

Dokumentáció a Programozási technológia 2. tantárgy beadandó feladatához

Téma: Internet kávézó

Készítette: Hete Csaba (TP75FW)

Email: csaba.hete (at) gmail.com

Gyakorlatvezet : Nagy Krisztián

Tartalomjegyzék

Bevezetés	2
Feladat leírása	3
Követelmény elemzése	4
I. Funkcionális követelmények.....	4
1.) számítógépekkel kapcsolatos funkciók:	5
2.) ügyfelekkel kapcsolatos funkciók:	8
3.) adminisztratív funkciók.....	13
II. Nem funkcionális követelmények	15
1.) Termék követelmények	15
2.) Szervezeti követelmények.....	15
3.) Küls követelmények.....	16
Fejlesztési dokumentáció	17
I. Adatbázis terv	17
1.) Adatbázis sémák	17
2.) Egyed-kapcsolat diagram	19
II. Statikus terv	20
1.) osztálydiagram	20
III. Felhasználói-felület modell	22

Bevezetés

Ez a dokumentáció a Programozási technológia 2. tantárgy beadandó feladat részeként készült el. Célja, hogy minél jobban megismerkedjünk a szoftverfejlesztés dokumentálási részfeladataival, elsajátítsuk a szoftverek rendszerezett, felügyelt, minősített fejlesztésének módszereit, lépéseit.

Jelen projekt feladata egy internet kávézó nyilvántartását kezelő program elkészítése. Elsődleges célja, hogy a tanulmányaim során szerzett tudásom és tapasztalataim felhasználásával egy olyan szoftvert fejlesszek, ami megfelel a tárgy, illetve az oktatók elvárásainak, használata egyszerű, a kódja megbízható, funkciói átláthatóak. Másodlagos cél, hogy a követelményeket meghaladva egy olyan informatikai terméket állítsak elő, ami a való életben is megállná a helyét egy kisebb internet kávézó mindennapi teendőinek ellátásában.

A program célközönsége a kisebb internet kávézók üzemeltetői, ügyintézői, akik jelen szoftver használatával egy kényelmes, és megbízható rendszerben végezhetik mindennapi ügyfélszolgálati teendőiket.

Feladat leírása

Készítsünk programot, amely egy internet kávézó nyilvántartását tudja kezelni, az alábbi funkciókkal:

- A programban megtekinthetők a számítógépek adatai (azonosító, hardver leírás, operációs rendszer).
- Felvehetünk új ügyfeleket, akiknek megadjuk a nevét, személyi igazolvány számát, címét, valamint felhasználónevét és jelszavát. Az adatokat a későbbiekben módosíthatjuk is (kivéve a felhasználónevet).
- Az ügyfeleket is listázhatjuk a programban (azonosító, cím, személyi igazolványszám). Az ügyfelet kiválasztásával beléptethetjük, valamint kiléptethetjük a kávézóból. Beléptetéskor kiválasztjuk a számítógépet, amelyet az ügyfél elfoglal (természetesen csak szabad gépet lehet megadni és egy ügyfél csak egy számítógépet használhat egyszerre). Az időpontok automatikusan rögzítésre kerülnek, és ezek alapján számolódik az ügyfél számlája, amelyet az ügyfeleknél tekinthetünk meg.
- A számla a következő módon áll össze:
 - Ø az ügyfél óránként bizonyos összeg alapdíjat fizet, majd a végösszegből levonásra kerül bizonyos mennyiség kedvezmény.
 - Ø A kedvezmény mértékének meghatározása:
 - Ø az ügyfelek minden használati óra után 2 pontot kapnak 16:00-óra előtt és 21:00 után
 - Ø 16:00 és 21:00 közötti intervallumban óránként 1 pontot, a nem egész órák kezdő intervallum esetén a kevesebb pontot kell elszámolni (pl.: 14:50-16:50-> 3 pont: 14:50-15:50 2 pont + 15:50-16:50 -> 1 pont)
 - Ø az ügyfél minden 150 pont után 1% állandó kedvezményt kap, de legfeljebb 10%-ot.
- Szintén az ügyfél kiválasztásával végezhetünk befizetést. A befizetett összegeket le kell számolnunk a számlából. Olyan ügyfél, akinek hiányzó befizetései vannak nem léptethető ki.

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

- ügyfelek (azonosító, név, személyi igazolvány szám, felhasználónév, jelszó, hűségpontok, regisztráció időpontja);
- címek (azonosító, ország, település, irányítószám, közterület neve, közterület jellege, házszám és további információk)
- számítógépek (azonosító, név, hardver leírás (CPU, alaplap, memória, VGA, háttértár információk), operációs rendszer, számítógép állapota (foglalt-e));
- munkamenet (azonosító, ügyfél azonosító, számítógép azonosító, bejelentkezés időpontja, kijelentkezés időpontja);
- befizetések (azonosító, ügyfél azonosító, munkamenet azonosító, nettó egységár, mennyiség, ÁFA kulcs, kedvezmény, bruttó összeg, pénznem, befizetés időpontja);

Követelmény elemzése

I. Funkcionális követelmények

Az alábbiakban a könnyebb átláthatóságért ez az egy diagram két részre van osztva, a Use Case-ek leírása majd ott lesz látható.



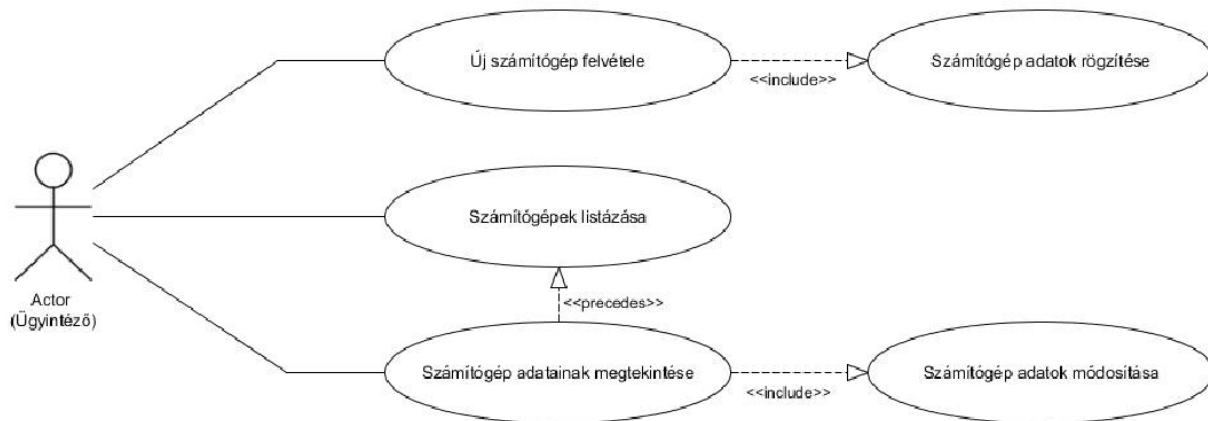
1. ábra: A teljes Use Casediagram

Aktorok: a feladat és a projekt leírásából következik, hogy csak egyetlen aktorunk van, a kávézó munkatársa (nevezzük a továbbiakban ügyintézőnek), az ügyfelek nem rendelkeznek hozzáféréssel a programhoz. Nem célunk külön autorizáció kialakítása az ügyintéző

azonosítására, feltételezzük, hogy a rendszer hozzáférhet sége a vendégek számára fizikailag korlátozva van.

A Use Case-eket három csoportra bonthatjuk:

1.) számítógépekkel kapcsolatos funkciók:



2. ábra: Számítógépekkel kapcsolatos funkciók

A továbbiakban a számítógép adatbázisbeli azonosítója „azonosító”, ügyintézési azonosítója „név” címkével szerepel.

Új számítógép felvétele	
Actor	Az ügyintéző .
El feltétel	–
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	Az új számítógép adatai rögzítésre kerülnek. A felhasználó a művelet sikerességéről visszajelzést kap.
Kiváltja	A felhasználó a menüben kiválasztotta a megfelelő menüpontot.
Események	<ol style="list-style-type: none"> 1. Az felhasználó megadja az új számítógép adatait: név, hardver leírás (CPU, alaplap, memória, VGA, háttértár információk), operációs rendszer. 2. A felhasználó az számítógép rögzítése gombra kattint. 3. A rendszer ellenőrzi az azonosító meglétét. 4. A rendszer ellenőrzi az azonosító egyediségét. 5. A rendszer az adatbázisba menti a számítógép adatait (azonosító, név, hardver leírás (CPU, alaplap, memória, VGA, háttértár információk), operációs rendszer, rögzítés időpontja). 6. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 3. a) A számítógép nevének hiánya hibát vált ki. b) A rendszer jelzi a hibát „Hibás vagy helytelen adat került megadásra” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul.

