

Webes alkalmazásfejlesztés I. beadandó

Bittner Barnabás

April 17, 2017

Contents

1	Feladat	3
2	Felhasználói dokumentáció	4
2.1	Felhasználói esetek	4
3	Fejlesztői dokumentáció	4
3.1	Core	5
3.2	UI	5
3.3	Adatbázis	6
4	Tesztelés	6

1 Feladat

Készítsük el egy bank ügyfelek kezelését, és az ügyfelekkel kapcsolatos tevékenységek adminisztrálását elősegítő rendszert. A webes felületen keresztül az ügyfelek érik el a bankolási funkciókat.

1. A főoldalon lehetőségünk van bejelentkezésre. Bejelentkezéskor meg kell adnunk a felhasználónevünket, jelszavunkat, számlaszámunkat (ha több van, akkor a legelsőt), valamint ellenőrző PIN kódunkat. Ezen felül a felhasználó választhat biztonságos üzemmódot is, ekkor minden művelet (számlatörténet lekérdezés, illetve átutalás) előtt a weblap ismét bekéri a felhasználó jelszavát. A bejelentkezést követően bármikor kijelentkezhet az ügyfél.
2. Sikeres bejelentkezés esetén lehetősége nyílik megtekinteni a számlái egyenlegeit (egy ügyfélhez legalább egy, de tetszőlegesen sok számla tartozhat), valamint azok történetét (átutalások, betétek, kivétek listája) egy hónapra visszamenőleg.
3. Lehetősége van új átutalást megadni (ha a számla nincs zárolva), ekkor ki kell tölteni az összeget, célszámla tulajdonosát, valamint számlaszámát, majd elküldhetjük az utalást, ekkor az összeg azonnal levonódik az egyenlegből (az átutalandó összeg nem lehet nagyobb a rendelkezésre álló egyenlegnél). Amennyiben a célszámla is a banknál van, akkor ott is meg kell jelennie az összegnek, mint befolyó utalás. A grafikus felületet a banki alkalmazottak használják a tranzakciók (betétek, kivételek és átutalások) kezelésére.
4. Az alkalmazottnak be kell jelentkeznie az alkalmazásba, ezt követően kiválaszthatja az ügyfelet, illetve annak bankszámláját.
5. Adott bankszámlára végezhet betétet, kivétet, illetve átutalást. Az első két esetben csak az összeget kell megadnia, míg a harmadik esetben (a webes felülethez hasonlóan) a célszámla adatait (számlaszám, tulajdonos) is. A tranzakció elküldésével az összeg azonnal levonódik/hozzáadódik az egyenleghez (átutaláskor, illetve kivételkor az összeg nem lehet nagyobb a rendelkezésre álló egyenlegnél). Amennyiben a célszámla is a banknál van, akkor ott is meg kell jelennie az összegnek, mint befolyó utalás.
6. Egy számla zárolható is, ekkor nem lehet semmilyen tevékenységet kezdeményezni a számlán (sem a grafikus, sem a webes felületen), amíg a zárolást fel nem oldja valamely alkalmazott. A zároláshoz a program kérjen megerősítést.

Az adatbázis az alábbi adatokat tárolja (ezek még nem feltétlenül a fizikai adattáblák):

1. ügyfelek (teljes név, felhasználónév, jelszó, PIN kód);

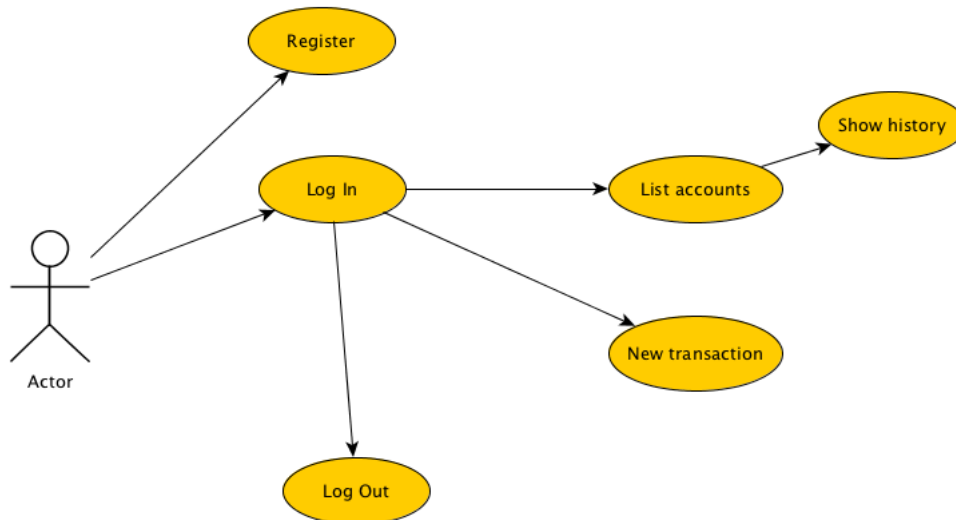
2. alkalmazottak (teljes név, felhasználónév, jelszó);
3. számlák (felhasználó, számlaszám, létrehozás dátuma);
4. tranzakciók (típus, forrás számlaszám, cél számlaszám, cél tulajdonos neve, dátum, összeg).

