

RUANDER Oktatási Kft.

SZAKDOLGOZAT

Nemes János
Szoftverfejlesztő

BUDAPEST

2020

RUANDER Oktatási Kft.

BeKi – A családi kassza

Konzulens:

Kis Balázs

Készítette:

Nemes János

Szoftverfejlesztő

BUDAPEST, 2020

Tartalomjegyzék

1. BEVEZETÉS	1
2. FELHASZNÁLÓI DOKUMENTÁCIÓ	3
2.1. Miért érdemes használatba venni a szoftvert?	3
2.2. A szoftver ablakai	3
2.2.1. Főablak.....	3
2.2.2. Pénzmozgás műveletek	5
2.2.3. Bizonylat	7
2.2.4. Kimutatás	8
2.2.5. Hónap generálás.....	8
2.2.6. Hónapépítés	9
2.2.7. Szerviz	10
2.3. Hibaesetek	12
2.4. Futtatási rendszerkövetelmények	15
3. FEJLESZTŐI DOKUMENTÁCIÓ.....	16
3.1. Feladatspecifikáció	16
3.1.1. Funkcionalitás	16
3.1.2. Kinézet	17
3.2. Az adatbázis.....	17
3.3. Package Diagram.....	18
3.4. Use Case diagram	20
3.5. Sztereotipikus osztálydiagram	22
3.6. Normál osztálydiagram.....	24
3.7. Sequence Diagram	25
3.8. State Machine Diagram	29
3.9. Deployment Diagram	32
3.10. Tesztesetek.....	34
4. ÖSSZEGZÉS.....	35
4.1. A megvalósult projekt	35
4.2. Továbbfejlesztési lehetőségek	35
5. IRODALOMJEGYZÉK.....	37

1. BEVEZETÉS

A XXI. század jó néhány szűkös évet hozott – többek között a magyar – emberek életébe. Bár a gazdasági világválságot 2008-tól datáljuk (Wikipédia, 2019) fontos emlékeznünk arra, hogy ebben az időszakban amelyik család nem rendelkezett devizahittel, könnyebben átvészelte, vagy akár meg sem érezte magán a világválság hatásait. A „2001. évi XCIII. törvény a devizakorlátozások megszüntetéséről, valamint egyes kapcsolódó törvények módosításáról” szóló jogszabály engedélyezte a devizahiteleket. *„Más kérdés, hogy erre külföldi kapcsolataink miatt valószínűleg mindenképp szükség volt – harmonizálni kellett a jogszabályokat az EU-val, az OECD-vel. Erről elég a törvény első mondatát elolvasni: „Az Országgyűlés a piacgazdaság kiépítésének következményeként, a forint konvertibilitását teljessé téve, az Európai Unióhoz történő csatlakozásra való felkészülés érdekében törvényt alkot a devizakorlátozások megszüntetéséről, a tőke szabad áramlásának elősegítéséről. Ám valószínűleg már akkor lehetett volna szabályozni, korlátozni a devizahitel-felvételt, ahogy azóta meg is tették. Például megakadályozni, hogy elegendő fedezet, önerő nélkül vagy minimális önerővel is hitelt lehessen felvenni. Vagy legalább sokkal jobb tájékoztatásra, felvilágosításra kötelezni a bankokat. Nem így történt.”* (Eidenpenz, 2013)

Ahogy számos ismerősöm is saját bőrén tapasztalta a devizahitelek súlyát – főleg 2006 és 2007-től kezdődően –, úgy én is. Szakdolgozatom bevezetőjének célja nem a felelősök keresése, hanem döntéseink, különösen a pénzügyi döntések fontosságára hívja fel a figyelmet.

Az fentiekben taglalt problémát tovább súlyosbítja az a tény, hogy napjainkban mikor egyre kevesebben szenvednek az adóságok súlya alatt, a munkanélküliség csökken, az életszínvonal nő, könnyebben kiadjuk a pénzt olyan dolgokra, amelyeket nem biztos, hogy megengedhetünk magunknak és ráadásul talán még feleslegesek is, de ez ellen nehéz tenni hiszen a reklámok vágykeltő hatása vitathatatlanul hatásos. Mindeközben hiányzik a cashflow, ami nélkül egy vállalkozás sem tudja anyagi stabilitását megteremteni és megőrizni, bizony fontos ez a magánéletben is az anyagi biztonság eléréséhez. Tisztában kell lennünk a jövőbeni várható kiadásainkkal és bevételeinkkel kapcsolatban minimum egy évre előre, de nagyobb tervek esetén akár az 5-10 tervezett év is bőven szükséges lehet. Nem tudhatjuk, hogy meddig dolgozunk még a jelenlegi munkahelyünkön, ott hogyan alakul majd a bérünk, vagy vállalkozóként milyen évünk lesz jövőre, és a másik oldalon lesz-e rendkívüli kiadásunk egy autós koccanás miatt, vagy elromlik a mosógép és be kell ruháznunk egy új készülékre.

Ellenben jövőre is fizetünk rezsit, telefonszámlát, tankoljuk az autót és étkezünk, valamint fél évente és évente ugyanúgy lesznek visszatérő kiadásink, mint például a gépjárműadó. Ezek a költségek döntő többségükben nem változnak olyan mértéken, hogy ne lehetne őket tervezni, ahogy a bevételünket is, hiszen legalább annyi bevételt elő akarunk teremteni jövőre is mint idén, vagy inkább többet. Az előző gondolatok alapján kijelenthető, hogy az anyagi biztonság megteremtésének fontos eleme a tervezés.

Habár a piacon rengeteg, jobbnál jobb ingyenes és fizetős megoldás fellelhető, a saját elképzeléseimet kielégítő szoftvert még nem találtam ezért választottam szakdolgozatom témájának egy házi költségvetés tervező-könyvelő szoftver elkészítését.

2. FELHASZNÁLÓI DOKUMENTÁCIÓ

2.1. Miért érdemes használatba venni a szoftvert?

A BeKi nevű szoftvert azért hoztam létre, hogy megszabadítsa a felhasználót a franciakockás füzetek és Excel táblázatok kavalkádjától. Nem vagyok ellene egyik eszköznek sem, mindkét példa igazán hasznos tud lenni bizonyos esetekben, de ha azokat választjuk, megfosztjuk magunkat bizonyos funkcióktól. Az egyszerűség és átláthatóság mellett a program képes alap tervezésekben segíteni, értesíteni a felhasználót az esedékességekről, bizonylatokat megőrizni a pénzmozgásokhoz. Lapozhatóvá teszi a hónapokat, legenerálja a várható költségeket, kiadásokat az előző hónapok alapján, de a felhasználó is felépítheti magának a havi tervét.

2.2. A szoftver ablakai

2.2.1. Főablak

A főablakból (1. ábra) érhető el majdnem minden (felvitel, módosítás, törlés, bizonylat műveletek, egyszerű kimutatás, beállítások). Itt tudunk lapozgatni a hónapok közt, látjuk a tárgyható pénzmozgásait is. A jobb felső sarokban látható az adott havi összes bevétel és kiadás összege.

1. ábra
A BeKi főablaka

The screenshot shows the main window of the BeKi application. At the top, there's a header bar with the title 'BeKi' and standard window controls. Below the header, there's a toolbar with various icons for financial operations. The main area is divided into two panels: 'Bevételek' (Income) on the left and 'Kiadások' (Expenses) on the right. Each panel has a table with columns for 'Kategória', 'Megnevezés', 'Összeg', 'Esedékesség', 'Teljesítve', and 'Rendszeresség'. The 'Bevételek' table has one row with the following data: 'Teszt bevétel', 'ez egy rendszeres havi bevét...', '1 000', '2020.02.06.', '2020.02.10.', and 'Havonta'. The 'Kiadások' table is empty.

Bevételek

Kategória	Megnevezés	Összeg	Esedékesség	Teljesítve	Rendszeresség
Teszt bevétel	ez egy rendszeres havi bevét...	1 000	2020.02.06.	2020.02.10.	Havonta

Kiadások

Kategória	Megnevezés	Összeg	Esedékesség	Teljesítve	Rendszeresség
-----------	------------	--------	-------------	------------	---------------

A bal felső sarokban (2. ábra) lévő nyilakkal előre és hátra léphetünk egy hónapot. A hónapot megadhatjuk az „év” és „hó” választó komponensekkel is. Ez a praktikusabb döntés, ha nagyobbakat akarunk lépni.

2. ábra
Mozgás a hónapok között

This close-up shows the date selection controls. It includes two dropdown menus: 'Év:' (Year) set to '2020' and 'Hó:' (Month) set to 'február'. Below these are two buttons with left and right arrows for navigating between months.

Év: 2020

Hó: február

2.2.2. Pénzmozgás műveletek

A „pénzmozgás műveletek” keretben az „Új” gombbal tudunk felvinni új pénzmozgást. Ezt megnyomva megnyílik egy új ablak, (3. ábra) ahol a következő adatokat adhatjuk meg:

Megnevezés	A pénzmozgás tetszőleges neve (nem lehet üres és maximum 64 karakter)
Összeg	A pénzmozgás értéke. (kötelező)
Típus	Ez bevétel, vagy kiadás? (kötelező)
Kategória	Tetszőleges, a felhasználó által definiált kategóriába sorolható a pénzmozgás a legördülő menüből kiválasztva. (kötelező választani)
Esedékesség (első) napja	Az első esedékességi nap, rendszeres pénzmozgás esetén az adott havi esedékesség napja. (kötelező választani)
Teljesítve	Bejelölve megadható a teljesítési dátum.
Teljesítési dátum	Mikor érkezett be, vagy mikor lett kiadva az összeg.
Rendszeresség	A „Nem rendszeres”, eseti pénzmozgást jelent. Ha bármi mást választ a felhasználó, a program felajánlja hónap generáláskor, mint ismétlődő tétel, betartva a megjelölt gyakoriságokat (havonta, negyedévente, fél évente, évente). (kötelező választani)
Csak könyvel	Ha egyszeri kiadás vagy bevétel volt és meg is érkezett, vagy ki is lett adva azonnal az összeg, akkor ezt érdemes bejelölni. Ilyenkor a program kitölti a teljesítési dátumot a kiválasztott esedékességi napra. A rendszeresség ilyenkor nem használható.

3. ábra
Pénzmozgás ablak

Ha minden szükséges adat meg lett adva, az OK gombbal felvitelre kerül az adatbázisba a pénzmozgás, mégsem gombbal megszakítható a felvitel, ilyenkor semmilyen adat nem kerül felvitelre.

Módosítás esetén a főablak bevétel vagy kiadás táblájában, ha van kijelölt elem és a felhasználó a „Módosít” gombra kattint, az előbb ismertetett ablak jelenik meg a választott tétel adataival. A felhasználónak lehetősége van bármit módosítani kivéve, hogy bevétel, vagy kiadás-e a tétel. Az OK gombbal elhagyva az ablakot, a módosítások érvénybe lépnek. Az adott tételen duplán kattintva is megnyitható a módosítás ablak.

A felhasználónak ezen az ablakon keresztül van lehetősége eljutni a „Kategóriák” ablakba (4. ábra). Ez csak innen, a „Pénzmozgás” ablakból érhető el, de a felvitel és módosítás módban is.

Típus kiválasztása	Ez bevétel vagy kiadás kategória? (kötelező)
Megnevezés	A kategória neve. Kötelező adat és az adott típusban csak két azonos nevű kategória nem lehet.
Új	Új kategória felviteli módba vált az ablak.
Módosít	A kijelölt kategória módosítását lehetővé tévő módba vált az ablak.
Töröl	A kijelölt kategória törlését végzi el.
Kész	Az ablak elhagyása.
Felvitel	Az új kategória felvitele. Módosítások felvitele.

Módosítás esetén csak a megnevezést lehet módosítani, típust nem.

4. ábra
Kategóriák ablak

A pénzmozgás törléséhez a főablakban (1. ábra) ki kell jelölni egy tételt és a „Törölés” gombra kattintani. Mielőtt a tétel törlődne, megerősítést kér a program.

2.2.3. Bizonylat

Kijelölve egy tételt és a „Bizonylat” gombra kattintva tetszőleges számú bizonylatot rögzíthetünk a pénzmozgáshoz. Engedélyezett formátumok: png, jpg, pdf. Ebben az ablakban a „Felvitel” gomb megnyomására kitallózhatunk egy engedélyezett formátumú fájlt és felvihetjük az adatbázisba azt.

A „Törölés” a kijelölt tételt törli.

A „Kimentés fájlba” a kijelölt tételből, a felhasználó által kiválasztott helyre kimentti a fájlt.

A tételre történő dupla kattintással a Windows fájl társításnak megfelelő alkalmazás jeleníti meg a bizonylatot. A nyomtatás is az esetleges külső szoftverrel oldható meg.

A „Kész” gombbal elhagyható az ablak.

Amennyiben a főablakban kijelölt pénzmozgáshoz tartozik bizonylat, a gomb szövege zöld színű.

5. ábra
„Nincs mentett bizonylat” és „van mentett bizonylat” a kijelölt pénzmozgáshoz



2.2.4. Kimutatás

Ez az ablak (6. ábra) a „Kimutatás” gombbal érhető el. Megnyílása után az adott év rendezett kiadásait és bevételeit tekinthetjük meg. A táblázatok rendezhetők a fejlécen duplán kattintva. A kijelölt cellák összegét a megfelelő összegző szövegdoboz jeleníti meg.