	<p>4.</p> <p>a) A megadott számítógépnév már szerepel a rendszer adatbázisában.</p> <p>b) A rendszer jelzi a hibát „A megadott számítógépnév már szerepel az adatbázisban!” üzenettel.</p> <p>c) Az adott Use Case sikertelen eredménnyel zárul.</p> <p>5.</p> <p>a) Az adatok rögzítése sikertelen.</p> <p>b) A rendszer jelzi a hibát „A számítógép rögzítése sikertelen!” üzenettel.</p> <p>c) Az adott Use Case sikertelen eredménnyel zárul.</p>
--	---

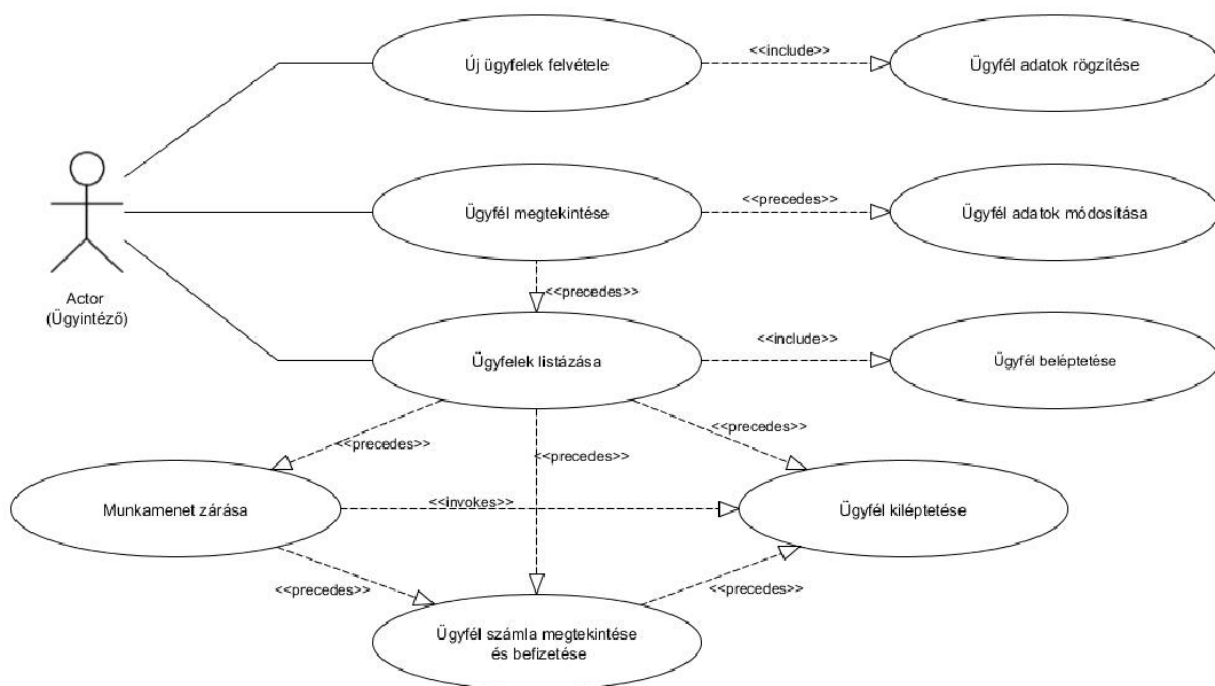
Számítógépek listázása	
Actor	Az ügyintéző .
El feltétel	–
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó a programban egy listában megtekintheti az adatbázisban szereplő számítógépek neveit, tulajdonságait, illetve, hogy foglalt-e. Ezen nevek segítségével válik lehetségessé az egyes gépek tulajdonságainak megtekintése.
Kiváltja	A felhasználó a menüben kiválasztotta a megfelelő menüpontot.
Események	<p>1. A rendszer lekéri az adatbázisban szereplő számítógépek adatait.</p> <p>2. Az adott Use Case sikeres eredménnyel zárul.</p>
Kiegészítés	<p>1.</p> <p>a) Az adatok lekérése sikertelen.</p> <p>b) A rendszer jelzi a hibát „A számítógépek listázása sikertelen!” üzenettel.</p> <p>c) Az adott Use Case sikertelen eredménnyel zárul.</p>

Számítógép adatainak megtekintése	
Actor	Az ügyintéző .
El feltétel	A számítógépek listázása sikeres volt.
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó a programban egy ablakban megtekintheti az általa kiválasztott számítógép tulajdonságait (azonosító, név, hardver leírás, operációs rendszer).
Kiváltja	A felhasználó a számítógépek listájából kiválaszt egyet.
Események	<p>1. A rendszer megjeleníti a számítógép adatait.</p>

	2. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	–

Számítógép adatainak szerkesztése	
Actor	Az ügyintéző .
El feltétel	A számítógép megtekintése sikeres volt.
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó módosítja az általa kiválasztott számítógép adatait: név, hardver leírás (CPU, alaplap, memória, VGA, háttértár információk), operációs rendszer.
Kiváltja	A felhasználó az adatok módosítása gombra kattint.
Események	<ol style="list-style-type: none"> 1. Az felhasználó megadja a számítógép adatait: név, hardver leírás (CPU, alaplap, memória, VGA, háttértár információk), operációs rendszer. 2. A felhasználó az számítógép rögzítése gombra kattint. 3. A rendszer ellenőrzi az azonosító meglétét. 4. A rendszer ellenőrzi az azonosító egyediségét. 5. A rendszer az adatbázisba menti a számítógép adatait (azonosító, név, hardver leírás (CPU, alaplap, memória, VGA, háttértár információk), operációs rendszer, rögzítés időpontja). 6. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 3. <ol style="list-style-type: none"> a) A számítógép nevének hiánya hibát vált ki. b) A rendszer jelzi a hibát „Hibás vagy helytelen adat került megadásra” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul. 4. <ol style="list-style-type: none"> a) A megadott számítógépnév már szerepel a rendszer adatbázisában. b) A rendszer jelzi a hibát „A megadott számítógépnév már szerepel az adatbázisban!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul. 5. <ol style="list-style-type: none"> a) Az adatok rögzítése sikertelen. b) A rendszer jelzi a hibát „A számítógép rögzítése sikertelen!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul.

2.) ügyfelekkel kapcsolatos funkciók:



3. ábra: Ügyfelekkel kapcsolatos funkciók

Új ügyfelek felvétele	
Actor	Az ügyintéző.
El feltétel	–
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	Az új ügyfél adatai rögzítésre kerülnek. A felhasználó a művelet sikerességéről visszajelzést kap.
Kiváltja	A felhasználó a menüben kiválasztotta a megfelelő menüpontot.
Események	<ol style="list-style-type: none"> 1. Az felhasználó megadja az új ügyfél adatait: Név, Személyi igazolvány szám, Cím, Felhasználónév, Jelszó. 2. A felhasználó az ügyfél rögzítése gombra kattint. 3. A rendszer ellenőrzi az adatok meglétét. 4. A rendszer ellenőrzi a felhasználónév foglalt-e. 5. A rendszer ellenőrzi, hogy a megadott két jelszó egyezik-e. 6. A rendszer az adatbázisba menti a felhasználó adatait (Név, Személyi igazolvány szám, Cím, Felhasználónév, Jelszó, Regisztráció időpontja). 7. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 3. a) Bármely ügyfél adat hiánya (Név, Személyi igazolvány szám, Cím, Felhasználónév, Jelszó) hibát vált ki. b) A rendszer jelzi a hibát „Hibás vagy helytelen adat került megadásra” üzenettel.

	<p>c) Az adott Use Case sikertelen eredménnyel zárul.</p> <p>4.</p> <p>a) Amennyiben a megadott felhasználónév már szerepel az adatbázisban, azt a rendszer hibával jelzi.</p> <p>b) A rendszer jelzi a hibát „A megadott felhasználónév már szerepel az adatbázisban!” üzenettel.</p> <p>c) Az adott Use Case sikertelen eredménnyel zárul.</p> <p>5.</p> <p>a) A megadott jelszavak nem egyeznek meg.</p> <p>b) A rendszer jelzi a hibát „A megadott jelszavak nem egyeznek meg” üzenettel.</p> <p>c) Az adott Use Case sikertelen eredménnyel zárul</p>
--	--

Ügyfelek listázása	
Actor	Az ügyintéző .
El feltétel	–
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó a programban egy listában megtekintheti az adatbázisban szereplő ügyfeleket (név, szem. ig. szám, felhasználónév, h. ségpontok, be van-e jelentkezve, rögzítés ideje, utolsó módosítás ideje). Ezen adatok segítségével válik lehetségessé az egyes ügyfelek adatlapjainak megtekintése.
Kiváltja	A felhasználó a menüben kiválasztotta a megfelelő menüpontot.
Események	<ol style="list-style-type: none"> 1. A rendszer lekéri az adatbázisban szereplő ügyfelek adatait. 2. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 1. <ul style="list-style-type: none"> a) Az adatok lekérése sikertelen. b) A rendszer jelzi a hibát „Az ügyfelek listázása sikertelen!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul.

