**Projektterv**

|  |  |
| --- | --- |
| **Projekt adatai** | |
| Projekt neve: *Programozás 1 választható beadandó projektfeladat (DUE)* | |
| Projektet készítette: *Beödők Levente* | |
| Neptun kód: JGZW76 | |
| Dátuma: *2024. 05. 13.* | Verzió száma: *1.0* |
| **Fejlesztési igény rövid megnevezése** | |
| *Kezdetleges banki rendszer megvalósítása C# konzolos alkalmazással* | |
| GitHub link: <https://github.com/BlackWolf222/Projects> | |

[**1. Program bemutatás 3**](#_Toc166702962)

[**2. Funkciók meghatározása, részletes bemutatása 4**](#_Toc166702963)

[*2.1 Fő menü 4*](#_Toc166702964)

[*2.2 Fiók létrehozása 4*](#_Toc166702965)

[*2.3 Számla egyenleg 5*](#_Toc166702966)

[*2.4 Befizetés számlára 5*](#_Toc166702967)

[*2.5 Pénz felvétele 6*](#_Toc166702968)

[*2.6 Átutalás 6*](#_Toc166702969)

[*2.7 Fiók adatai 7*](#_Toc166702970)

[**3. Tesztesetek megfogalmazása, tesztelési elvárások 8**](#_Toc166702971)

[**4. Egyéb funkciók 8**](#_Toc166702972)

[**5.Hibák/fejlesztési lehetőségek 8**](#_Toc166702973)

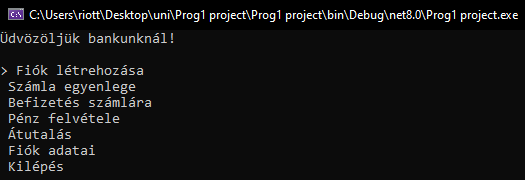
1. Program bemutatás

A beadandó programomnak egy kezdetleges banki rendszert találtam ki. A program rengeteg egyedi és tanult megoldást hordoz magában. A programot C# nyelven, Visual Studio 2022 fejlesztői környezet felhasználásával valósítottam meg. A fejlesztés során törekedtem arra, hogy minél több hasznos a való életben használt funkciót tudjak be építeni a programba. Fontos volt, hogy a program kezdetlegeségéhez képest használható legyen ezért is választottam ezt a menüs megoldást (2.1 Fő menü). Továbbá a fejlesztés lényeges részét képezte, hogy a program hasznos vissza jelzést adjon a felhasználónak hiba esetén, valamint hogy valamennyi hibát le tudja kezelni és ne fagyjon ki.

2. Funkciók meghatározása, részletes bemutatása

2.1 Fő menü

A főmenüben egy üdvözlő felírat fogad minket és egy felsorolás a különböző funkciókkal. Azért választottam a menünek az alábbi kinézetett, hogy a fel használó könnyen eligazodjon rajta. A menüs megoldást kihívás ként választottam valamint elég hasznosnak is bizonyul mivel így a programba több funkciót is létrehozhattam amik nem egyszerre futnak le. Ezzel is jobban hasonlít egy funkcionálisabb programra.



A kiválasztott funkciót egy ’>’ jelel jelöli a program.

2.2 Fiók létrehozása

Ezzel a funkcióval a felhasználó egy saját fiókot hoz létre, amit a többi funkció során fog majd használni. A fiók létrehozás egy kötelező lépés a program további használatához. A létrehozás során a felhasználónak egy nevet, egy PIN kódot, valamint egy számla típust kell megadnia.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

A felhasználó névnek csak olyan nevet adhat meg ami még nem létezik, ha ennek a követelménynek nem tesz eleget a program egy üzenettel jelzi, hogy ez a név már létezik és kéri a felhasználót egy másik név megadására.

A képen szöveg, Betűtípus, képernyőkép, szám látható

Automatikusan generált leírás

A PIN kódnak egy tetszőleges szám sorozatot kell megadnia a felhasználónak, ha ennek nem tesz eleget a program újra kéri a PIN kódot.

A számla típus kiválasztásánál a felhasználó a csekk, valamint a takarék számla közül választhat. Ha más értéket ad meg a program kéri, hogy újra válaszon amíg nem ad megfelelő értéket. Ha a felhasználó helyes adatokat ad meg akkor a program a következő megerősítő felülettel fogadja.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Itt újonnan megtalálható a létrehozás dátuma, a számlaszám, amit a program random generál, nem létezhet két ugyan olyan számlaszám, valamint az egyenleg, aminek kezdő értéke nulla. A program az adatokat a ’context.txt’ szöveges fájlba menti le. Minden funkció során található egy almenü, ahol a felhasználó ki tud lépni a programból, valamint vissza tud lépni a fő menübe.