A form felső részén található az évszámválasztó, mellyel kiválaszthatjuk az áttekinteni kívánt esztendő.

6. ábra
Kimutatás ablak

The screenshot shows a software window titled 'Kimutatás'. At the top, there is a year selector set to '2020'. Below this, the window is divided into two main sections: 'Bevételek' (Income) and 'Kiadások' (Expenses). Each section contains a table with columns for months (január to december) and a total column ('Összesen').

Bevételek (Income) Table:

megnevezés	január	február	március	április	május	június	július	augusztus	szeptember	október	november	december	Összesen
csak könyvel funkció teszt	111												111
egyszeri bevétel - rendezett	100												100
ismétlődő bevétel - havi - re...		5 466											5 466

Kiadások (Expenses) Table:

megnevezés	január	február	március	április	május	június	július	augusztus	szeptember	október	november	december	Összesen
csak könyvel funkció teszt	11												11
egyszeri kiadás - rendezett		654											654
ismétlődő kiadás - rendezett	65 544												65 544

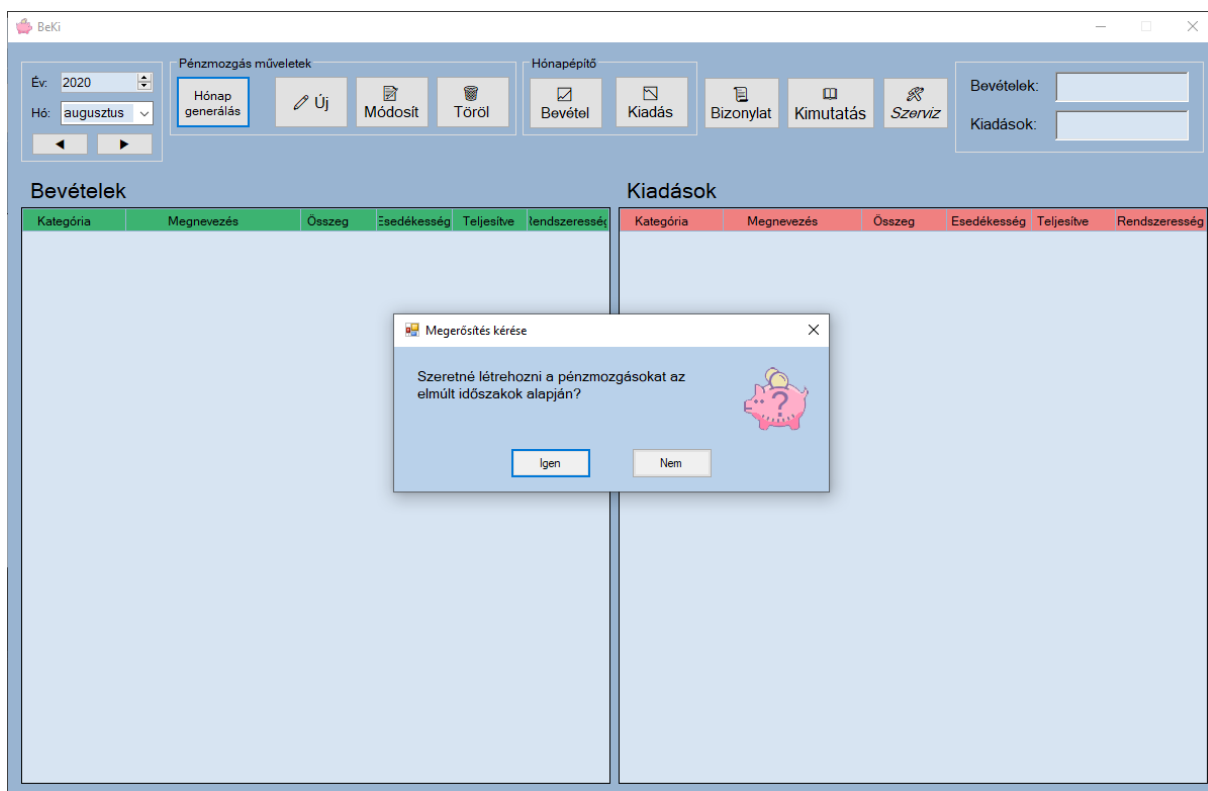
Below each table, there is a large light blue area for additional information or comments. To the right of each table, there is a label 'Kijelölt tételek összesen:' followed by a text box for the total of the selected items.

2.2.5. Hónap generálás

A „Hónap generálás” gombra kattintva (7. ábra) az előző hónapok kiadásai és bevételei alapján automatikusan felépül a főablakban aktuálisan kiválasztott hónap. Ezt a szoftver a pénzmozgások rendszerességeit használva képes elkészíteni függetlenül attól, hogy az adott tétel pénzügyileg rendezett vagy sem, hiszen attól még, hogy nyitott a tétel ugyanúgy esedékessé válik a rendszeressége alapján.

Amennyiben a rendszeresség meg lett szakítva, úgy az többé nem generálódik. Például egy januári, havi rendszerességű tétel ha nem lesz áthozva februárra, vagy februárban törlésre kerül, akkor márciusban már nem generálódik ki.

7. ábra
Hónap generálás gombra kattintva megjelenik a megerősítést kérő ablak



2.2.6. Hónapépítés

A hónap generálás helyett, vagy annak kiegészítésére használható funkció. Annak megfelelően, hogy a hónap bevétel vagy a kiadás tételeit szeretnénk „összepakolgatni”, rá kell kattintanunk a „Kiadás” vagy „Bevétel” gombra. Mindig az a hónap kerül építésre, ami a főablakban ki van választva. A megnyíló ablakban (8. ábra) kiválasztjuk az „Esedékességi időszak választó” komponensével, hogy mely két dátum közé eső esedékességű pénzmozgásokból szeretnénk összeválogatni a tételeket. Ezután a „Betöltés” gombra kattintva betöltődnek a bal oldali listába a kívánt tételek.

8. ábra
Hónapépítő ablak

Hónapépítő

Esedékességi időszak választó

Dátumtól: 2020. 02. 01.

Dátumig: 2020. 02. 29.

Betöltés

Az építés alatt lévő hónap: 2020. február

Hónapépítés

Tételek a választott időszakban →

Kategória	Megnevezés	Összeg	Rendszeresség	Esedékesség
alap bevétel	ismétlődő bevét...	33546	Havonta	2020.02.12.
alap bevétel	ismétlődő bevét...	5466	Havonta	2020.02.12.

A hónap tételei

Kategória	Megnevezés	Összeg	Rendszeresség	Esedékesség
alap bevétel	ismétlődő bevét...	33546	Havonta	2020.02.12.
alap bevétel	ismétlődő bevét...	5466	Havonta	2020.02.12.

OK Mégsem

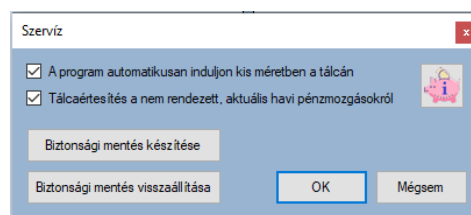
Jelöljük ki egy vagy több tételt és a nyíl gomb segítségével csináljunk az építendő hónapnak megfelelő dátummal létrejövő másolatot! A „kuka” gombbal a jobb oldali tételeket tudjuk törölni, egyszerre többet is, amennyiben többet jelöltünk ki.

Ha végeztünk, az OK gombot megnyomva megerősítést kér a szoftver. Amennyiben az „Igen” gombra kattintunk, úgy bejegyzésre kerülnek a módosítások az adatbázisba. A „Nem” gombbal a figyelmeztetés bezáródik és egyelőre nem történik adatbázis bejegyzés. A „Mégsem” gomb az ablak, hónapmódosítás nélküli elhagyását teszi lehetővé. Ebben az esetben szintén meg kell erősítenünk választásunkat, nehogy a munkánk véletlenül kárba vesszen.

2.2.7. Szerviz

A „Szerviz” ablakban (9. ábra) a megfelelő pipákat elhelyezve, vagy törölve tudjuk a kívánt beállításokat elvégezni.

9. ábra
Szervíz ablak



Két lehetőség kínálkozik:

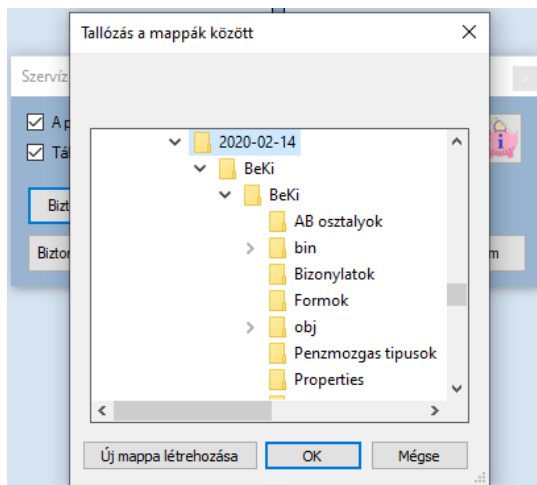
„A program automatikusan induljon kis méretben a tálcán” bejelölésével a következő Windows indulásnál elindul a program és háttérben marad.

A „Tálcaértesítés a nem rendezett, aktuális havi pénzmozgásokról” funkció csak akkor üzemel, ha a program háttérben fut és a hozzá tartozó pipa el van helyezve a jelölőnégyzetben.

Mindkét beállításon történt módosítás az OK gombbal történő ablak elhagyásra lesz jóváhagyva. A „Mégsem” -el módosítás nélkül lépünk ki az ablakból.

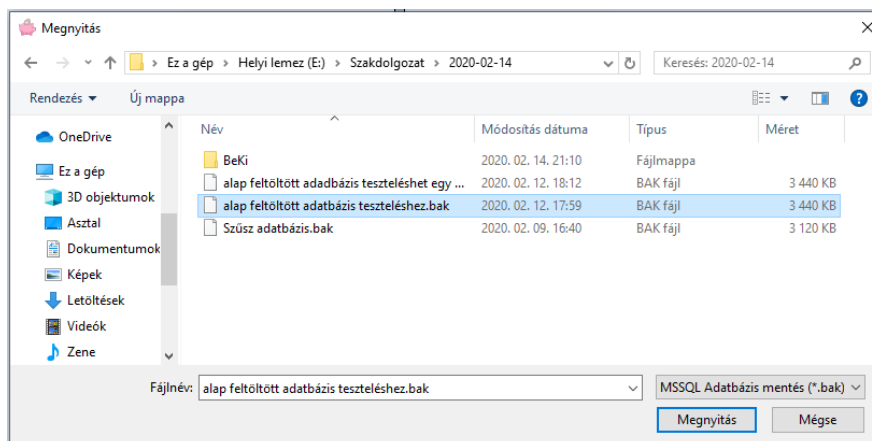
Az OK és a Mégsem gombok nincsenek kihatással az adatbázis mentés és visszaállítás gombbal végrehajtásra kerülő műveletekre.

10. ábra
Biztonsági mentés készítése



Amennyiben biztonsági mentést készítünk az adatbázisról (10. ábra), ki kell választanunk a célmappát, a visszaállításkor (11. ábra) pedig a megfelelő *.bak kiterjesztésű, a program által valamikor korábban létrehozott biztonsági mentést. Ezesetben figyelmeztetés jelenik meg, hogy visszavonhatatlanul lecserélődik minden a program adatbázisában a mentés fájlban lévő adatokra és ez nem visszavonható. Ha a felhasználó bizonytalan, ajánlott egy gyakorlottabb felhasználó segítségét kérnie ehhez a művelethez.

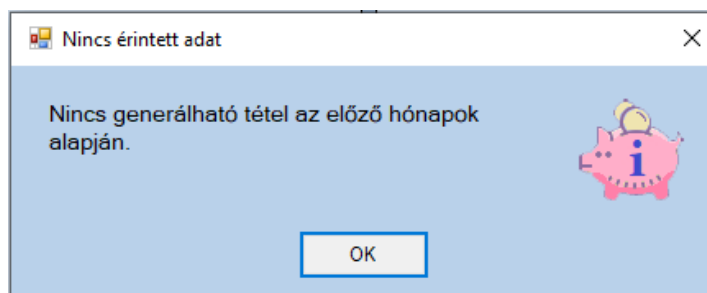
11. ábra
Biztonsági mentés visszaállítása



A „malac” gombra kattintva a program infó jelenik meg.