2 Felhasználói dokumentáció

A program használata rendkívül egyszerű csak egy böngészőprogramban meg kell nyitni a SmartBank weboldalt. Ha a felhasználónak még nincs fiókja regisztrálhat egyet. Regisztráció után azonnal be lehet jelentkezni, ahol a felhasználó elérheti az összes saját bankszámláját, rajta az összesített összeggel. Minden számlán le lehet kérdezni a számlatörténetet, mely után a program listázza az elmúlt havi tranzakciók listáját.

2.1 Felhasználói esetek

Figure 1: Use case diagram



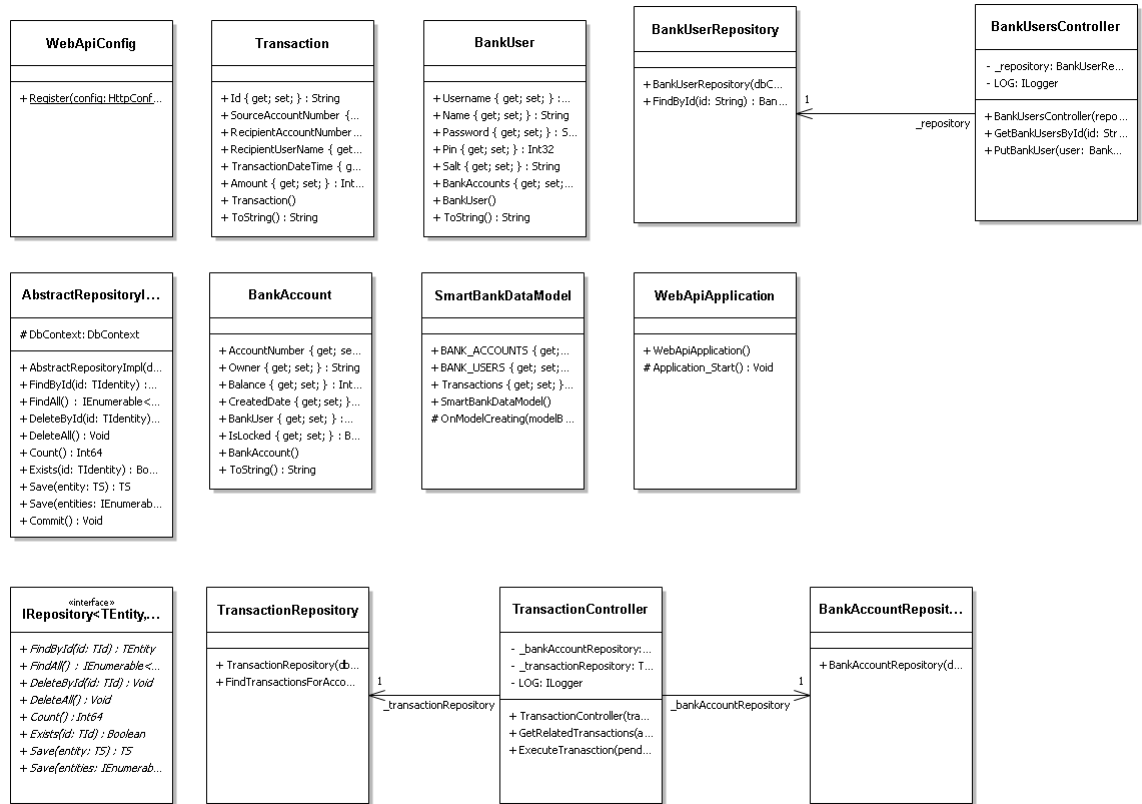
3 Fejlesztői dokumentáció

A program alapvetően három részből áll, melyek: Core, UI, Adatbázis. Ezeket a következő szekciókban részletezem.

3.1 Core

A program funkcionalitásának magja, itt találhatóak az üzleti logika részei. A megvalósítás C# WebAPI segítségével történik, a felhasználói felület HTTP hívásokkal kommunikál az üzleti logikával. A Core alapvetően a Domain Driven Design architektúrában készült, hexagonális architektúrát követve.