Ügyfél adatainak megtekintése	
Actor	Az ügyintéző .
El feltétel	Az ügyfelek listázása sikeres volt.
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó a programban egy ablakban megtekintheti az általa kiválasztott ügyfél adatait, az ablakban szereplő adatokat módosíthatja.
Kiváltja	A felhasználó az ügyfelek listájában a kiválasztott ügyfélre az egér

	jobb gombjával kattintva a megjelen menüb l kiválasztja a megfelel funkciót.
Események	<ol style="list-style-type: none"> 1. A rendszer az ügyfél azonosítója alapján lekéri annak adatait. 2. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a) Az adatok lekérése sikertelen. b) A rendszer jelzi a hibát „Az ügyfél adatok lekérése sikertelen!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul.

Ügyfél adatainak módosítása	
Actor	Az ügyintéző .
El feltétel	Az ügyfél megtekintése sikeres volt.
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibáüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó módosítja az általa kiválasztott ügyfél adatait (Név, Személyi igazolvány szám, Cím, Felhasználónév, Jelszó).
Kiváltja	A felhasználó a mentés gombra kattint.
Események	<ol style="list-style-type: none"> 1. Az ügyintéző megadja az ügyfél adatait: Név, Személyi igazolvány szám, Cím, Felhasználónév. 2. Az ügyintéző az adatok mentése gombra kattint. 3. A rendszer ellenőrzi az adatok meglétét. 4. A rendszer ellenőrzi a felhasználónév foglalt-e. 5. A rendszer az adatbázisba menti a felhasználó adatait (Név, Személyi igazolvány szám, Cím, Felhasználónév, Jelszó, Szerkesztés időpontja). 6. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 3. <ol style="list-style-type: none"> a) Bármely ügyfél adat hiánya (Név, Személyi igazolvány szám, Cím, Felhasználónév, Jelszó) hibát vált ki. b) A rendszer jelzi a hibát. c) Az adott Use Case sikertelen eredménnyel zárul. 4. <ol style="list-style-type: none"> a) Amennyiben a megadott felhasználónév már szerepel az adatbázisban, azt a rendszer hibával jelzi. b) A rendszer jelzi a hibát „A megadott felhasználónév már szerepel az adatbázisban!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul.

	<p>5. A megadott jelszavak nem egyeznek meg.</p> <p>a) A rendszer jelzi a hibát „A megadott jelszavak nem egyeznek meg” üzenettel.</p> <p>d) Az adott Use Case sikertelen eredménnyel zárul</p>
--	---

Ügyfél beléptetése	
Actor	Az ügyintéző .
El feltétel	Az ügyfelek listázása sikeres volt.
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó belépteti az ügyfelet a rendszerbe. A belépés időpontja rögzítésre kerül, a kiválasztott számítógép állapota foglaltra változik.
Kiváltja	A felhasználó az ügyfelek listájában a kiválasztott ügyfélre az egér jobb gombjával kattintva a megjelenő menüben kiválasztja a megfelelő funkciót.
Események	<ol style="list-style-type: none"> 1. A rendszer ellenőrzi, az ügyfél be van-e lépve. 2. A rendszer ellenőrzi, van-e szabad számítógép. 3. Amennyiben van szabad számítógép, a rendszer rögzíti a belépés időpontját, a számítógép állapotát foglaltra állítja, belépteti az ügyfelet, rögzíti melyik géphez került a ügyfél. 4. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a) Amennyiben az ügyfél be van lépve, az hibát vált ki. b) A rendszer jelzi a hibát „A felhasználó már be van lépve!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul. 2. <ol style="list-style-type: none"> a) Amennyiben nincs szabad számítógép, az hibát vált ki. b) A rendszer jelzi a hibát „Nincs szabad számítógép!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul.

Munkamenet zárása	
Actor	Az ügyintéző .
El feltétel	Az ügyfelek listázása sikeres volt.
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó megtekintheti az ügyfél aktuális munkamenetének adatait, a munkamenet zárása gomb segítségével zárhatja a munkamenetet.

Kiváltja	A felhasználó az ügyfelek listájában a kiválasztott ügyfélre az egér jobb gombjával kattintva a megjelen menüb l kiválasztja az ügyfél kiléptetése funkciót.
Események	<ol style="list-style-type: none"> 1. A rendszer ellen rzi, az ügyfél be van-e lépve. 2. A rendszer megjeleníti az adott ügyfél aktuális munkamenetének adatait. 3. A rendszer a munkamenet számítógépének állapotát szabadra állítja. 4. A rendszer zárja a munkamenetet. 5. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a) Amennyiben az ügyfél nincs belépve, az hibát vált ki. b) A rendszer jelzi a hibát „A felhasználó nincs belépve!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul.

Ügyfél számla megtekintése és befizetése	
Actor	Az ügyintéző .
El feltétel	A munkamenet zárása sikeres volt.
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó megtekintheti az ügyfél aktuális számlaegyenlegét.
Kiváltja	A felhasználó az ügyfelek listájában a kiválasztott ügyfélre az egér jobb gombjával kattintva a megjelen menüb l kiválasztja az ügyfél kiléptetése funkciót, vagy a befizetések listájában az egér jobb gombjával kattintva a megjelen menüb l kiválasztja a befizetés teljesítése funkciót
Események	<ol style="list-style-type: none"> 6. A rendszer ellen rzi, az ügyfél be van-e lépve. 7. A rendszer megjeleníti az adott ügyfél egyenlegét. 8. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> d) Amennyiben az ügyfél nincs belépve, az hibát vált ki. e) A rendszer jelzi a hibát „A felhasználó nincs belépve!” üzenettel. f) Az adott Use Case sikertelen eredménnyel zárul.

Ügyfél számla befizetése	
Actor	Az ügyintéző .
El feltétel	Az ügyfél számla megtekintése sikeres volt.
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.

Sikeres eredmény	A felhasználó rögzíti az ügyfél számlájának befizetését.
Kiváltja	A felhasználó a befizetés megerősítése gombra kattint.
Események	<ol style="list-style-type: none"> 1. A rendszer leszámolja a befizetett összeget az adott ügyfél egyenlegéből. 2. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	–

Ügyfél kiléptetése	
Actor	Az ügyintéző.
El feltétel	A munkamenet zárása és az ügyfél számla befizetése sikeres volt, az ügyfélnek nincsenek hiányzó befizetései.
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó kilépteti az ügyfelet.
Kiváltja	A felhasználó az ügyfél kiléptetése gombra kattint.
Események	<ol style="list-style-type: none"> 1. A rendszer ellenőrzi, az ügyfél be van-e lépve. 2. A rendszer ellenőrzi az adott ügyfél egyenlegét. 3. A rendszer kilépteti az ügyfelet. 4. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a) Amennyiben a felhasználó nincs belépve, az hibát vált ki. b) A rendszer jelzi a hibát „A felhasználó nincs belépve!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul.

3.) adminisztratív funkciók



4. ábra: Adminisztratív funkciók

Befizetések listázása	
Actor	Az ügyintéző.
El feltétel	–
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibaüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó a programban egy listában megtekintheti az adatbázisban szereplő befizetéseket. (Felhasználó azonosítója, neve, befizetés összege, időpontja)

Kiváltja	A felhasználó a menüben kiválasztotta a megfelelő menüpontot.
Események	<ol style="list-style-type: none"> 1. A rendszer lekéri az adatbázisban szereplő befizetéseket. 2. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a) Az adatok lekérése sikertelen. b) A rendszer jelzi a hibát „A befizetések listázása sikertelen!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul.

Befizetés teljesítése	
Actor	Az ügyintéző.
El feltétel	A befizetések listázása sikeres volt.
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibáüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó megtekintheti a kiválasztott befizetés adatait, a befizetést teljesítettnek jelölheti.
Kiváltja	A a befizetések listájában az egér jobb gombjával kattintva a megjelenő menüben kiválasztja a befizetés teljesítése funkciót
Események	<ol style="list-style-type: none"> 1. A rendszer megjeleníti az befizetés adatait. 2. A felhasználó megerősíti a befizetést 3. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a) Az adatok mentése sikertelen b) A rendszer jelzi a hibát. c) Az adott Use Case sikertelen eredménnyel zárul.