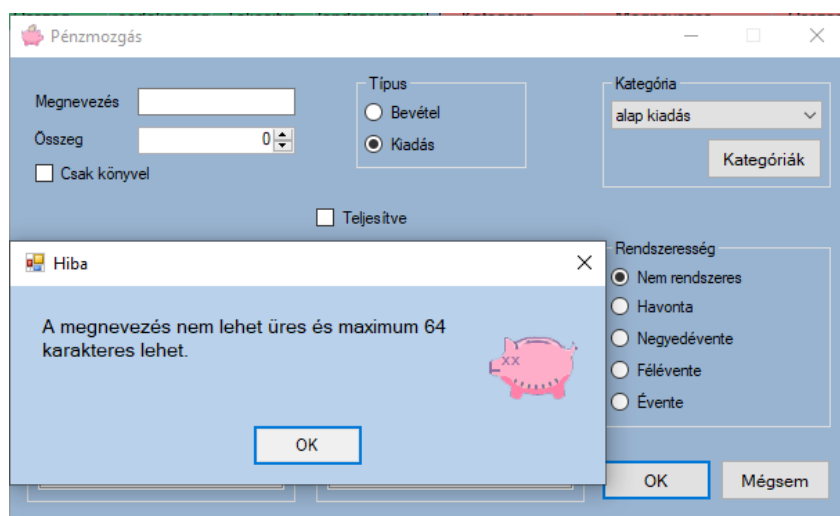
2.3. Hibaesetek

12. ábra
Hónapgenerálási hibaüzenet



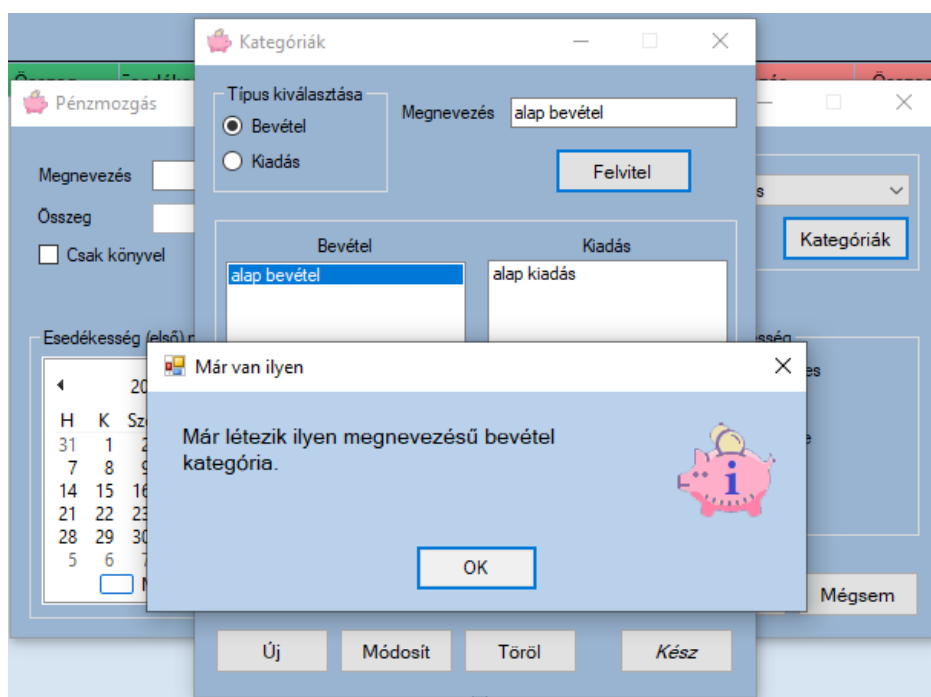
Nincs generálható pénzmozgás tétel az előző hónapok alapján, mert nem is volt ilyen (még), vagy egy előző hónapban meg lett szakítva a folyamatosság (12. ábra).

13. ábra
Túl rövid, vagy túl hosszú megnevezés



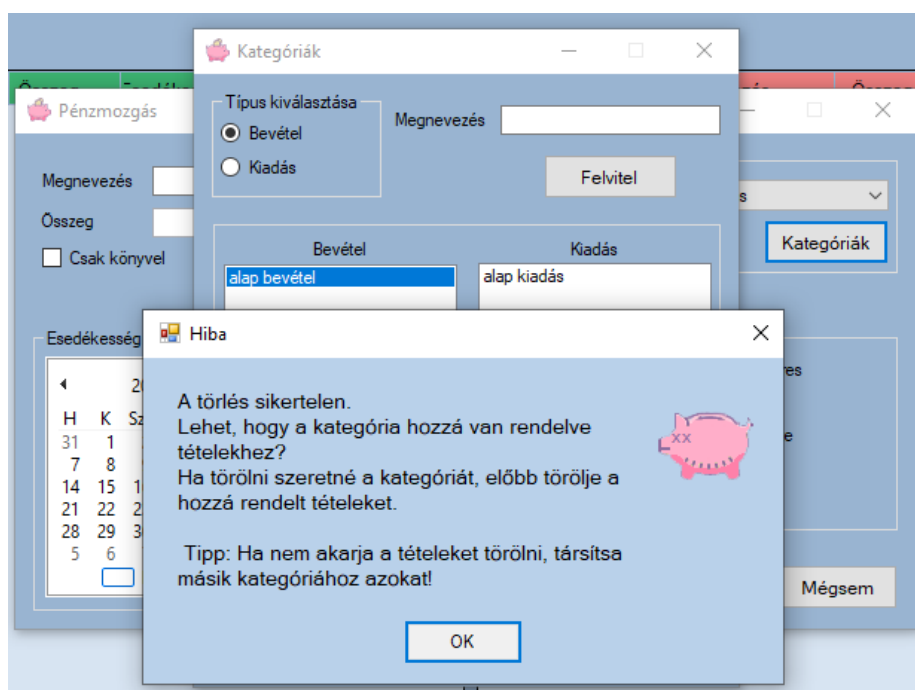
A megnevezés melletti szövegdobozba nem írt be a felhasználó megnevezést, vagy 64 karakternél hosszabb lett az (13. ábra).

14. ábra
Létező kategória megnevezés hibaiüzenet



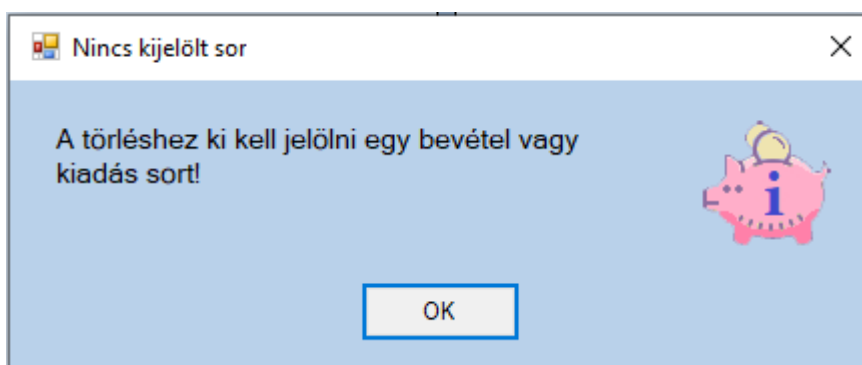
Ha már létezik a megadott megnevezés az adott típusban, akkor nem vihető fel újra ugyanaz.

15. ábra
Olyan kategória törlésének kísérlete, melyhez tartozik pénzmozgás



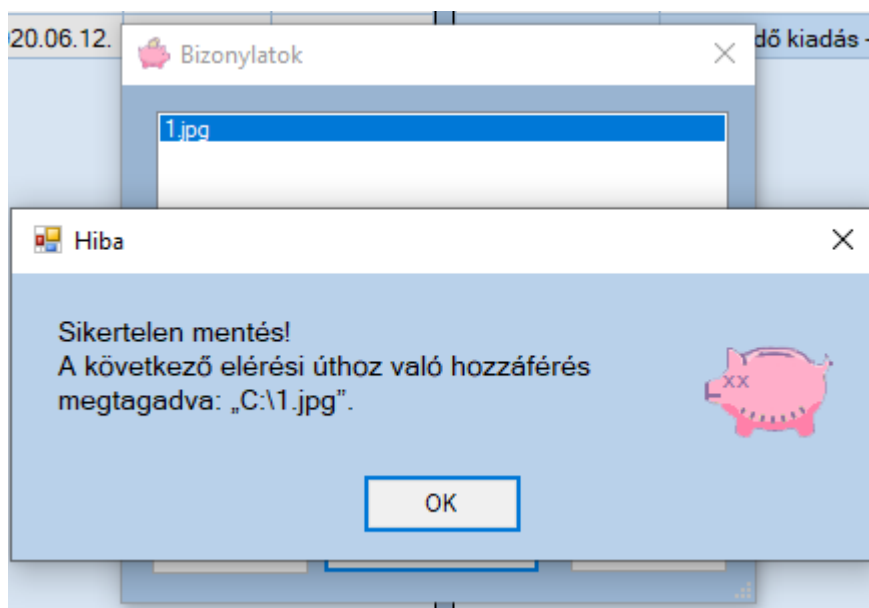
Nem törölhető olyan kategória melyhez pénzmozgás tartozik (15. ábra), mivel pénzmozgás nem létezhet kategória nélkül. Ha törölhető lenne kategória nélküli pénzmozgás jöhetne létre.

16. ábra
Pénzmozgás törlési kísérlet hibaiüzenete



Pénzmozgás törlésének kísérlete a főablakban. (16. ábra) A felhasználó nem jelölt ki tételt. Lehet, hogy nincs is kijelölhető tétel.

17. ábra
A bizonylat kimentéséhez választott útvonal nem írható



Nem megy végbe a bizonylat adatbázisból történő fájlba kimentése, ha a felhasználó olyan útvonalat választ, mely nem írható, vagy nincs a felhasználónak jogosultsága írni azt. (17. ábra)

2.4. Futtatási rendszerkövetelmények

A szoftver MSSQL lokális adatbázisban tárolja az adatokat, ezért szükséges a Microsoft SQL Server Express telepítése.

Minimális szoftverkövetelmények:

Windows 10

Microsoft SQL Server Express 2016 (13.1.4001.0)

Minimális hardverkövetelmények:

CPU: 1 GHz

RAM: 2 GB

Merevlemez-terület: 100 MB üres lemezterület (főleg az adatbázis mérete határozza meg)

Képernyőfelbontás: 1280 x 1024

Ajánlott hardverkövetelmények:

CPU: 3 GHz

RAM: 4 GB

Merevlemez-terület: 300 MB üres lemezterület (főleg az adatbázis mérete határozza meg)

Képernyőfelbontás: 1920 x 1080

3. FEJLESZTŐI DOKUMENTÁCIÓ

3.1. Feladatspecifikáció

3.1.1. Funkcionalitás

A cél egy olyan Windows operációs rendszer alatt futó grafikus szoftver elkészítése melyben könyvelhetők és tervezhetők a havi bevételek és kiadások. Az adatok tárolására MSSQL adatbázis legyen alkalmazva. A bevétel és kiadás oldali pénzmozgásokra a továbbiakban leegyszerűsítve „pénzmozgás” szóval hivatkozom.

A programban legyen lehetőség pénzmozgásokat létrehozni, melyek kategóriákba sorolhatók. A kategóriákat a felhasználó definiálja. Egy kategória csak bevétel vagy kiadás lehessen. Egyszerre a kettőt nem képviselheti. Minden pénzmozgás esetén kötelező lesz annak irányát (bevétel, kiadás), kategóriáját, esedékességét, vagy esedékességét és teljesítését, összegét megadni. Legyen lehetőség definiálni az adott pénzmozgás rendszerességét, amennyiben ismétlődő, de tudjon a felhasználó eseti pénzmozgásokat is rögzíteni.

A pénzmozgások módosítása és törlése ne legyen tiltva, hiszen otthoni felhasználásra van, így a felhasználó csak magának tartozik elszámolással s nem egy cégnek, ezért ebből a szempontból nem kötelező a felhasználói jog korlátozása.

Az egyes pénzmozgásokhoz lehessen bizonylatot rögzíteni úgymint bérlap, vásárlás nyugtája, jótállási jegy. Ezeket a program tárolja el az adatbázisban. Lehessen onnan fájlként újra kimenteni, vagy csak megtekinteni. A program jelezze valamilyen módon, ha a pénzmozgáshoz bizonylat tartozik.

A programban fontos, hogy a pénzügyi tervezést segítse, ezért egy hónapot kiválasztva, legyen a felhasználónak lehetősége az előző hónapokból azt kigenerálni a rendszerességek alapján, amennyiben azok rendszeressége egy előző esedékességkor nem lett megszakítva.

Kényelmi funkcióként, egy tetszőleges időintervallumba eső pénzmozgások alapján is lehessen „felépíteni” a hónap tervét.

A felhasználónak legyen lehetősége eldönteni, hogy szeretné-e, ha a program magától indul a Windows rendszerrel, vagy manuálisan, ő szeretné az elindítani.

A program fusson kis méretben a tálcán (systray), ahol időnként jelenítsen meg értesítéseket a tárgyhóban még nem rendezett pénzmozgásokkal kapcsolatban. Ez a funkció legyen opcionális.

Hogy az adatokat biztonságban tudhassuk, meg kell teremteni adatbázis biztonsági mentésének lehetőségét, illetve a mentés visszaállítását is.

3.1.2. Kinézet

Mivel nem feltétlenül csak gyakorlott, számítógépes programokban jártas felhasználók részére készül a program, így a cél az egyszerű, letisztult, érthető kinézet kialakítása, ahol a funkciók elérése minél kevesebb lépésben történhet meg.