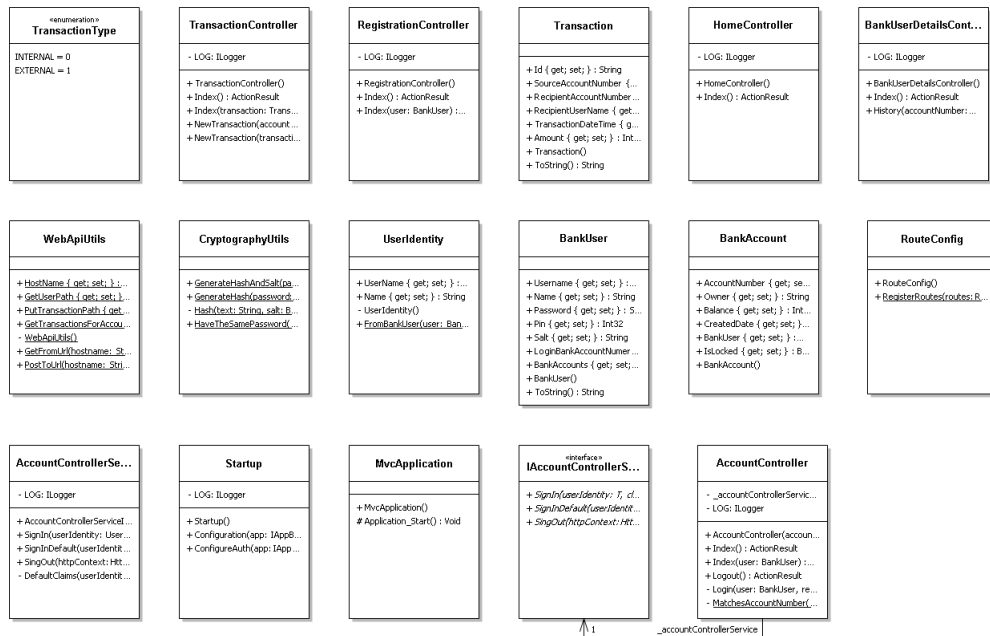
Figure 2: Core Class Diagram



3.2 UI

A felhasználói felület C# ASP MVC technológiával készült el, bármilyen modern böngészőben megjeleníthető. A felület HTTP hívásokkal kommunikál a core-al, amiktől csak adatokat kap, amiket megfelelő módon megjelenít.

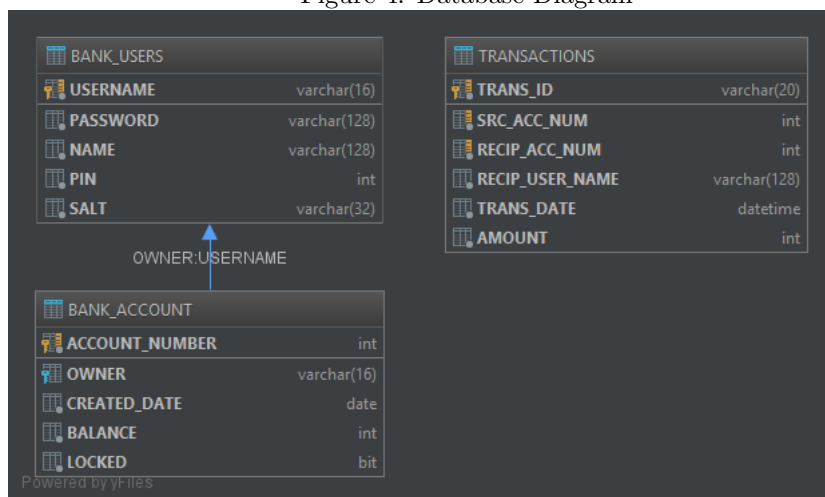
Figure 3: UI Class Diagram



3.3 Adatbázis

A program a működéshez használt adatokat fizikai adatbázisban tárolja, melyek tartalmazzák a regisztrált felhasználókat, azok néhány adatát. A felhasználók jelszavát enkriptálva tároljuk, esetleges adatszivárgás esetén ne lehessen (meglehetősen sok időt vegyen igénybe) a jelszavakat megszerezni. A tranzakciókat szintén az adatbázisban tároljuk, a bankszámlákkal együtt melyekhez eltároljuk a rajtuk lévő összeget. Az összeget csakis tranzakciókon keresztül lehet módosítani.

Figure 4: Database Diagram



4 Tesztelés

A program helyességét Unit testekkel ellenőriztük, melyek nagyban lefedik a program által szolgáltatott lehetőségeket.