2.3 Számla egyenleg

A funkciónak a használatához a felhasználónak létező fiókra van szüksége! Ezzel a funkcióval a felhasználó a sikeres felhasználó név és PIN kód megadása után meg tudja nézni mennyi pénz található a számláján.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

A számlán található egyenleget a context.txt fájlba tároljuk, amit a program a funkció kiválasztása után beolvas egy listába, ahonnan egy egyszerű LINQ lekérdezés segítségével le tudjuk kérdezni az egyenleget. Helytelen felhasználó név vagy PIN esetén a program hiba üzenet küld a felhasználónak.

2.4 Befizetés számlára

A funkciónak a használatához a felhasználónak létező fiókra van szüksége! Ezzel a funkcióval a felhasználó pénzt tölthet fel a számlájára. A felhasználónak meg kell adni a nevét, valamint a PIN kódját, majd a feltölteni kívánt összeget. Helytelen érték vagy létező név esetén a program újra kéri az adatot.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Sikeres feltöltés esetén a felhasználó az alábbi üzenetet kapja.

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Automatikusan generált leírás

Ellenkező esetben a program a "Nem sikerült a feltöltés" szöveget írja ki.

2.5 Pénz felvétele

A funkciónak a használatához a felhasználónak létező fiókra van szüksége! Ez a funkció lehetővé teszi a felhasználó számára, hogy név és PIN kód megadása után felvegyen egy adott összeget a számlájáról.

A képen képernyőkép, szöveg, Betűtípus látható

Automatikusan generált leírás

Amennyiben a kívánt összeg nem áll rendelkezésre a program ezt jelzi a felhasználónak.

Valamint, ha egyéb hiba adódik akkor a program lekezeli azt.

2.6 Átutalás

A funkciónak a használatához a felhasználónak létező fiókra van szüksége! A felhasználó a szokásos falhasználó név és PIN kód után meg adhatja a kedvezményezett nevét és számlaszámát és az utalni kívánt összeget.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Sikeres tranzakció esetén a program az alábbi visszaigazoló üzenetet küld.

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Automatikusan generált leírás

2.7 Fiók adatai

A funkciónak a használatához a felhasználónak létező fiókra van szüksége! A felhasználó a sikeres felhasználónév és PIN kód megadása után meg tekintheti az adatait.

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Automatikusan generált leírás

Valamint a két új almenüpontba megtekintheti a be és ki menő tranzakciókat

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás A képen szöveg, képernyőkép, szoftver, képernyő látható

Automatikusan generált leírásEzeket az adatoakat LINQ utasitásokal kérdezük le.

3. Tesztesetek megfogalmazása, tesztelési elvárások

A fejlesztés során a programot minden új funkció esetén teszteltem. A tesztelés során funkcionális és globális tesztet hajtottam végre. A rendszerben található adatok általam kitalált fiktív adatok, melyeket a tesztelési folyamatok során rögzítettem. A rendszer szembeni elvárás, hogy minden esetben működőképes legyen, valamennyi részfolyamat esetén kezelje le a hibákat, azaz ne fagyjon ki.

4. Egyéb funkciók

A program a felhasználók adatait a context.txt fájlba menti le, valamint ebből a fájlból olvassa be. A beolvasót adatokat egy listában tárolja, amit minden adatváltoztatásnál a fájlal együtt frissít így adatváltoztatás után nem kell újra indítani a programot hogy a megváltozott adattokkal tudjuk dolgozni.

A context.txt fájl felépítésé:

* Név
* Létrehozás dátuma
* Számla szám
* PIN kód
* Egyenleg
* Számla típus

Ezen kívül a tranzakciók tárolására létrehoztam a Data.txt nevű fájlt. Ebbe a fájlba kerül az összes tranzakció, amit később a legkérdezésnél be olvasunk és listában tárolunk.

A Data.txt fájl felépítésé:

* Név
* Tranzakció dátuma
* Számla szám
* Egyenleg
* Tranzakció összege
* In/out-al jelöljük, hogy a tranzakció be- vagy kimeneti

Ezeket a fájlokat az átláthatóság érdekében egy DataSource nevű mappában tároljuk el.

5.Hibák/fejlesztési lehetőségek

* Elavult kezelő felület lecserélése modernebbre
* Esetleges le nem kezelt hibák kezelése
* Adatok titkosítása, legalább a PIN kód titkosítása
* txt-s adatforrás lecserélése adatbázisra