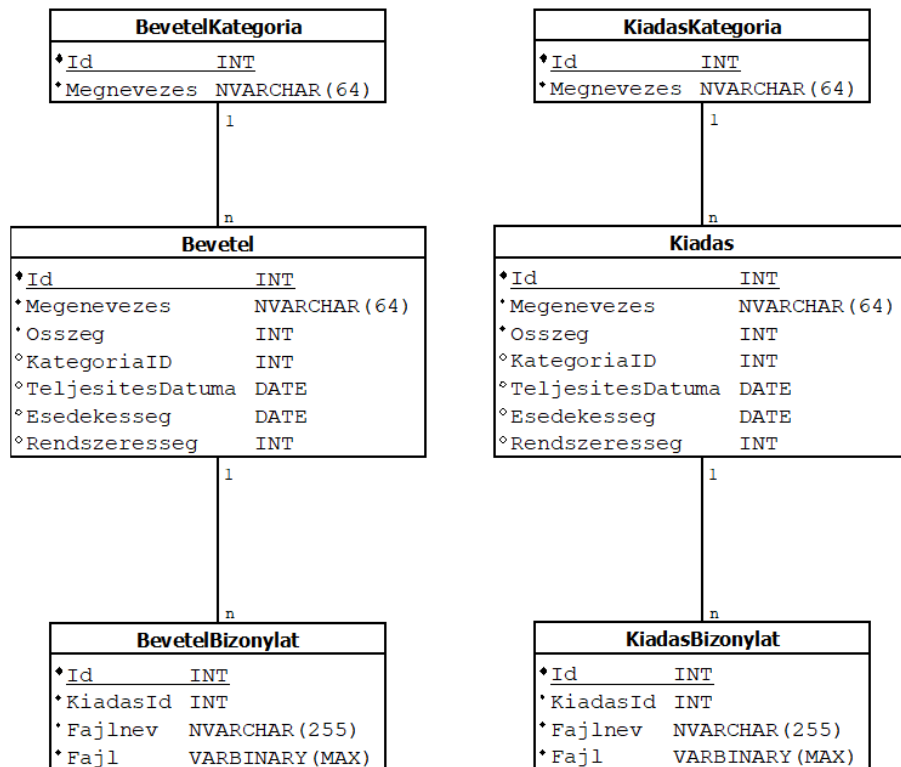
Használata szintén legyen jellemző az egyszerűség a komponensek sorrendje kövesse a használati logikát!

3.2. Az adatbázis

A program egy egyszerű, de a célnak tökéletesen megfelelő MSSQL adatbázisban végzi az adatok tárolását. Az alábbi ER diagramon (18. ábra) látszik, hogy a pénzmozgás típusok külön táblában vannak tárolva. Bár azonos attribútumok esetén a két reláció helyett egy tábla is elegendő lehetne egy plusz oszlop bevezetésével, mely gondoskodna az összegek előjeléről, mégis jobbnak tartom a külön táblázatokban történő elhelyezést a későbbi továbbfejleszthetőség biztosítása érdekében, melynek során létrejöhet olyan eset, mikor az egyik pénzmozgási irány igényli egy újabb tulajdonság hozzáadását, ami a másikra nem jellemző, vagy csak épp nem releváns, nem szükséges tárolni azt. A két tábla használata csökkenti a hibalehetőséget a szoftver írása közben, ha a programkódban a T-SQL utasításban párhuzamot mutat a tábla neve a pénzmozgás fajtájával, minthogy egy bevezetett BIT attribútum megadásával kellene jelölni a fajtát.

Egy „BevetelKategoria” több „Bevetel” -hez tartozhat és egy „Bevetel” -hez több BevetelBizonylat. A „Bevetel” táblában a „KategoriaID”, a „BevetelBizonylat” -ban a „BevetelId” tölti be az idegen kulcs szerepét. A kiadási oldalon ugyanez a logika.

18. ábra
Az adatbázis, adatok tárolását végző tábláinak ER diagramja



Az beállítások tárolása is adatbázisban egy erre létrehozott relációban történik. Ez alól kivétel lesz a program automatikus indulását szolgáló beállítás, mely a Windows regisztrációs adatbázisába tesz bejegyzést és a beállítások bárbeszédpanel megnyitásakor onnan történő lekérdezés alapján határozza meg a jelölőnégyzet állapotát.

19. ábra
A beállítások tárolása az adatbázisban

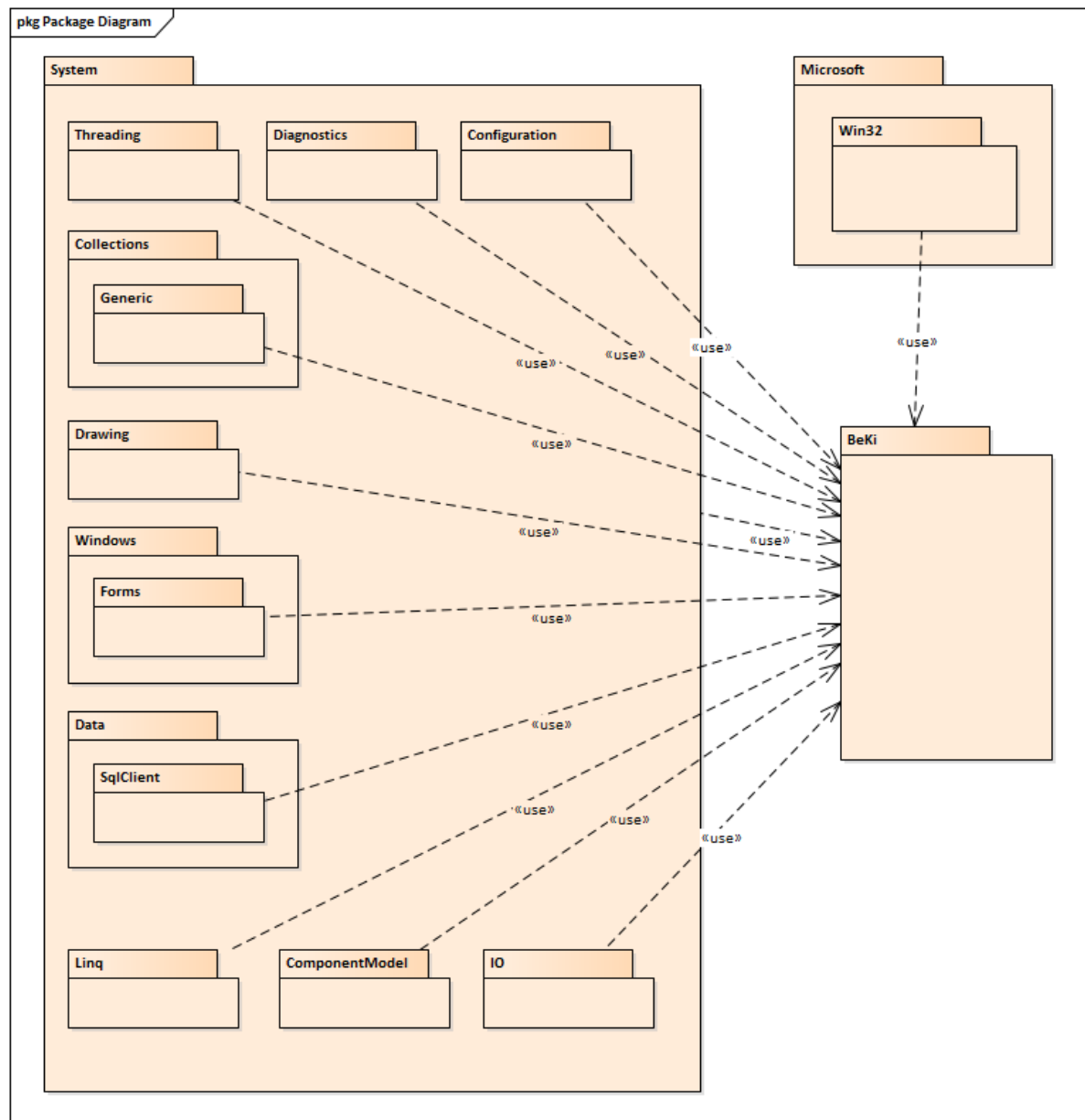
Beallitasok	
*Beallitas	NVARCHAR(64)
*Bekapcsolva	BIT

3.3. Package Diagram

A program futásához célszerű a .NET adta lehetőségek kiaknázása mely a keretrendszer részét képző más névterekben található osztályok, funkciók, metódusok felhasználása.

A package diagram szemlélteti a felhasználandó részegységeket (20. ábra).

20. ábra
Package Diagram



System.Threading namespace: Behívása szükséges a Timer osztály használatához, melyből létrehozott objektum fog felelni az értesítések időnkénti megjelenítéséért.

System.Collections.Generic: Interface-eket és osztályokat tartalmaz, az erősen típusos gyűjtemények kezeléséhez szükséges.

System.Drawing: Az értesítési területen megjelenő ikon és egyéb GDI műveletek végzéséhez elengedhetetlen.

System.Windows.Forms: A grafikus felhasználói felület létrehozásához kell.

System.Data.SqlClient: Az adatbázis kapcsolódás és minden CRUD művelet elvégzését is lehetővé tévő osztályokat tartalmazó namespace.

System.Linq: Nyelvi integrált lekérdezéseket teszi lehetővé.

System.ComponentModel: „*A névtér olyan osztályokat biztosít, amelyeket az összetevők és a vezérlők futási és tervezési idejű viselkedésének megvalósításához használnak. Ez a névtér tartalmazza az alaposztályokat és az interfészeket az attribútumok és a típus konverterek megvalósításához, az adatforrásokhoz történő kötéshez és az engedélyezési alkotóelemekhez.*”
(Microsoft, dátum nélk.)

System.IO: Fájl és mappaműveletek, adatfolyamok, és az azokkal kapcsolatos hibák kezeléséhez szükséges osztályokat egybegyűjtő névtér.

Microsoft.Win32: A regisztrációs adatbázissal kapcsolatos műveletvégzés és fájl dialógus ablakok használatához elengedhetetlen.

3.4. Use Case diagram

Ezen a diagramon a programfunkciókat és azok működését láthatjuk madártávlatból. Jól kivehetők a felhasználói interakciók belépési pontjai, illetve, hogy az adott funkció milyen további folyamatokat indít, vagy indíthat el.

A szoftver Use Case diagramján (21. ábra) a sötétebb színnel jelölt ovális elemek mutatják, hogy a felhasználó merre „tud elindulni”. Az első két eset (Pénzmozgás- és Kategória műveletek) létrehozás, módosítás, törlés műveleteket mutat be a hozzájuk tartozó, felhasználónak szóló hibaüzenetekkel melyet az adott eset kiválthat (extend). A generalizációs nyilakkal becsatolt elemek tulajdonképpen kiegészítik azt az esetet, ami felé mutatnak.

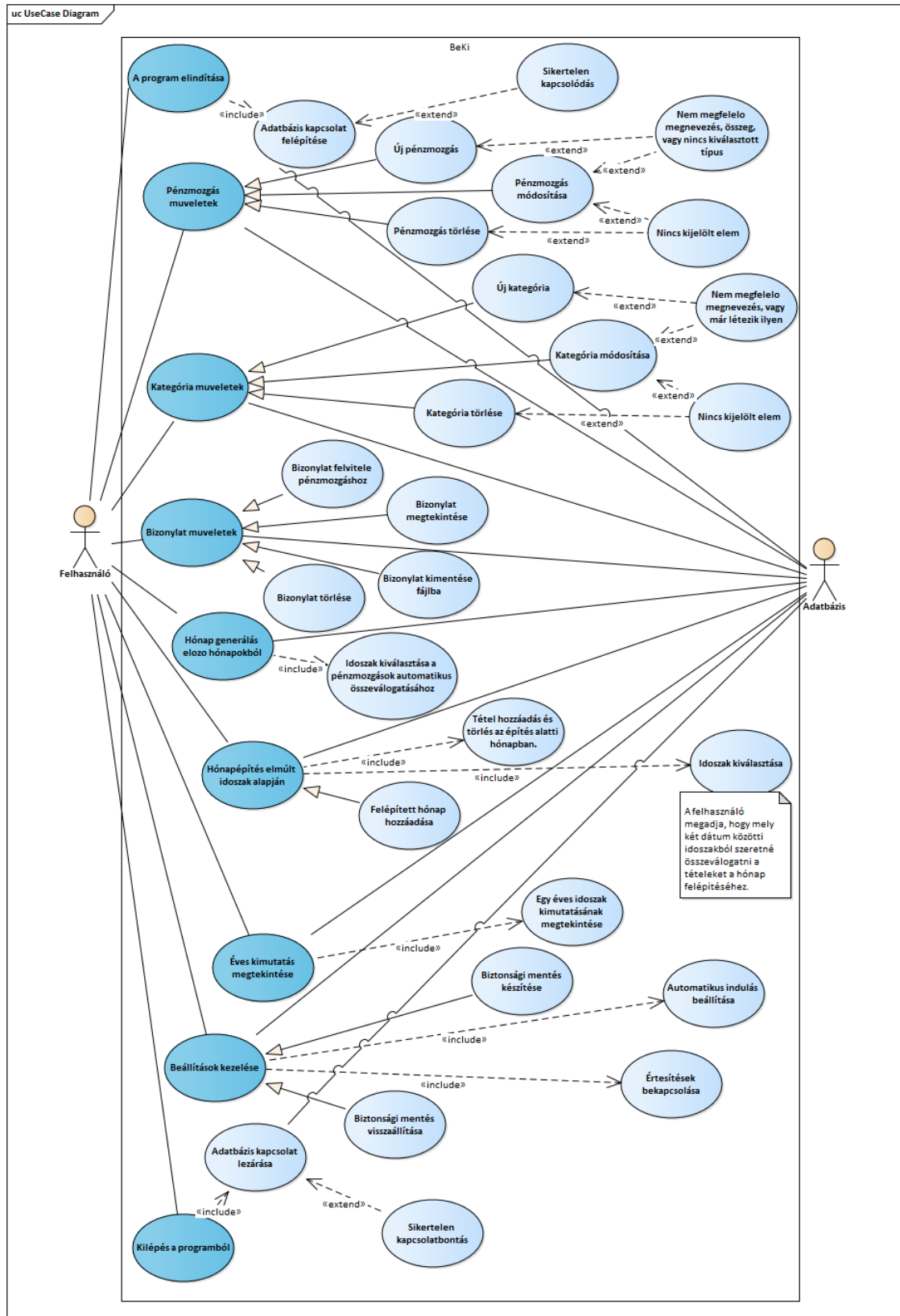
A „Bizonylat műveletek” után négy további eset lehetséges, de egyszerre természetesen csak egyet tud kiváltani a felhasználó. Itt bizonylat vihető fel az egyes pénzmozgásokhoz, akár több is, szintén a törlés, megtekintés és a fájlba történő kimentés lehetősége is megjelenik.