Rendszerbeállítások módosítása	
Actor	Az ügyintéző.
El feltétel	–
Sikertelen eredmény	A rögzített adatok nem változnak, adatmentés nem történik. A felhasználó hibáüzenetben megkapja a felmerülő probléma leírását.
Sikeres eredmény	A felhasználó módosítja a rendszer beállításait.
Kiváltja	A felhasználó a beállítások módosítása gombra kattint.
Események	<ol style="list-style-type: none"> 1. Az ügyintéző megadja az új beállításokat (pénznem, nettó ár/óra, árképzéshez használt ÁFA kulcs, az ügyfeleknek járó kedvezmény számításához használt osztó, ügyfeleknek járó maximális kedvezmény). 2. Az ügyintéző a beállítások mentése gombra kattint. 3. A rendszer ellenőrzi az adatok helyességét. 4. A rendszer a beállításokat tartalmazó fájlba menti a beállításokat.

	5. Az adott Use Case sikeres eredménnyel zárul.
Kiegészítés	3. <ul style="list-style-type: none"> a) Az ÁFA kulcs értéke nem megfelel ; az ügyfeleknek járó kedvezmény számításához használt osztó nullával egyenlő , ügyfeleknek járó maximális kedvezmény nem megfelel . b) A rendszer jelzi a hibát „Hibás beállítások!” üzenettel. c) Az adott Use Case sikertelen eredménnyel zárul.

II. Nem funkcionális követelmények

1.) Termék követelmények

A rendszer úgy készül, hogy elsajátítása minél könnyebb legyen. Felhasználói felülete jól áttekinthető és a különböző ablakok tartalma logikailag összetartozó információkat jelenítsen meg. Ezáltal a betanulási idő egy tapasztalatlan felhasználó esetében is lekorlátozódik egy órára. Alapértelmezett nyelve magyar, és az egyes gombok, menük feliratai is érthetően rávilágítanak arra, mi történik, ha a felhasználó rákattint. A szoftver felhasználói felülete teljesen hagyományosnak mondható, követi azt az elrendezést, ami a mai szoftverek esetében jól bevált, nem tartalmaz semmi olyan "furcsának" mondható elemet vagy logikát, ami a felhasználót elbizonytalanítaná, esetleg nem tudná kellőképpen értelmezni.

A felhasználók által támasztott minimális biztonsági követelmény, a felhasználók egyértelmű azonosíthatósága, a felhasználók fiókjának a relatív biztonsága.

A program hordozhatósága: A program minden olyan számítógépen fut, amelyik rendelkezik a Java Runtime Environment legalább 8-as verziójával, ami letölthető a következő hivatkozáson keresztül: [Java SE Runtime Environment 8 Downloads](#)

Fontos megjegyezni, hogy az adatbázis konzisztenciájának megőrzése érdekében a programnak egyszerre csak egy példánya futhat, másodszori futtatásra a program erre figyelmeztető üzenet után beárul.

2.) Szervezeti követelmények

- A programot háromréteg (model/view/controller) architektúrában kell felépíteni, amelyben a megjelenítés rétege elkülönül az adatkezelő modelltől és az adatokon végezhető műveleteket tartalmazó vezérlő rétegtől.
- A programhoz létre kell hozni a megfelelő adatbázist, amelyet a leírtak mentén kell megvalósítani. Az adatbázisban kell számú mintaadatnak kell lennie a megfelelő teszteléshez (táblánként minimum 20 sor).
- Az adatok bevitelénél törekedni a felhasználóbarát, hibamenetes megoldásokra. A program nem fogadhat el hibás bemenetet. Ahol lehetséges, biztosítsuk a kiválasztási lehetőséget, adatbevitelnél ellenőrizni kell az adatok helyességét (pl. az időpont ne lehessen múltbeli érték, a mennyiség ne lehessen negatív szám).
- Az implementációt Unit tesztekkel is tesztelni kell.

- A dokumentációnak tartalmaznia kell a feladat elemzését, felhasználói eseteit (UML felhasználói esetek diagrammal), a program szerkezetének leírását (UML osztálydiagrammal), valamint az adatbázis felépítésének leírását (egyedkapcsolati diagrammal).
- Az implementáció során törekedni kell a jól olvasható, átlátható, tiszta kód írására¹.

3.) Küls követelmények

A fejleszt köteles a gyakorlatvezet összes tanácsát és útmutatását megfontolni; bármilyen elakadása folyamán segítséget kérni gyakorlatvezet jét l.

¹ (Clean Code: A Handbook of Agile Software Craftsmanship by Robert C. Martin; Chapter 1-10)

Fejlesztési dokumentáció

I. Adatbázis terv

1.) Adatbázis sémák

TÁBLA NEVE		
ATTRIBUTUMOK	TÍPUSOK	LEÍRÁS

COMPUTERS		
ID	BIGINT	A számítógép azonosítója, elsődleges kulcs (PK)
NAME	VARCHAR(10 CHAR)	A számítógép egyedi neve (felhasználóbarát azonosítója) (UNIQUE)
CPU	VARCHAR(50 CHAR)	A processzor típusa, tulajdonságai
MOTHERBOARD	VARCHAR(50 CHAR)	Az alaplap típusa, tulajdonságai
MEMORY	VARCHAR(50 CHAR)	A memória tulajdonságai
VGA	VARCHAR(50 CHAR)	A grafikus tulajdonságai
MASS_STORAGE	VARCHAR(50 CHAR)	A háttértár és esetleges optikai meghajtók tulajdonságai
OS	VARCHAR(50 CHAR)	A számítógépen futó operációs rendszer
BUSY	BOOLEAN	A PC használatban van-e
CREATED_AT	TIMESTAMP	Rögzítés ideje
UPDATED_AT	TIMESTAMP	Utolsó módosítás ideje

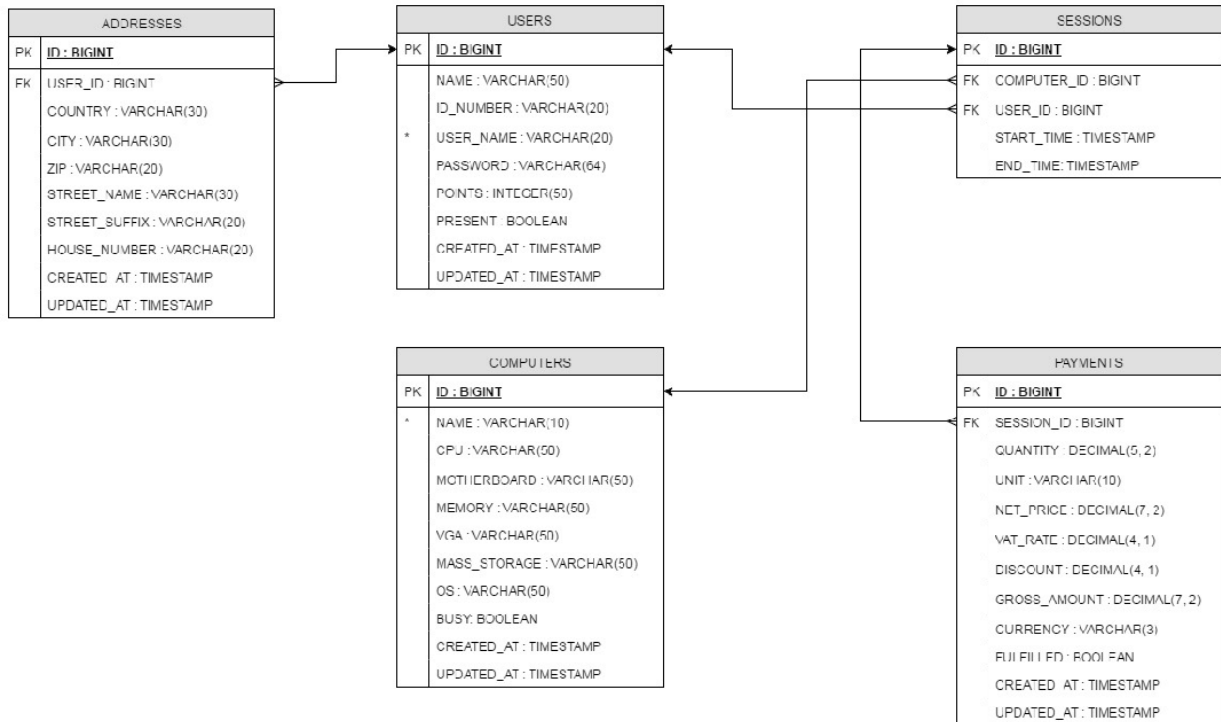
USERS		
ID	BIGINT	Az ügyfél azonosítója, elsődleges kulcs (PK)
NAME	VARCHAR(50 CHAR)	Az ügyfél neve
ID_NUMBER	VARCHAR(20 CHAR)	Az ügyfél személyi igazolvány száma
USER_NAME	VARCHAR(20 CHAR)	Az ügyfél felhasználóneve
PASSWORD	VARCHAR(64 CHAR)	Az ügyfél jelszava hashelve
POINTS	INTEGER	Az ügyfél hűségpontjai
PRESENT	BOOLEAN	Az ügyfél be van-e jelentkezve
CREATED_AT	TIMESTAMP	Rögzítés ideje
UPDATED_AT	TIMESTAMP	Utolsó módosítás ideje

