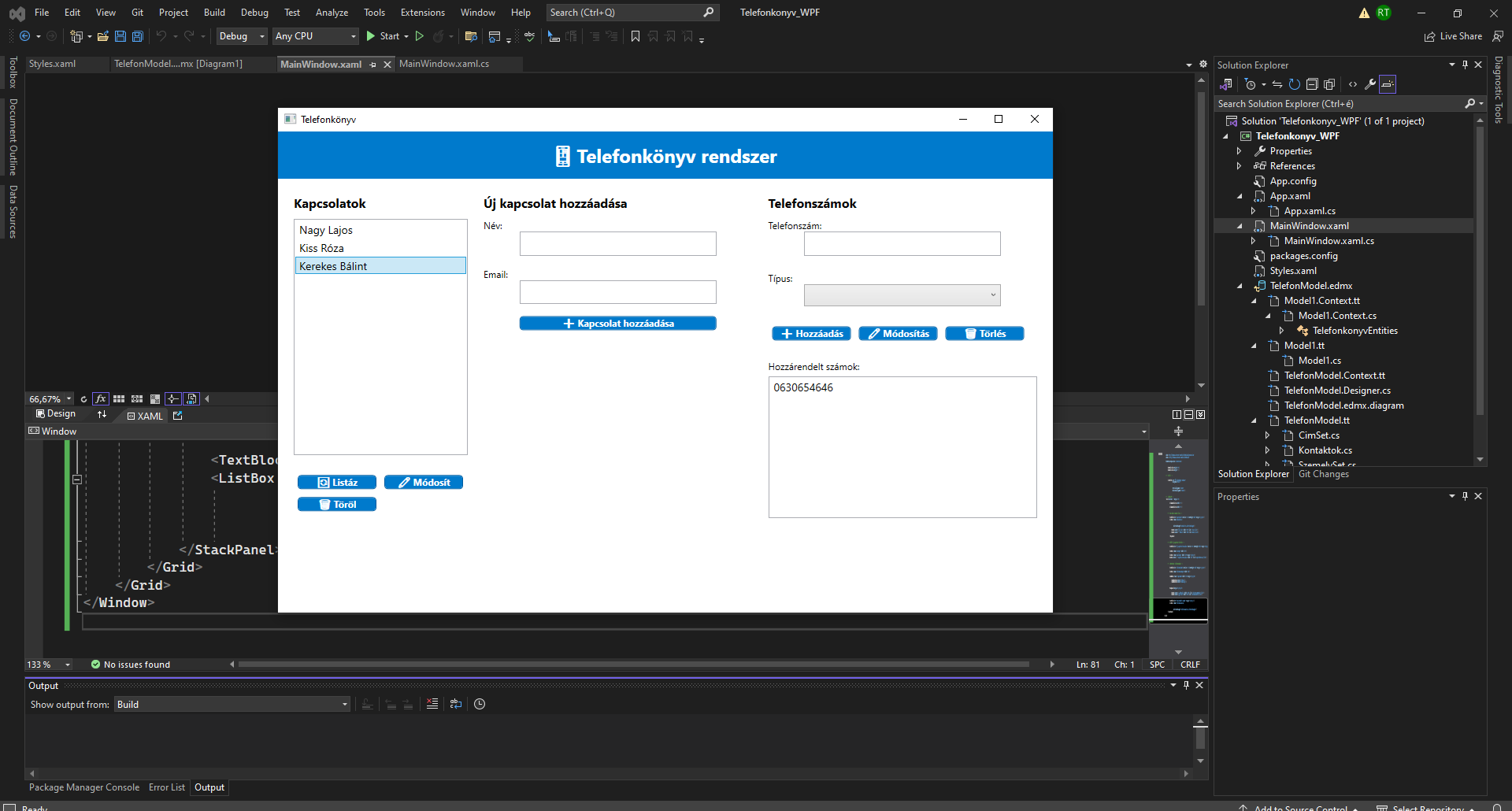
**Resource Dictionary (Styles.xaml)**

* **Egységes gombstílus:** kék háttér, fehér szöveg, lekerekített sarkok
* **Hover-effektus:** gomb fölé lépve színe sötétebb kékre változik
* **Kattintás effektus:** még sötétebb árnyalat
* **Letiltott gombok:** szürke szín

**További UI elemek:**

* Unicode ikonok (pl. ➕ ✏️ 🗑️) a gombok elé, hogy vizuálisan is könnyebb legyen értelmezni
* Fejléc (📱 Telefonkönyv rendszer): kék sávban középen, nagy betűvel
* Alkalmazás színezése világos, letisztult



****

**6. Eseménykezelés és logika**

Az összes gomb Click eseménye és lista SelectionChanged eseménye külön metódusba került. A háttérben az Entity Framework DbContext példány kezeli az adatbázisműveleteket. A kapcsolatok és telefonszámok adatai mindig frissen töltődnek be a UI-ra.

A kijelölt telefonszámot egy aktualisSzam nevű változó tárolja, amit a módosítás/törlés gomb használ.

**7. Összegzés**

A beadandó feladat célja teljesült:

* ✔️ MS SQL Server Express adatbázis használata
* ✔️ Legalább két kapcsolt tábla
* ✔️ WPF alapú grafikus felhasználói felület
* ✔️ Teljes CRUD funkcionalitás
* ✔️ Modern, letisztult dizájn
* ✔️ Működő eseménykezelés

A projekt bármikor továbbfejleszthető: további mezőkkel, validálással, mentések megerősítésével, vagy adatvizualizációval.

**GitHub elérhetőség:**

**Felhasználónév:** ZE4F8M

**Repositorie:** [vizualprog](https://github.com/trobi08/Vizualis-programozas-beadando)

https://github.com/ZE4F8M/vizualprog