A hónap generálásnál a felhasználónak csak ki kell választania, hogy melyik hónap költségei generálódjanak ki az elmúlt időszak alapján, a többit a program elvégzi.

A hónapépítésben a felhasználó kiválaszthatja, hogy milyen időszakból kívánja összeválogatni a pénzmozgásokat az új hónap felépítéséhez. Ezt mutatja egy include nyíl. A másik include jelzi a lehetőségét a hónap hozzáadás előtti szerkesztéséhez. Itt hozzáadni és törölni tud tételt a felhasználó. A generalizációs nyíl lehet akár egy a formon lévő OK gomb, mely végrehajtja az hónap létrehozását és felvitelét az adatbázisba, ami az előbb ismertetett folyamatban még csak egy listaként létezik.

Az éves kimutatás egy újabb form lesz, melyen a felhasználó által kiválasztott év minden teljesített pénzmozgása áttekinthető formában megjelenik.

21. ábra
Use Case Diagram



A beállítások kezelése szintén egy form. A generalizációs nyilak egy az adatbázis mentés és egy visszaállítás funkció által végrehajtandó műveleteket csatolják, a két include nyíl egy adatbázisban mentendő (értesítések) és egy Windows regisztrációs adatbázisba bejegyzésre kerülő (automatikus indulás) opciót mutat.

A program induláskor kapcsolatot létesít, leállításkor bontja azt az adatbázissal. A diagramon az adatbázis aktor már nem a program indulásához és leállításához van kötve, hanem az ezek által végrehajtandó kapcsolat létesítéséhez és annak felbontásához.

3.5. Sztereotipikus osztálydiagram

Ez a diagram a szoftver működéséhez szükséges osztályok kapcsolatainak megtervezését segíti, ezen felül látható a típusuk, rendeltetésük is.

Az általam készített diagramon (22. ábra) boundary, control, entity típusú osztályok is fellelhetők.

A boundary osztályok a felhasználóval való kapcsolattartásért felelősek, ezek tulajdonképpen „formok”. Egy fő formmal (BeKi) indul a program, arról érhető el az összes többi form, illetve a pénzmozgási adatokkal (bevételek, kiadások) kapcsolatos műveletek végzéséhez kapcsolatban kell lennie egyes control osztályokkal.

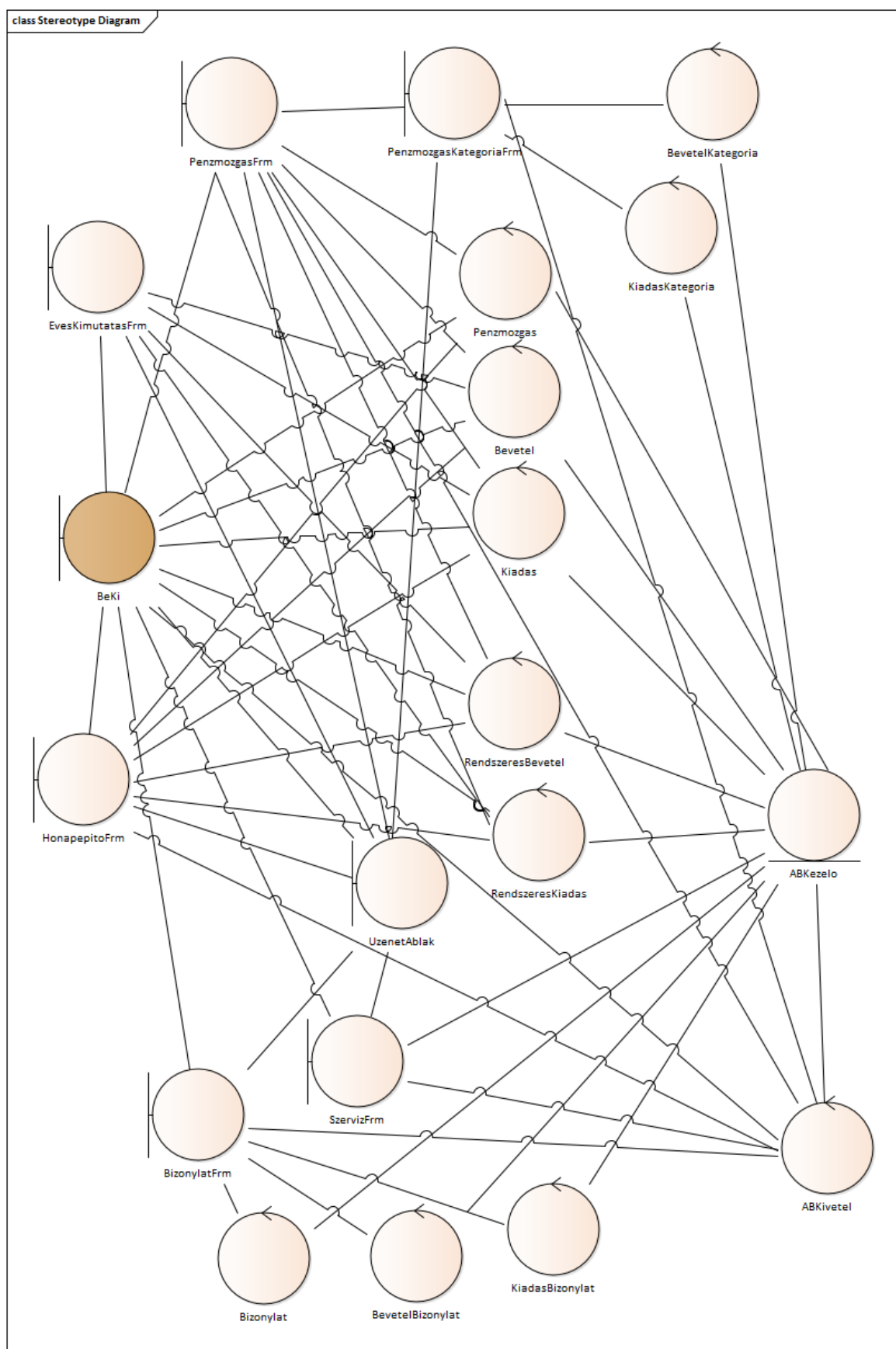
A PenzmozgasFrm, EvesKimutatasFrm, HonapepitoFrm ugyanazzal az 5 control osztállyal van összecsatolva, az azokkal történő műveletvégzések miatt. A PenzmozgasFrm-ről érhető el a pénzmozgás kategóriákkal kapcsolatos műveletekért felelős PenzmozgasKategoriaFrm.

A SzervizFrm a program beállításait gyűjti egybe, illetve az adatbázis mentés és visszaállítás is innen lesz elérhető, ezért kapcsolata van az ABKezelo-vel.

Az UzenetAblak a .NET saját párbeszédpanelét hivatott leváltani az arcukat könnyebb testreszabhatóságának céljából, ezért ez minden boundary osztállyal kapcsolatban kell legyen.

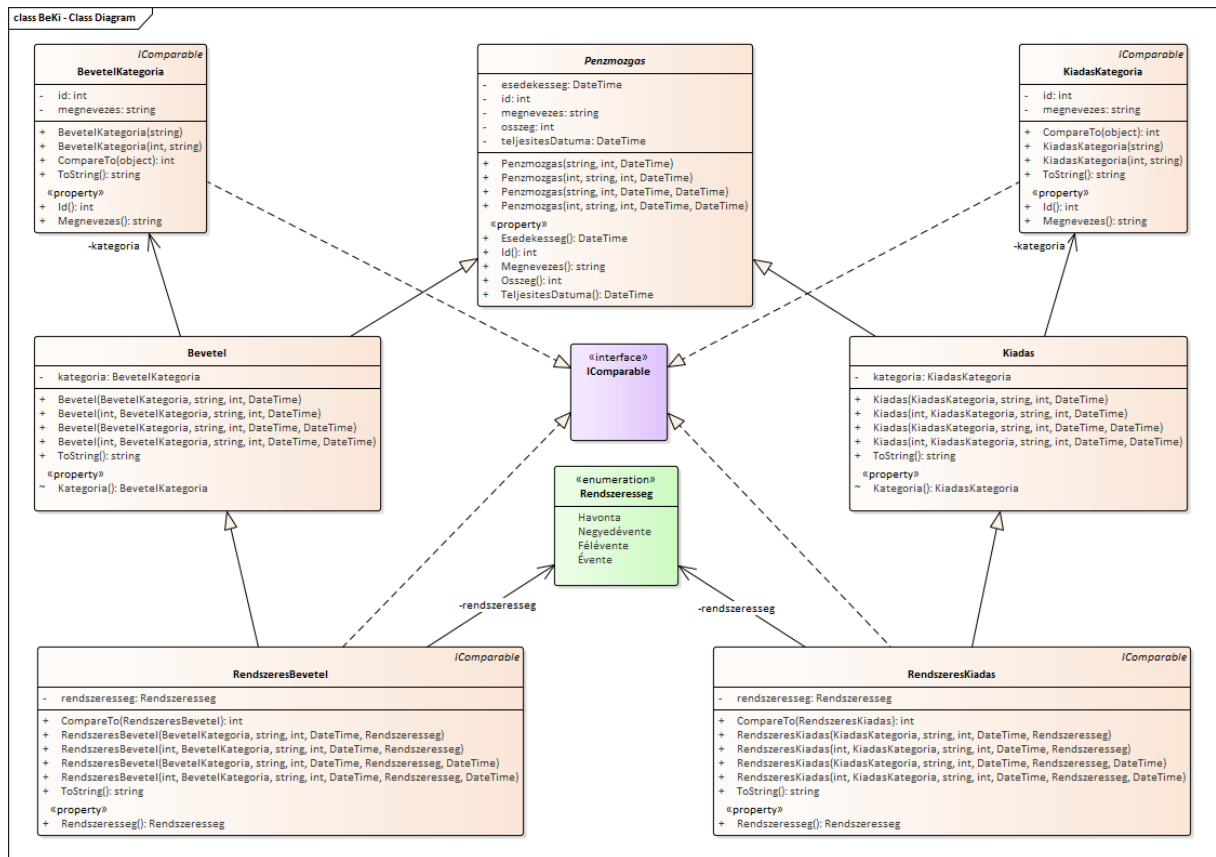
A BizonylatFrm bármely pénzmozgáshoz képes kell legyen bizonylatot társítani, ezért szükséges az ABKezelo-vel való kapcsolatát biztosítani.

22. ábra
Stereotype Class Diagram



3.6. Normál osztálydiagram

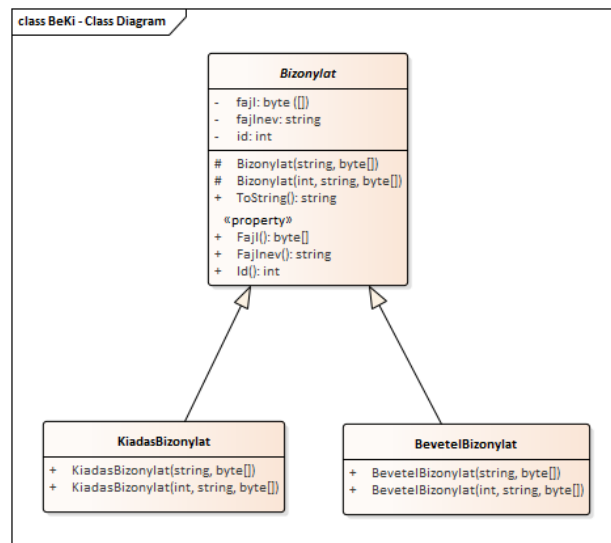
23. ábra
Class Diagram I.