ADDRESSES		
ID	BIGINT	A cím azonosítója, els dleges kulcs (PK)
USER_ID	BIGINT	Idegenkulcs (FK) az ügyfélre
COUNTRY	VARCHAR(30 CHAR)	Ország neve
CITY	VARCHAR(30 CHAR)	Település neve
ZIP	VARCHAR(20 CHAR)	Írányítószám
STREET_NAME	VARCHAR(30 CHAR)	Közterület neve
STREET_SUFFIX	VARCHAR(20 CHAR)	Közterület jellege
HOUSE_NUMBER	VARCHAR(20 CHAR)	Házszám és kiegészít információk
CREATED_AT	TIMESTAMP	Rögzítés ideje
UPDATED_AT	TIMESTAMP	Utolsó módosítás ideje

SESSIONS		
ID	BIGINT	A munkamenet azonosítója, els dleges kulcs (PK)
USER_ID	BIGINT	Idegenkulcs (FK) az ügyfélre
COMPUTER_ID	BIGINT	Idegenkulcs (FK) a számítógépre
START_TIME	TIMESTAMP	Munkamenet kezdete
END_TIME	TIMESTAMP	Munkamenet vége

PAYMENTS		
ID	BIGINT	A befizetés azonosítója, els dleges kulcs (PK)
SESSION_ID	BIGINT	Idegenkulcs (FK) a munkamenetre
QUANTITY	DECIMAL(5,2)	A munkamenet hossza órákban kifejezve
UNIT	VARCHAR(10 CHAR)	Mennyiségi egység (óra)
NET_PICE	DECIMAL(7,2)	Óránkénti nettó ár
VAT_RATE	DECIMAL(4,1)	A díjszámításhoz használt ÁFA kulcs (százalékban)
DISCOUNT	DECIMAL(4,1)	Az ügyfélnek járó kedvezmény (százalékban)
GROSS_AMOUNT	DECIMAL(7,2)	A munkamenet bruttó összege
CURRENCY	VARCHAR(3 CHAR)	Befizetés pénzneme
FULFILLED	BOOLEAN	Teljesítette-e a befizetés
CREATED_AT	TIMESTAMP	Befizetés ideje
UPDATED_AT	TIMESTAMP	Utolsó módosítás ideje

2.) Egyed-kapcsolat diagram

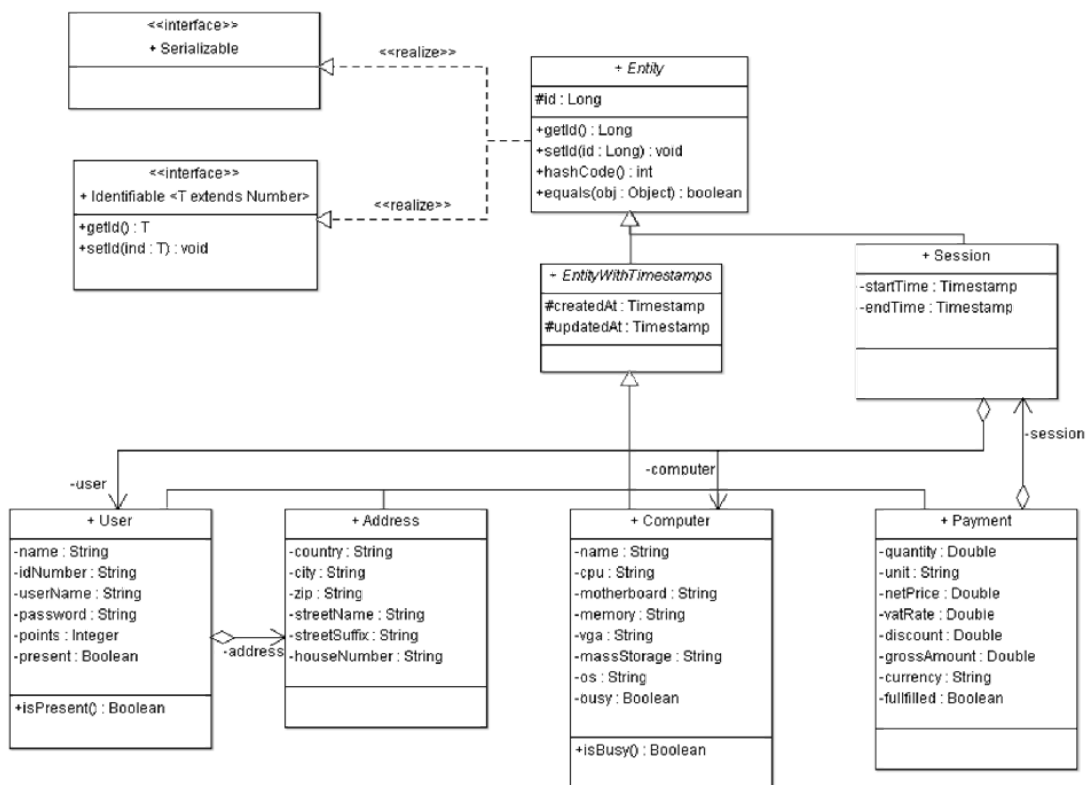


5. ábra: Egyed-kapcsolat diagram

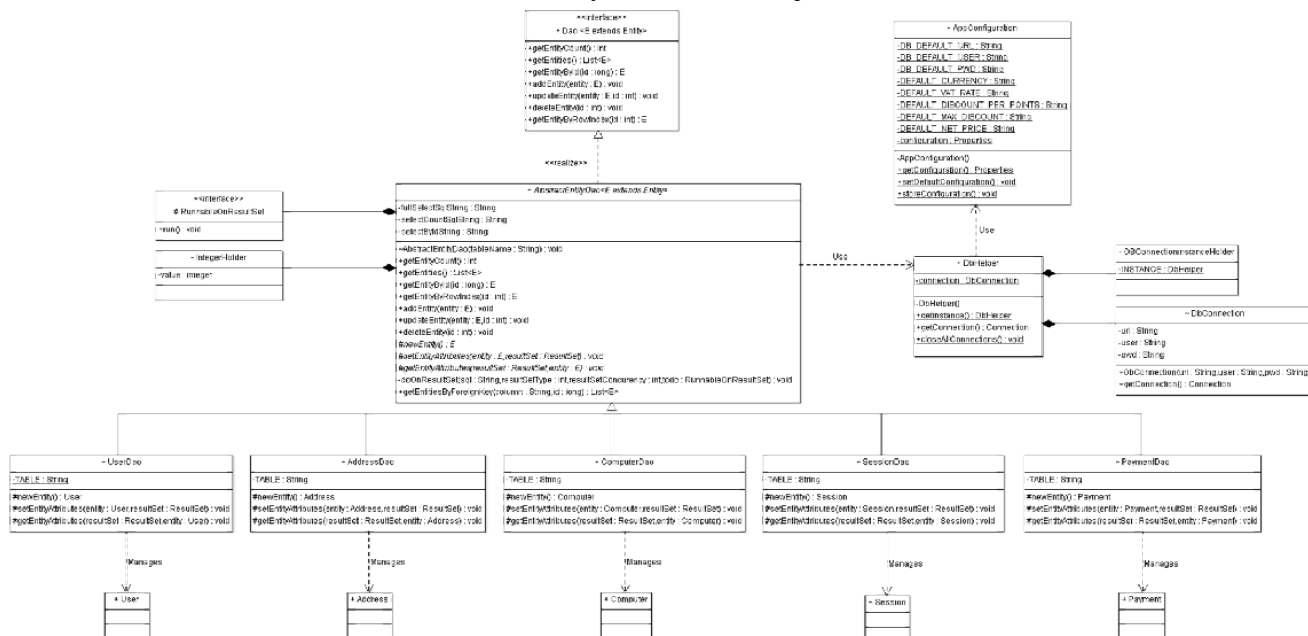
II. Statikus terv

1.) osztálydiagram

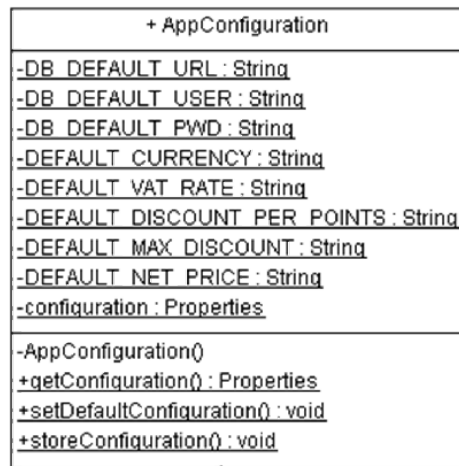
Az adatbázisban tárolt adatok reprezentálásában a következő osztályok fognak részt venni:



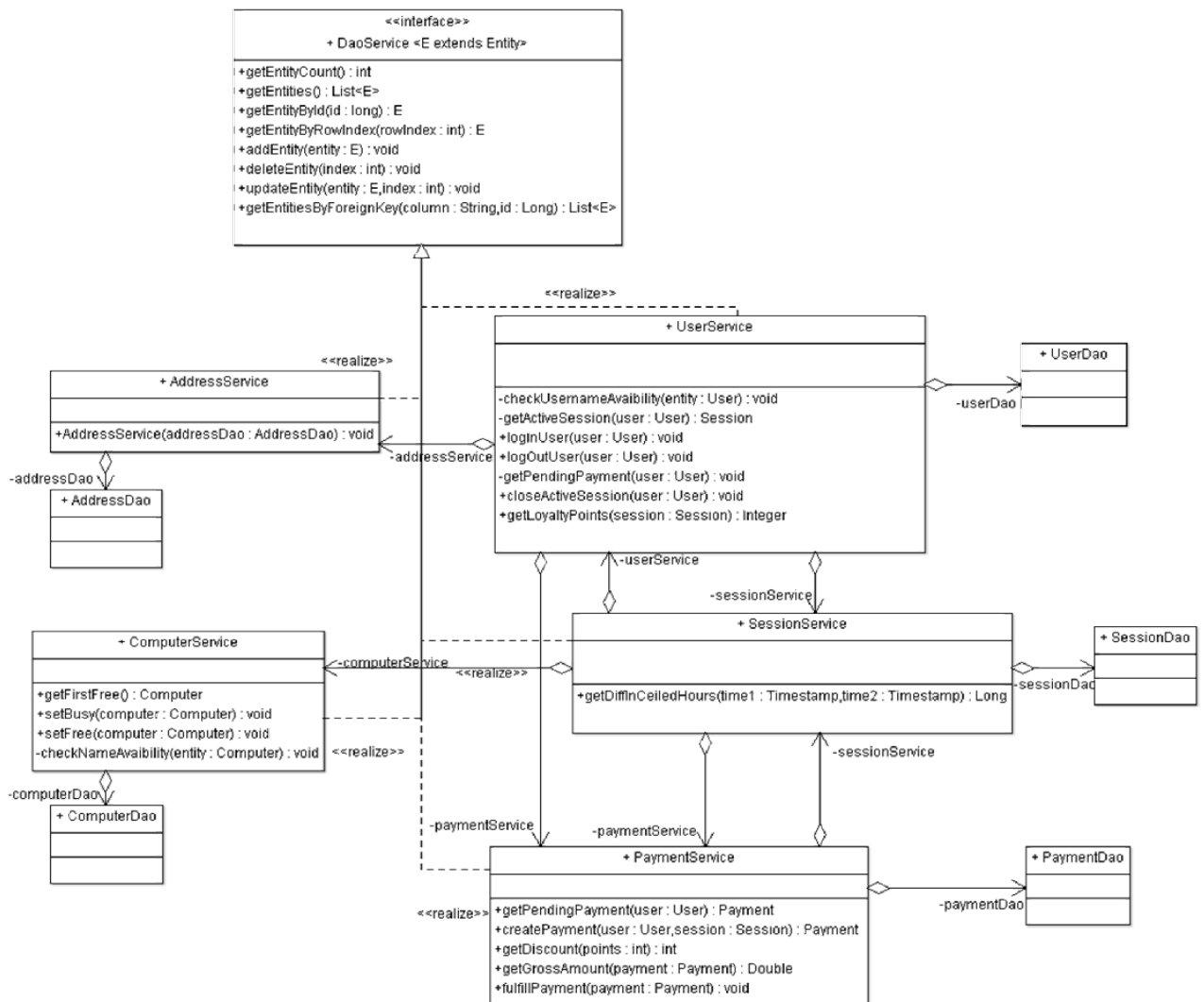
Az adatbázis-m veletekért a következ osztályok felelnek majd:



Az alkalmazás alapértelmezett beállításait, és felhasználói konfigurációját az `AppConfiguration` osztály fogja tárolni, illetve érvényesíteni a `java.util.Properties` osztály felhasználásával.



A következ Service-ken keresztül érhetjük el a Dao (Data Access Object) szolgáltatásait:



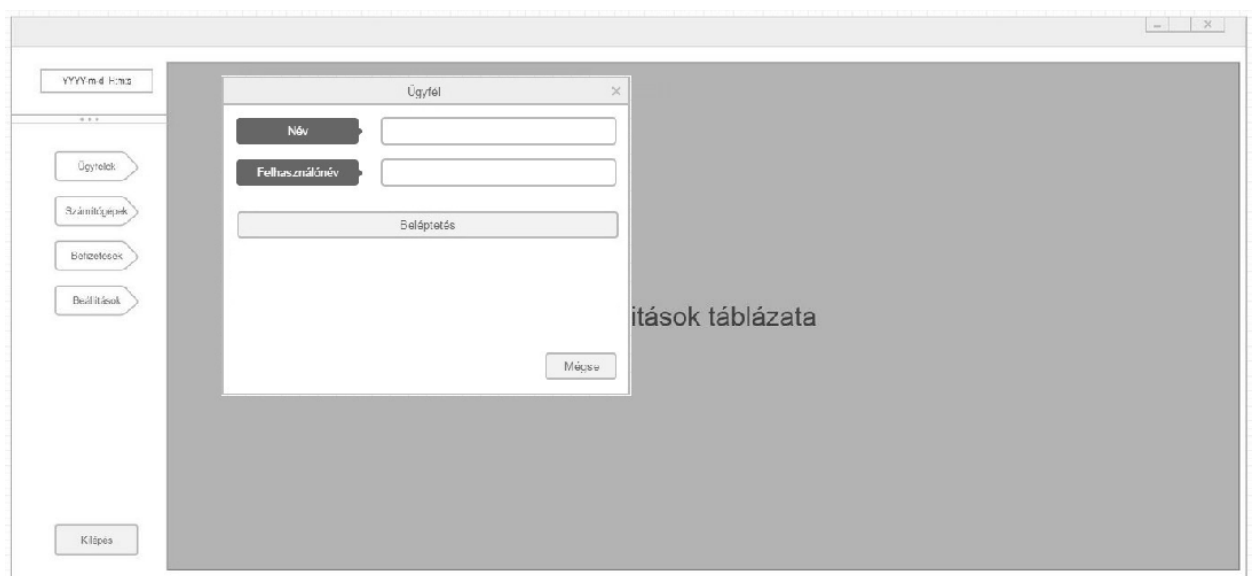
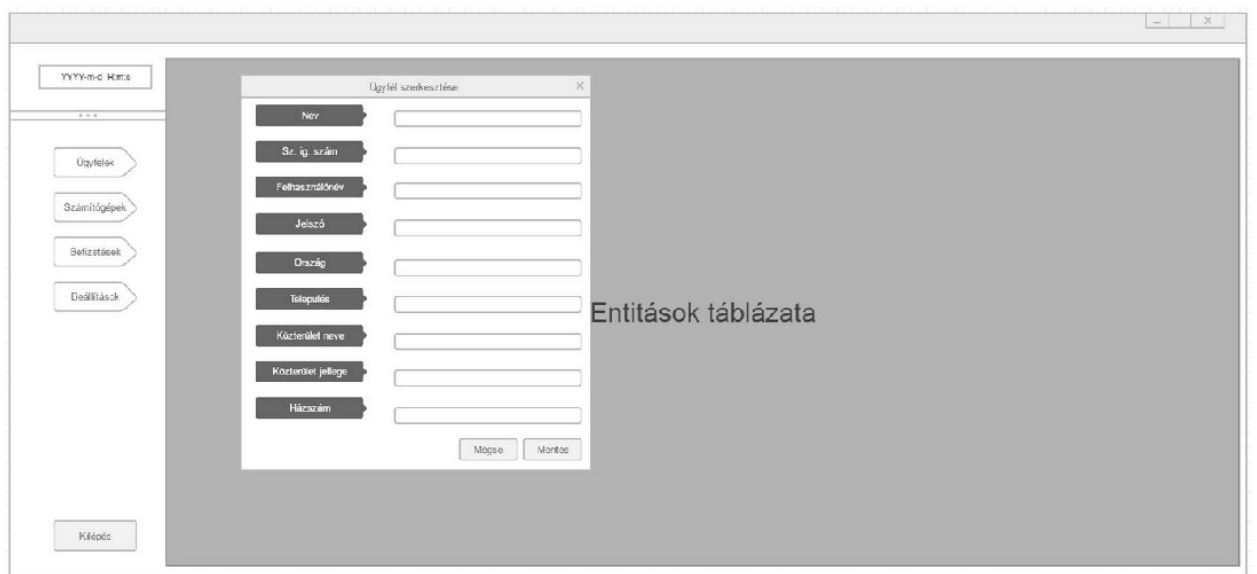
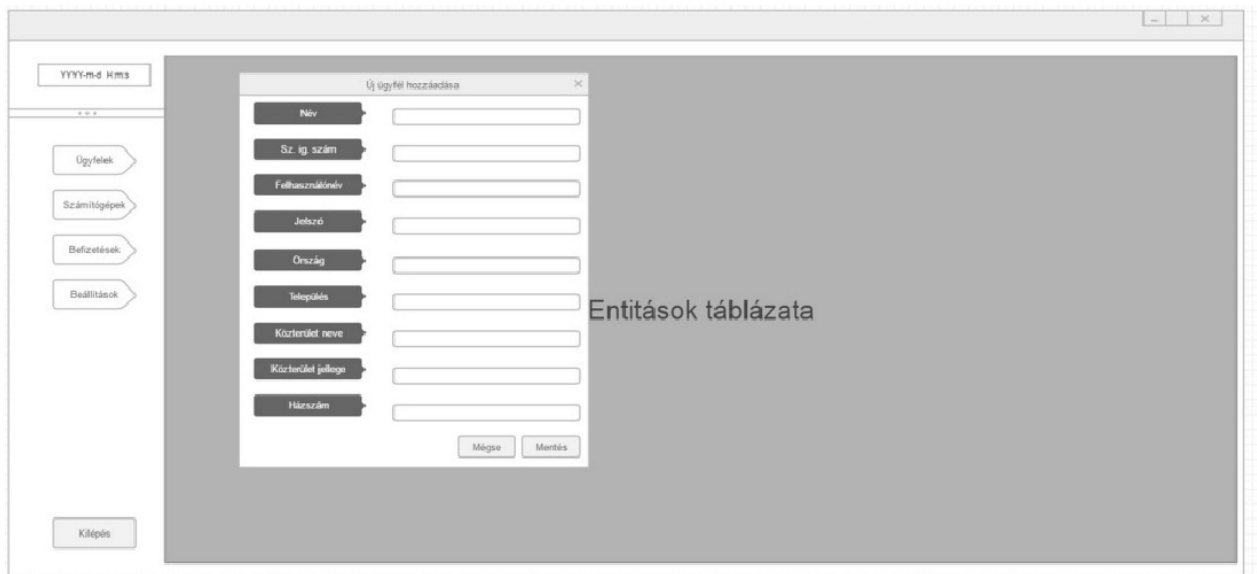
III. Felhasználói-felület modell

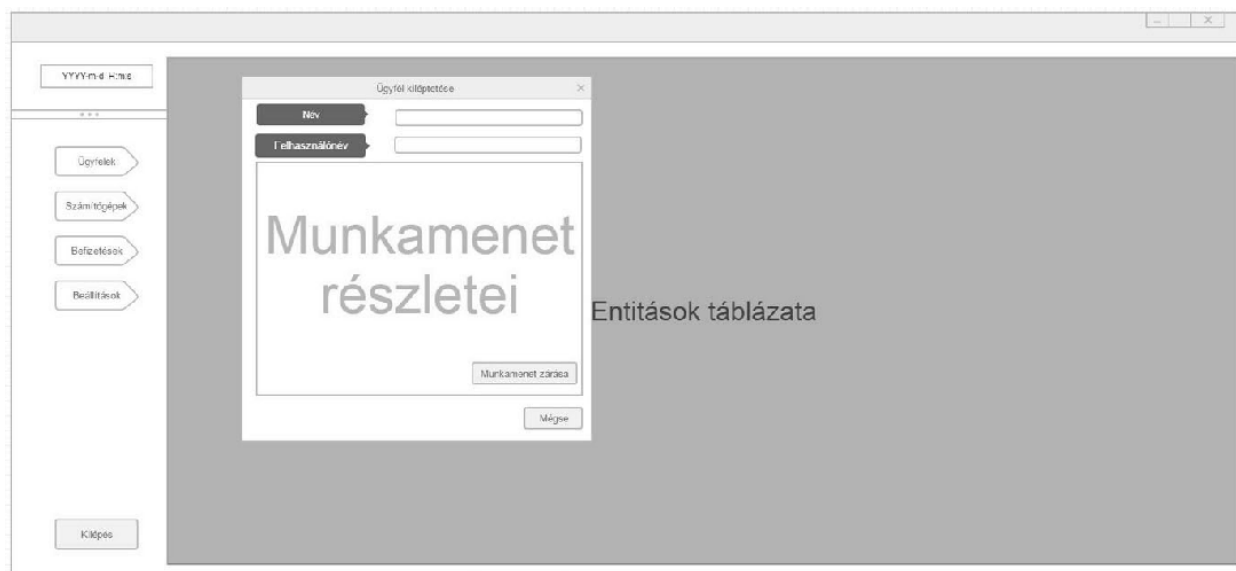
Az alkalmazás felhasználói felülete jól áttekinthető, a különböző ablakok tartalma logikailag összetartozó információkat jelenít meg. Nyelve magyar, és az egyes gombok, menük feliratai is erre teljesen rávilágítanak arra, mi történik, ha a felhasználó rákattint. Az esetleges összefüggő lépések sorozatait ugyanabban a párbeszéd-ablakban jelennek meg, egyértelműen tájékoztatva a felhasználót a folyamat állapotáról (Ilyen például a felhasználó munkamenetének zárása, befejezésének megerősítése kiléptetés előtt).

Az esetleges hibüzenetek szabványos felugró ablakban jelennek meg, szövegük egyértelmű tájékoztatást ad a hiba természetéről, az elhárításához szükséges teendőkről.

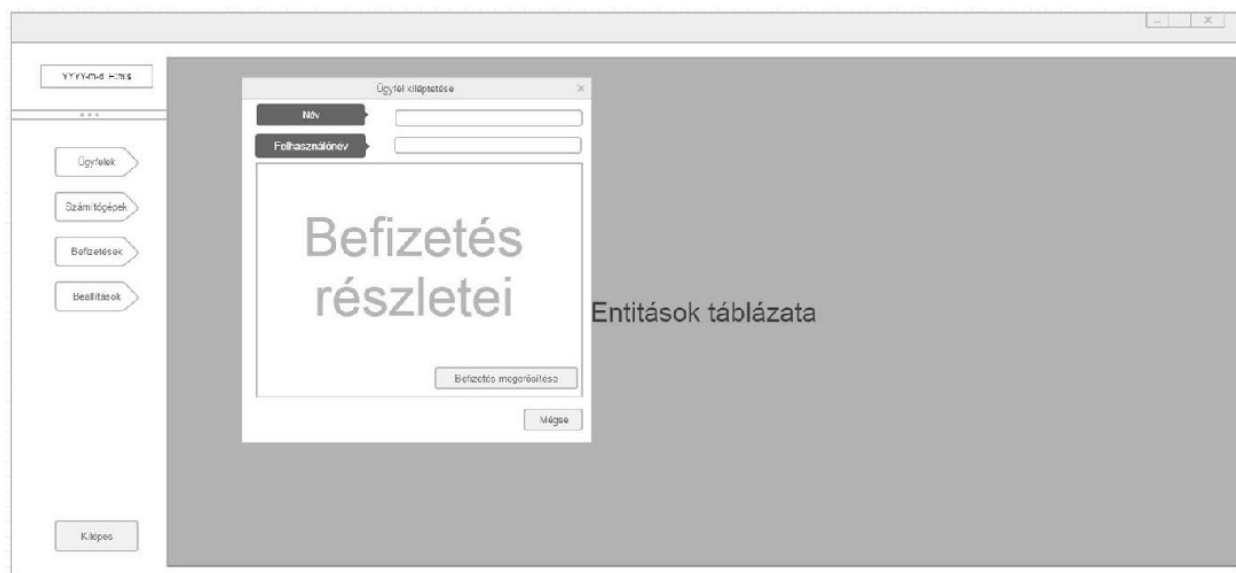
A felhasználói felület vázlatos tervét a következő néhány kép mutatja be.