A normál osztálydiagram (23. ábra) szorul a legkevesebb magyarázatra. Jól látható a leszármazás a Penzmozgas (absztrakt!) osztályból Bevetel és Kiadas irányba, melyek azonnal bővülnek a kategóriába sorolhatóság lehetőségével. Ezután újabb altípusok jönnek létre „Rendszeres” előtaggal melyek a Rendszeresseg enummal bővülnek. Ez az öröklődéssel történő típusmegkülönböztetés a program funkcionalitásában kulcsfontosságú lesz.

A Bevetel- és KiadasKategoria illetve a RendszeresBevetel és RendszeresKiadas osztályok megvalósítják az IComparable interface-t.

24. ábra
Class Diagram II.



A Bizonylat absztrakt osztállyal kapcsolatos részt külön ábrán jelenítem meg (24. ábra), mivel közvetlen, „objektum szintű” kapcsolatba nem kívánom hozni a többi osztállyal, de az adatbázisban idegen kulcs kötéssel szükséges kapcsolni a pénzmozgáshoz. A Bizonylat szintén kétfelé örököltet, Kiadas- és BevetelBizonylat irányba.

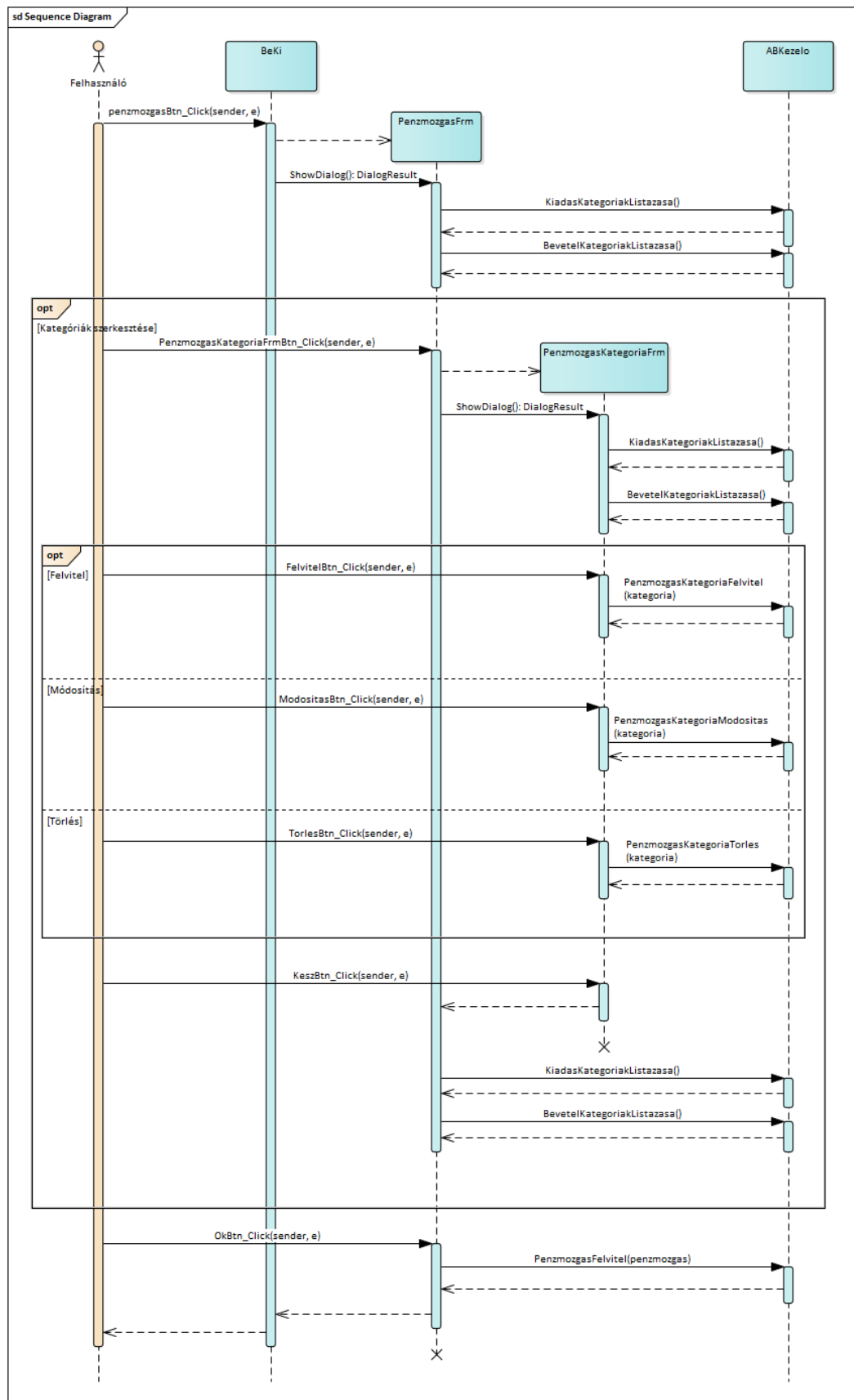
3.7. Sequence Diagram

Az alábbiakban 3 eset kerül bemutatásra az osztályok egymás közti kommunikációjával kapcsolatban. Erre a sequence diagram a legjobb választás, mert függőlegesen a relatív időt tudjuk érzékeltetni, vízszintesen pedig az, hogy épp mi történik.

Az első diagramon (25. ábra) egy pénzmozgás felvitelt mutatok be, opcionálisan elvégezhető lépésekkel együtt.

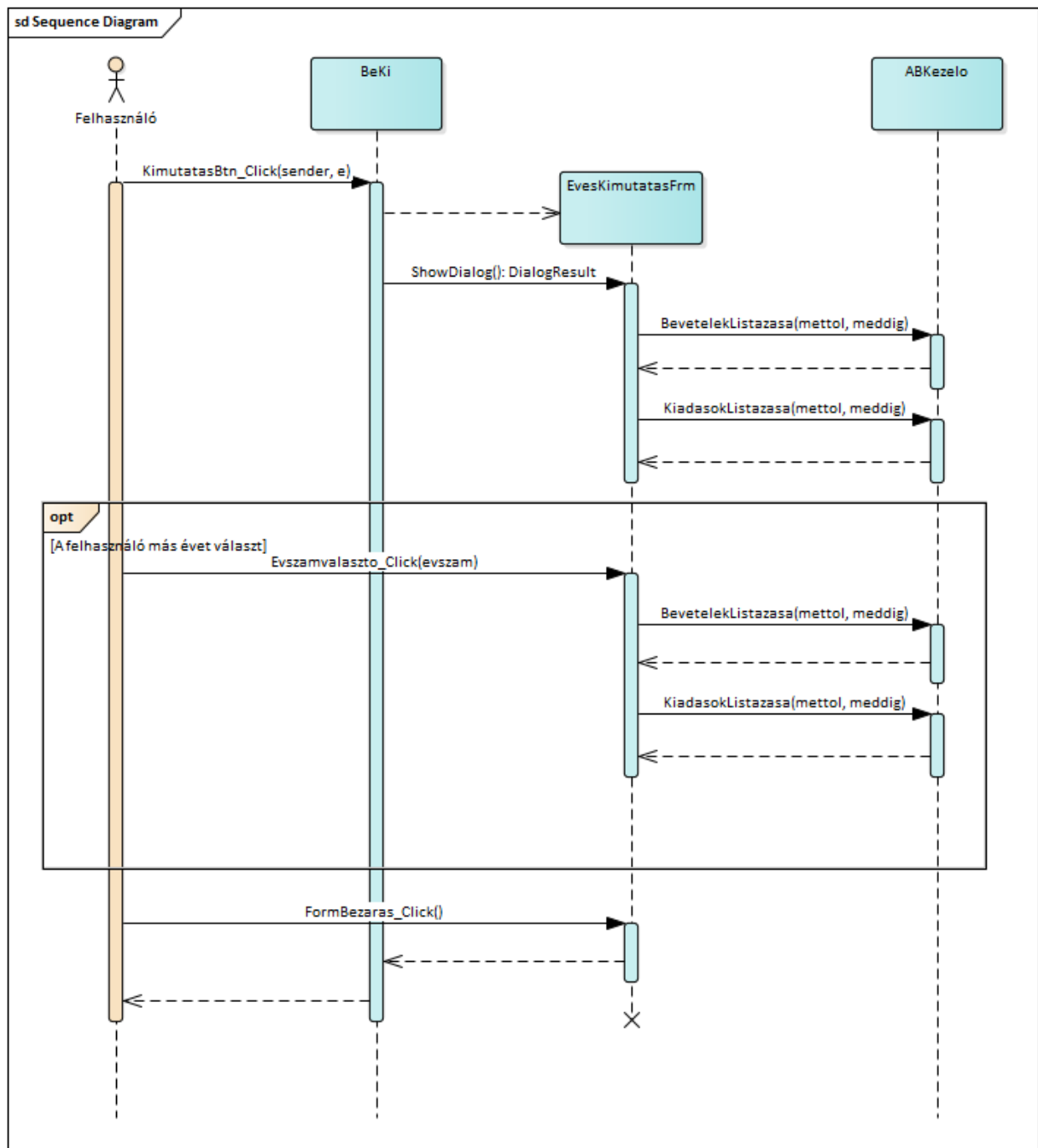
A felhasználó kattint a pénzmozgás felvitelére gombra, amely példányosítja a megfelelő formot a művelet elvégzéséhez. A form lekérdezi az adatbázisból az abban jelenleg megtalálható bevétel és kiadás kategóriákat a megfelelő függvény segítségével. Ezután a felhasználónak lehetősége van további kategóriákat felvinni, meglévőket módosítani, vagy törölni. Ehhez az aktuális form egy újabb formot kell példányosítson, ami erre a feladatra hivatott. Amennyiben a form bezárásra kerül, úgy a pénzmozgás form újra lefrissíti a kategóriákat a listáin, mivel lehet, hogy módosultak, újak készültek, vagy épp törölve lettek. Az előbbi opcionális művelet után a felhasználó felviszi az adatokat és az OK gomb megnyomásával bejegyzésre kerül az adatbázisba a pénzmozgás.

25. ábra
Sequence diagram – Pénzmozgás felvitele

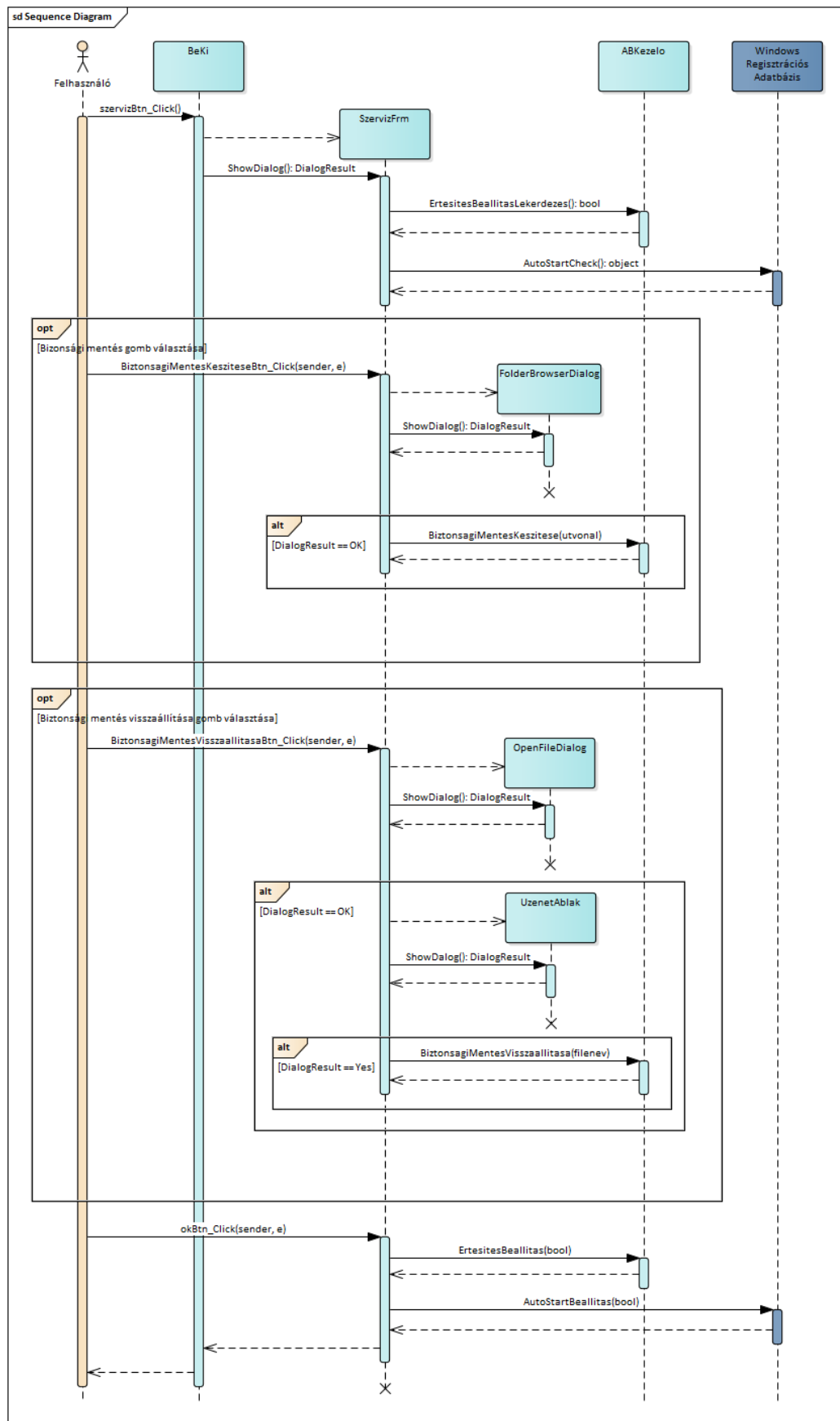


A második diagram (26. ábra) az éves kimutatást megjelenítő form, ahol a hasonlóan az előzőhöz, a fő formon (BeKi) kattint a felhasználó és példányosításra kerül az „EvesKimutatasFrm”. Ez frissíti a két listát a megfelelő függvényekkel, melyek az adatbázisból betöltik az adott évben rendezett pénzmozgásokat. Az adatok megjelenése után választhat más tárgyét a felhasználó, melyre újfent lefrissülnek a listák és megjelennek az adatok.

26. ábra
Sequence diagram – Éves kimutatás



27. ábra
Sequence diagram – Szerviz ablak



A harmadik diagram (27. ábra) a szerviz, vagy más néven beállítások és az adatbázis biztonsági mentését lehetővé tévő form működését szemlélteti. Két checkbox szükséges a beállítások elvégzéséhez és azok állapotának kijelzéséhez. Ezek megfelelő állapotba hozását példányosítás után elvégzi a form a szoftver lokális adatbázisából és a Windows regisztrációs adatbázis megfelelő bejegyzéseit lekérdezve. Mielőtt ezek átállítása után az OK gombra kattintana a felhasználó, lehetősége nyílik a biztonsági mentés elvégzésére, vagy annak visszaállítására az alábbiak szerint.

Biztonsági mentés létrehozásának esetén a form példányosít egy FolderBrowserDialog-ot, amin keresztül kitallózza a mappát, ahova a mentést szeretné letárolni.

A biztonsági mentés visszaállítása választásakor OpenFileDialog jelenik meg a felhasználó válaszára várva. Amennyiben kiválasztja a megfelelő fájlt egy párbeszédpanel fogadja, mely megerősítést kér, mivel a művelet minden adatbázisban lévő adatot a mentés file-ban lévő adatra cserél le.

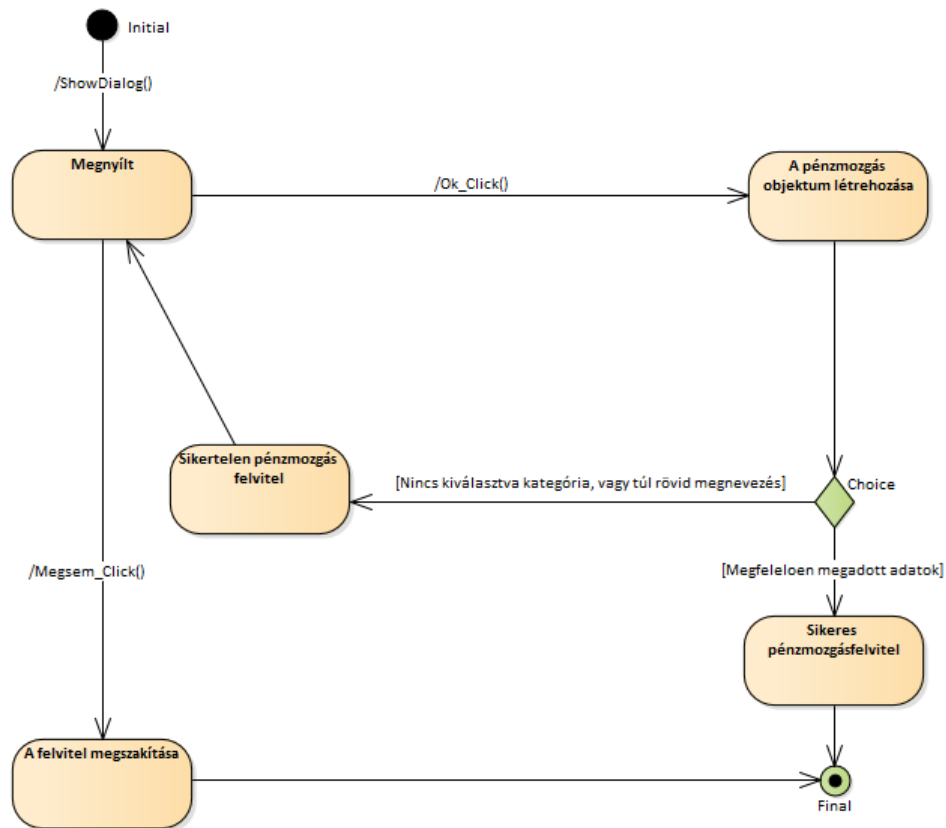
Az OK gombra kattintva a kiválasztott beállítások a megfelelő helyre bejegyzésre kerülnek, majd bezáródik a form.

3.8. State Machine Diagram

A következő diagramokkal egy-egy form működését mutatom be. Itt nem az objektumok egymás közötti kommunikációt, hanem a belső működésüket kell megtervezni. Ez a tervezési módszer igazán praktikusnak mondható, mert a jól átgondolt és megtervezett működés rengeteg plusz munkától és újratekintéstől óvja meg a fejlesztőt. Az objektum határait a belépési (initial) és kilépési (final) pontok jelölik.

Az első diagramon (28. ábra) egy új pénzmozgás felvitele történik. A form betöltődése után a felhasználó kitölti a szükséges mezőket, kiválasztja a rádiógombokat, checkboxokat és elvégzi további komponensek által biztosított, opcionális és kötelező műveleteket. A diagramon látható, hogy az OK gombra kattintva a form kísérletet tesz az objektum (Penzmozgas) létrehozására. Ha minden adat megfelelő, bejegyzésre kerülnek az adatbázisba, de ezt egy másik osztály végzi, így ez nincs külön részletezve a diagramon. A sikeres felvitel után a form bezáródik. Ha a kapcsolaton feltüntetett problémákba ütközik a form, sikertelen műveletre hivatkozva a „megnyílt” állapotba kerül, de a beírt adatok nem törölődnek róla, hogy a felhasználónak ne kelljen előről kezdenie, csak a hibás vagy hiányzó adatot javítania, pótolnia. A „Mégsem” gomb bezárja a formot.

28. ábra
State Machine Diagram – Pénzmozgás felvitele



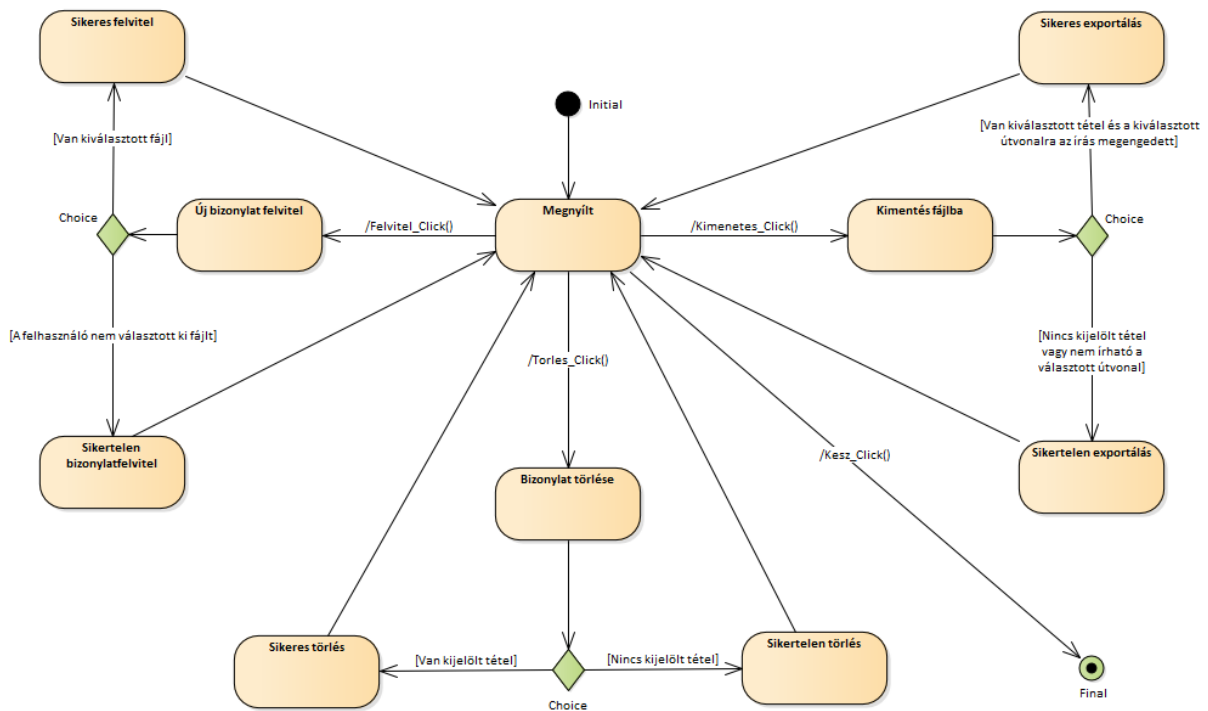
A következő diagramon (29. ábra) a bizonylattal kapcsolatos műveletek láthatók. A diagram szerint a felhasználónak 3 lehetősége van: bizonylatot rögzít, töröl, vagy kimentheti fájlba. A felvitel esetén a felhasználó által kitallózott, megfelelő típusú fájl kerül felvitelre. Ha a felhasználó nem választ ki egyetlen fájlt sem, a felvitel sikertelen lesz. Az, hogy a felhasználó esetleg nem megfelelő fájltypust választ, ez esetben nem releváns hibalehetőség, mert a program szűrt listát mutat a fájlokról, így ez hibaként nem merülhet fel.

A törlésnél, ha van kijelölt tétel akkor a törlés megtörténik, ha nincs akkor természetesen nem.

A bizonylat, adatbázisból fájlba történő kimentése akkor hiúsul meg, ha nincs kijelölt tétel, vagy a felhasználó olyan mappát adott meg célként, ami nem írható. Például C:\ gyökérmappa, vagy egy lezárt optikai lemezre mutató útvonal.

A „Kész” gombra való kattintással zárható be a form.

29. ábra
State Machine Diagram – Bizonylattal kapcsolatos műveletek

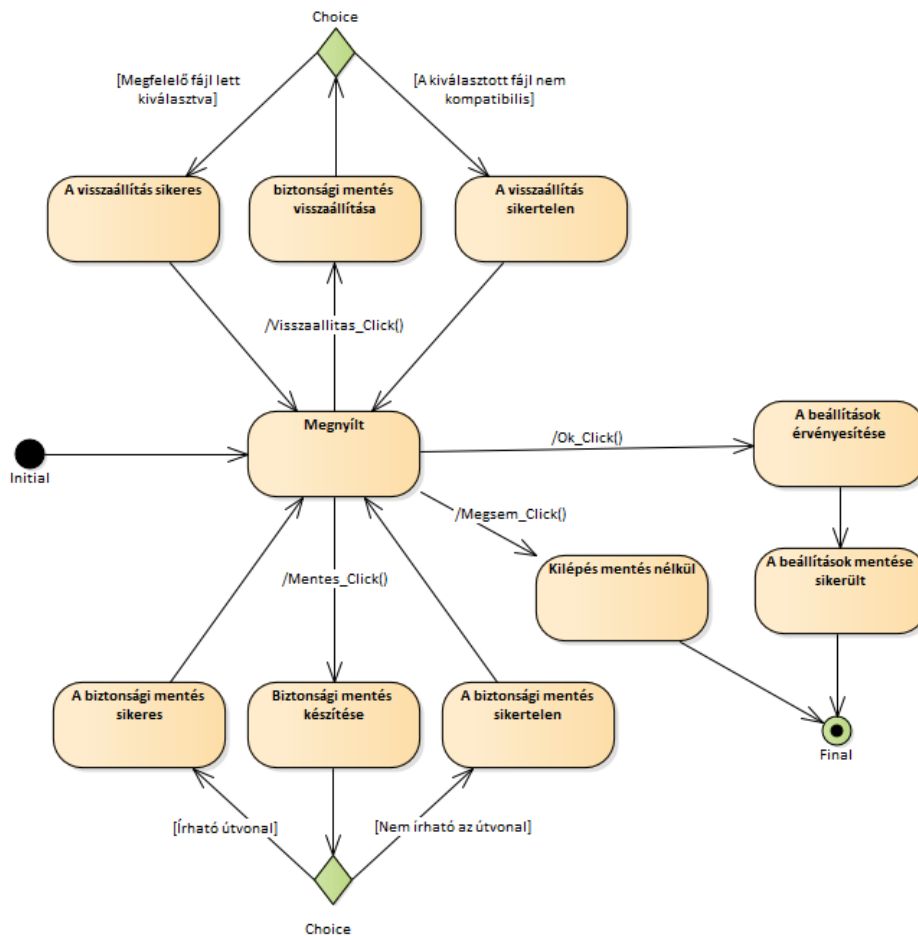


A szerviz form biztosítja az adatbázis mentés és visszaállítás lehetőségét. Mindkét eset hasonlít a bizonylat formon elvégezhető bizonylat rögzítés és kimentés funkcióra, annyi különbséggel, hogy nem egy objektumhoz tartozó adattal dolgozunk, hanem az egész adatbázist érintő műveletről van szó.