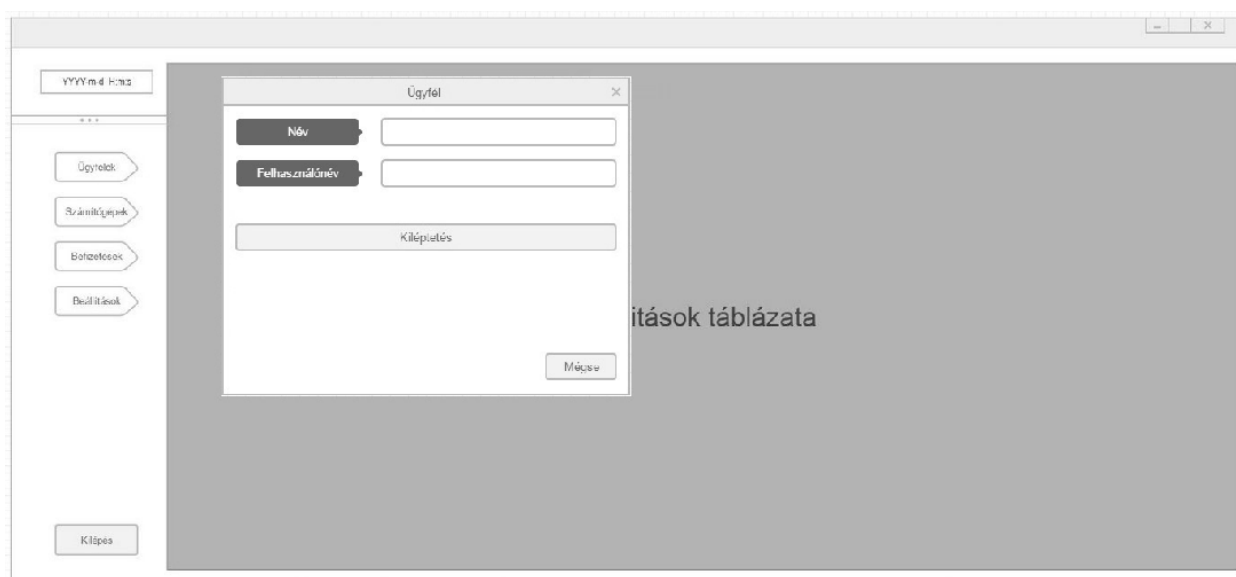




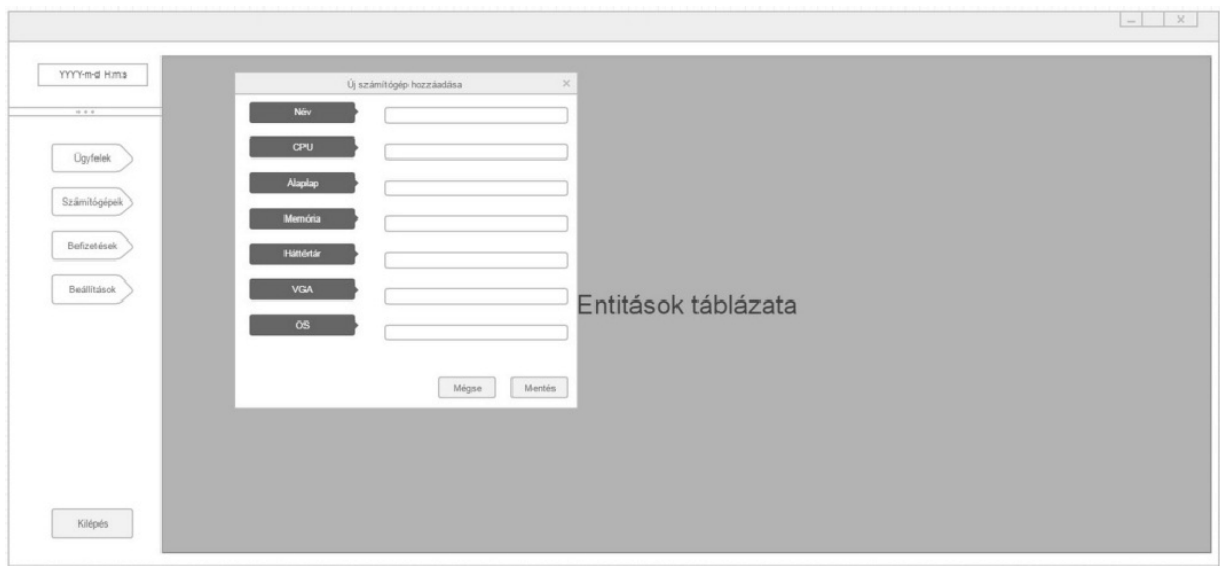
Entitások táblázata



Entitások táblázata



Entitások táblázata





IV. Implementáció

1.) Technológiák és API-k

A fejlesztés során több külső könyvtárat is felhasználunk:

Derby - nyílt forráskódú, relációs adatbázis-megvalósítás.

JUnit - egy egységteszt keretrendszer Java programozási nyelvhez.

Maven - egy szoftver, amelyet szoftverprojektek menedzselésére és a build folyamat automatizálására lehet használni.

Project Lombok – egy kód generáló, melynek segítségével annotációk használatával generálhatunk különböző egyszerűbb (például getter, setter, toString) metódusokat.

V. Tesztelés

1.) Manuális tesztek

A manuális tesztelés a felhasználói felületen keresztül történik, mely során hibás, hiányos értékek rögzítését, kétszeres be-, illetve kijelentkezést, próbálunk végrehajtani. Megkíséreljük egy felhasználó beléptetését a rendszer egy olyan állapotában, amikor nincs szabad számítógép.

2.) Egység tesztek

PaymentServiceTest – a rendszer beállításainak figyelembevételével teszteli a befizetésekhez számított kedvezmény százaléértékének számítását, a bruttó befizetendő összeg kalkulálásának helyes mivoltát.

testGetDiscount – egyszer teszt a felhasználói kedvezmény százaléértékének kiszámítására.

testGetGrossAmount_zeroVatRate – teszt a bruttó befizetendő összeg kiszámítására 0%-os áfakulcs esetén.

testGetGrossAmount_fullVatRate – teszt a bruttó befizetendő összeg kiszámítására 100%-os áfakulcs esetén

testGetGrossAmount_zeroDiscount – teszt a bruttó befizetendő összeg kiszámítására 0%-os kedvezmény esetén

testGetGrossAmount_fullDiscount – teszt a bruttó befizetendő összeg kiszámítására 100%-os kedvezmény esetén

SessionServiceTest – teszt a munkamenet számlázandó idő tartamának helyes kalkulációjára.

testGetDiffInCeiledHours_startEqualsEnd – a munkamenet kezdete megegyezik a munkamenet végével, nincs számlázandó mennyiség.

testGetDiffInCeiledHours_startIsBeforeEnd – a munkamenet kezdete nem egyezik meg a munkamenet végével, van számlázandó mennyiség. A módszer első paramétere a kezdési időpont, a második a végzési időpont.

testGetDiffInCeiledHours_startIsAfterEnd – a munkamenet kezdete nem egyezik meg a munkamenet végével, van számlázandó mennyiség. A módszer első paramétere a végzési időpont, a második a kezdési időpont.

testGetDiffInCeiledHours_testCalculation – a munkamenet kezdete nem egyezik meg a munkamenet végével, van számlázandó mennyiség. Tesztelés a megkezdett órák számlázásának biztosítására.

UserServiceTest – teszt a feladat feltételeinek megfelelő pontszám kalkuláció biztosítására.

getLoyaltyPoints – egyszer tesztelés a feladat leírásában szereplő időpontok használatával.

getLoyaltyPoints_aFullDay – egyszer tesztelés egy teljes napot kitöltő munkamenet szerinti pontszámításhoz. A munkamenet kezdete és vége egyaránt egész óra.

`getLoyaltyPoints_alwaysLowerIfStartWasNotInAnExactHour` – egyszer
teszetestet egy teljes napot kitölt munkamenet szerinti pontszámításhoz. A munkamenet kezdete
és vége nem egész óra.