Amennyiben a felhasználó a kívánt beállítási lehetőségeket elvégzi, és az OK gombbal hagyja el a formot, a beállítások bejegyződnek a megfelelő helyekre és bezárul az ablak. A bejegyzéseket egy másik objektum végzi el, így az nem tartozik a feltüntetendő state-ek közé. A mégsem gombbal mentés nélkül lehet bezárni a formot.

Fontos megjegyezni, hogy az adatbázis mentés és visszaállítás műveletek elvégződnek függetlenül attól, milyen módon hagyja el az ablakot a felhasználó, hiszen azok nem bezáródáskor érvényesülnek. Ez egyértelműen látszik a diagramon is.

30. ábra
State Machine Diagram – Szerviz.form



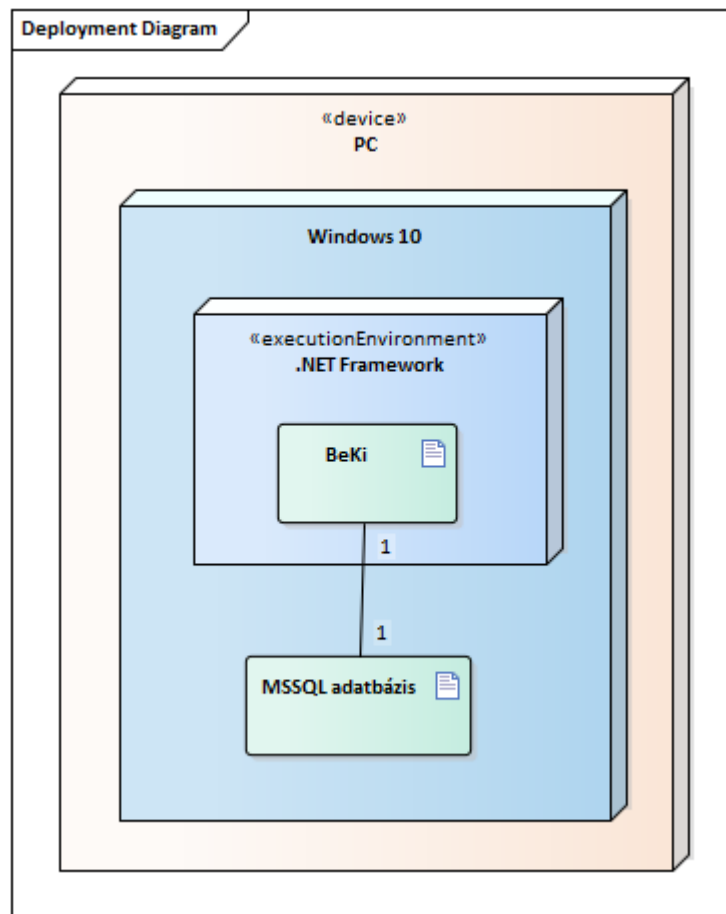
3.9. Deployment Diagram

A szoftver lokális adatbázissal kommunikál, 1-1 kapcsolatban vannak egymással, tehát a programból egy példány fut és csak az fér hozzá az adatbázishoz, illetve az adatbázis nem kommunikál más külső szoftverrel. (31. ábra)

A „BeKi” szoftver működéséhez elengedhetetlen a .NET keretrendszer, ami a Windows része. Az összes említett szoftver a munkaállomáson fut.

A szoftver és hardverigények a felhasználói dokumentációban megtalálhatóak.

31. ábra
Deployment Diagram



3.10. Tesztesetek

A tesztesetekkel a felhasználótól, vagy a program funkcionalitásából fakadó kivételeket, hibás működéseket próbáljuk kiszűrni a feltételezhetően bekövetkező eseteket előidézve. Az alábbi táblázatban részegységenként felsorolva látható néhány szituáció.

tesztelt részegység	teszteset	teszt eredménye
Pénzmozgás form	A megnevezés nem lehet üres	A form nem engedte a hozzáadást a megnevezés kitöltése nélkül.
Pénzmozgás form	Nincs kiválasztott kategória	A form nem engedte a hozzáadást a kategória nélkül.
Pénzmozgás form	Ha felvitelkor minden adat helyesen van megadva, elvégzi-e a program a hozzáadást úgy a bevitt adatok a megfelelő táblázatba, helyesen kerülnek visszaolvasásra?	A szoftver hibátlanul viszi fel és helyesen jeleníti meg az adatokat.
BeKi (fő form)	Törléshez nincs kijelölt elem	A form figyelmeztetett, hogy nincs kijelölt tétel.
BeKi (fő form)	A módosításhoz nincs kijelölt elem	A form figyelmeztetett, hogy nincs kijelölt tétel.
BeKi (fő form)	Nincs kijelölt pénzmozgás tétel a bizonylat felvitelhez.	A form figyelmeztetett, hogy nincs kijelölt tétel.
BeKi (fő form)	A form bezárásakor a program futása nem ér véget, hanem tovább fut a háttérben?	Igen, a szoftver futása megfelelően folytatódik.
Kategóriák form	Az újonnan felvitelre kerülő kategóriához nincs típus kiválasztva.	A form figyelmeztetett, hogy nincs kijelölt típus.
Kategóriák form	Már létező kategória megnevezés tiltott.	A form figyelmeztetett, hogy már létezik ilyen megnevezésű kategória az adott típushoz.
Szerviz form	Biztonsági mentés visszaállítása előtt jelezze a form, hogy ezzel minden jelenlegi adat lecserélődik a mentésben tároltakra.	A szoftver helyesen járt el, figyelmeztette a felhasználót.
Szerviz form	Hozzáadás után a kívánt beállítások érvénybe lépnek-e?	Igen, a beállítások megfelelően bejegyzésre és alkalmazásra kerülnek.

4. ÖSSZEGZÉS

4.1. A megvalósult projekt

A specifikációban összefoglalt célkitűzéseimet szemlélve az elkészült szoftver megfelel azoknak. Alapvető elvárások voltak a használhatóság és a funkcionalitás. A program képes a pénzügyi tranzakciók mindkét irányát könyvelni és tervezni, azokhoz bizonylatokat rögzíteni. A felhasználónak lehetősége van több módszerrel is terveznie a pénzmozgásait, lehetősége van áttekinteni egy éves időintervallumot. Biztonságban tudhatja adatait az adatbázis mentés lehetőségével és nem kell folyton fejben tartania a havi esedékes kiadásokat, amennyiben engedélyezi a program háttérben történő futását, és az értesítéseit.

4.2. Továbbfejlesztési lehetőségek

A világ jelenlegi számítástechnikai fejlettsége egy ilyen típusú szoftver tovább fejleszthetőségének aligha szabhat határt, inkább csak a felhasználói igények és a programozó fantáziája korlátozhatja merre folytatódik az alkalmazás fejlődése. Mivel minden feltétel adott, így több irányba is indulhatunk. Ami elsők között merül fel, az a platformfüggetlenség, hogy a felhasználó ne legyen rákényszerítve egy eszközre, hanem azon használhassa a szoftvert, amin szeretné. Az utóbbi években a mobileszközök elterjedése háttérbe szorította a PC-k használatát a magánemberek és üzleti felhasználók között is, hiszen egy notebook sem lehet mindig kéznél és talán nehezebb is megszoknia a használatát egy gyakorlatlan felhasználónak, de egy szoftver felhasználói felülete és annak funkcionalitásai sokkal áttekinthetőbbek egy személyi számítógépen, mint egy tableten vagy egy telefonon. Mégis el kell fogadni a reális igényt, hogy mobileszközön is szükség van ilyen szoftverre, gondoljuk a költségek azonnali felvitelére egy vásárlás alkalmával. Ezért az egyik irány a platformfüggetlenség, az adatok szinkronizálásával felhő szolgáltatáson keresztül, hogy minden eszközön a lokális adatbázis azonos adatokat tartalmazzon.

A másik lehetőség a jelenlegi állapot továbbfejlesztése újabb funkcionalitások hozzáadásával. Ilyenek például a kiadási oldalon időszakos rendszerességgel rögzíthető fogyasztásmérő óraállások, vásárlási bizonylatok kategóriákra bontása rögzítés közben, mint tisztítószer, élelmiszer, élvezeti cikk stb., továbbá minden bevétel és kiadás, illetve fogyasztásmérő óraállás grafikus kimutatásának megjelenítése, más időszakokkal való

összehasonlítása. Külön modullal a családi gépkocsikhoz köthető költségek, óraállások, tankolási bizonylatok rögzítése, mely utóbbi egy gépkocsieladásnál is árfelhajtó tényező lehet, mert igazolható, hogy az autó milyen üzemanyagot kapott és nagyjából kikövetkeztethető – a tankolások gyakorisága által –, az autó kilométer számláló állásának valóságossága. Ezt egy eladás alkalmával több oldalas pdf-be exportálva át lehetne adni a vásárlónak. Külön rögzíthető lenne műszaki cikkek jótállásának vagy garanciájának várható lejáratát, hogy a szoftver értesítse a felhasználót előtte egy hónappal, így az átnézhető, hogy az esetleges hibák időben jelezve legyenek a forgalmazónak, vagy akár az eszköz lecserélését is megfontolhatja az alkalmazás, ugyanis segít megbecsülni „meddig ér még a takaró”, milyen költségeket vállalhat a család.

5. IRODALOMJEGYZÉK

Eidenpenz, J. (2013). *privatbankar.hu*. Letöltés dátuma: 2020. február 20., forrás: privatbankar.hu:
<https://privatbankar.hu/penzugyek/111-kinos-tenyezo-a-devizahitelekkel-kapcsolatban-260735>

Microsoft. (dátum nélk.). *System.ComponentModel Namespace*. Letöltés dátuma: 2020. február 19,
forrás: <https://docs.microsoft.com>: <https://docs.microsoft.com/en-us/dotnet/api/system.componentmodel?view=netframework-4.8>

Wikipédia. (2019. október 8). *A 2008–2009-es gazdasági világválság kronológiája*. Letöltés dátuma:
2020. február 20, forrás: Wikipédia: https://hu.wikipedia.org/wiki/A_2008%E2%80%932009-es_gazdas%C3%A1gi_vil%C3%A1gv%C3%A1ls%C3%A1g_kronol%C3%B3gi%C3%A1ja

6. FELHASZNÁLT SZOFTVEREK ÉS EGYÉB ANYAGOK

Visual Studio 2019 Community 16.4.2

Microsoft SQL Server Express 2016 (13.1.4001.0)

Enterprise Architech Academic Edition 13.5

Dia 0.97.2

A szoftver ikonjának ingyenesen felhasználható alapja az alábbi helyről lett letöltve:

<https://icons8.com/icons/set/piggy-bank>

A kép grafikus módosításait én végeztem el.

7. MELLÉKLETEK

CD lemez, melynek tartalma a szoftver teljes forráskódja, és a szakdolgozat pdf formátumban.

Nyilatkozat

8. ÁBRAJEGYZÉK

1. ábra A BeKi főablaka	4
2. ábra Mozgás a hónapok között	4
3. ábra Pénzmozgás ablak	6
4. ábra Kategóriák ablak.....	7
5. ábra „Nincs mentett bizonylat” és „van mentett bizonylat” a kijelölt pénzmozgáshoz	7
6. ábra Kimutatás ablak.....	8
7. ábra Hónap generálás gombra kattintva megjelenik a megerősítést kérő ablak	9
8. ábra Hónapépítő ablak	10
9. ábra Szerviz ablak.....	11
10. ábra Biztonsági mentés készítése	11
11. ábra Biztonsági mentés visszaállítása	12
12. ábra Hónapgenerálási hibaüzenet	12
13. ábra Túl rövid, vagy túl hosszú megnevezés	13
14. ábra Létező kategória megnevezés hibaüzenet	13
15. ábra Olyan kategória törlésének kísérlete, melyhez tartozik pénzmozgás.....	14
16. ábra Pénzmozgás törlési kísérlet hibaüzenete.....	14
17. ábra A bizonylat kimentéséhez választott útvonal nem írható.....	15
18. ábra Az adatbázis, adatok tárolását végző tábláinak ER diagramja	18
19. ábra A beállítások tárolása az adatbázisban	18
20. ábra Package Diagram.....	19
21. ábra Use Case Diagram	21
22. ábra Stereotype Class Diagram	23
23. ábra Class Diagram I.	24
24. ábra Class Diagram II.	25
25. ábra Sequence diagram – Pénzmozgás felvitele	26
26. ábra Sequence diagram – Éves kimutatás.....	27
27. ábra Sequence diagram – Szerviz ablak	28
28. ábra State Machine Diagram – Pénzmozgás felvitele	30
29. ábra State Machine Diagram – Bizonylattal kapcsolatos műveletek	31
30. ábra State Machine Diagram – Szerviz form	32
31. ábra Deployment Diagram	33