

# **ZEUSZ PROJEKT**

Megvalósítás

A csapat:

„Kentaur Development Team”

Projektvezető:	Dori Bence
Kapcsolattartó:	Schönberger Zsófia
Adminisztrátor:	Esse Zsolt
Demonstrátor:	Balogh Tamás

## **A megrendelő igényei**

A megrendelő egy olyan szoftverrendszer elkészítésére adott megbízást, amely a „Szabadítsd fel az erőforrást, ha már nem használod” Társaság Szabadszellemű Magániskolája számára nyújt támogatást a mindennapi ügymenetekhez.

A hallgatók a rendszer segítségével tudnak a megfelelő időszakban tantárgyakat felvenni, illetve ezeket leadni. Lehetőségük van kérelemmel jelezni a vezető felé, ha következő félévben passzív félévet szeretnének, továbbá azt, ha el szeretnék hagyni az iskolát. Utóbbi esetben archiválódnak az adatok.

Felvett tantárgyaikról módjukban áll listát kérniük és egy gombnyomással listászerű órarendet is lekérniük. Továbbá lehetőségük van a már teljesített tárgyakat is kilistázni, érdemjegyeiket megnézni.

A tanárok tudni kell tantárgyaikat létrehozni, azok adatait, leírását ott megadni, majd létrehozási szándékot kérelemmel vezető felé jelezni. Ezeket módosítani, törölni szintén csak vezetőhöz benyújtott kérelmekkel lehet.

Tanároknak a hallgatók értékelését is biztosítja a rendszer. Megfelelő évfolyam , tantárgy kiválasztása után kilistázhatja a diákokat és beírhatja az érdemjegyeket. A teljesítményről itt van lehetőség kimutatás kérésére. A kimutatás készítését a vezetőtől kell kérvényezni. Ezen a felületen tud a tanár csoportos üzenetet küldeni az adott tantárgyat hallgatók számára.

Minden személynek (hallgató vagy tanár) módjában áll saját adatainak módosítására, amit szintén a vezetőnek kell küldeni, kérelem formájában.

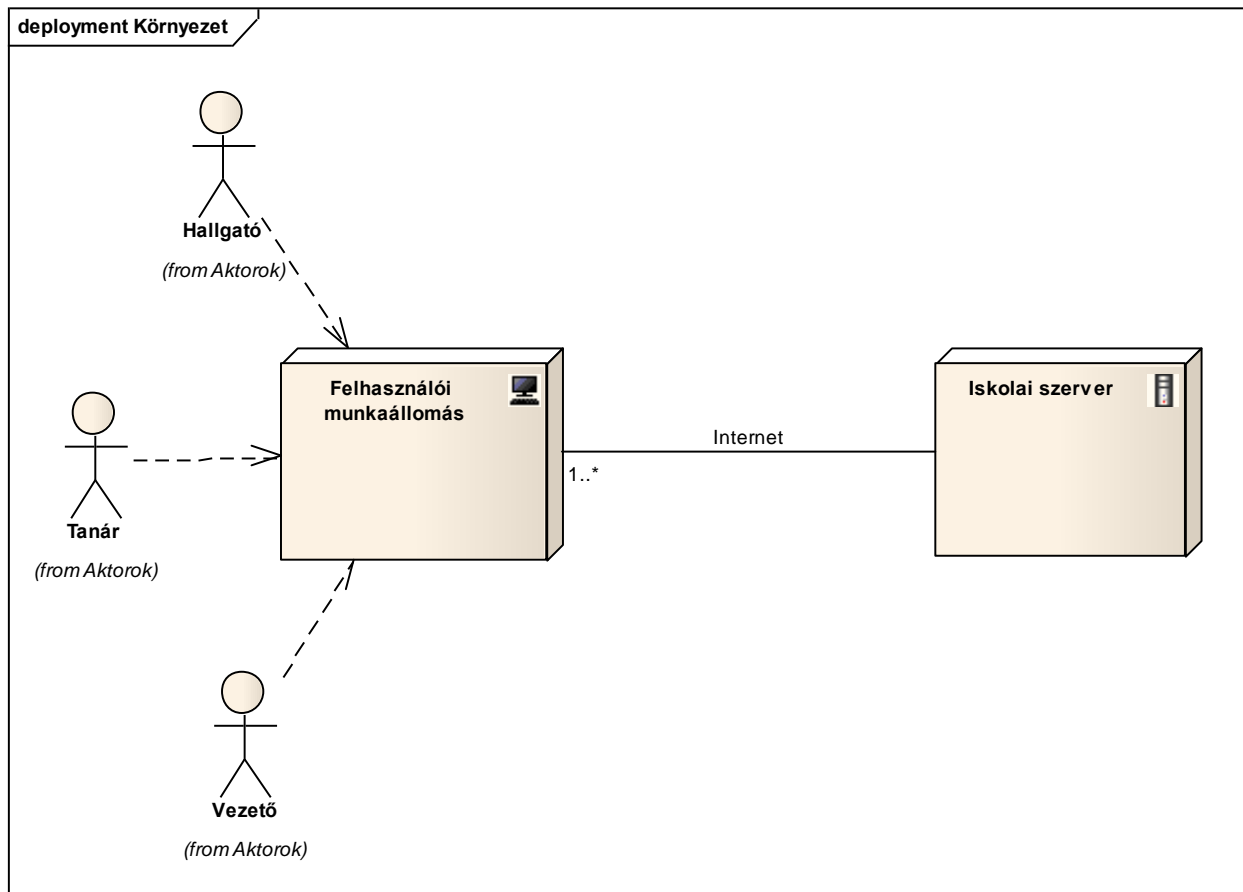
A vezető egyben az adminisztrátori feladatkört is betölti, teljes hozzáférése van minden adathoz, tárgyhoz, nyilvántartáshoz. Ő tud felvenni új személyeket a rendszerben. A hallgatóktól és a tanároktól beérkezett kérelmeket ő bírálja el, aminek eredményéről visszajelzést küld, továbbá elvégzi a módosításokat, ha szükséges.

A rendszer első használatakor lehet egy vezetőt regisztrálni. Utána már csak a vezető hozhat létre új személyeket. Rendszerbe belépni előre generált egyedi zeusz kóddal és jelszóval lehet a felhasználóknak.

## A rendszer környezete

A tervezett rendszer egy kliens-szerver környezetben fog működni. A szerver az iskola területén fog elhelyezkedni, melyhez a felhasználók egy univerzális kliens felületen keresztül csatlakozhatnak.

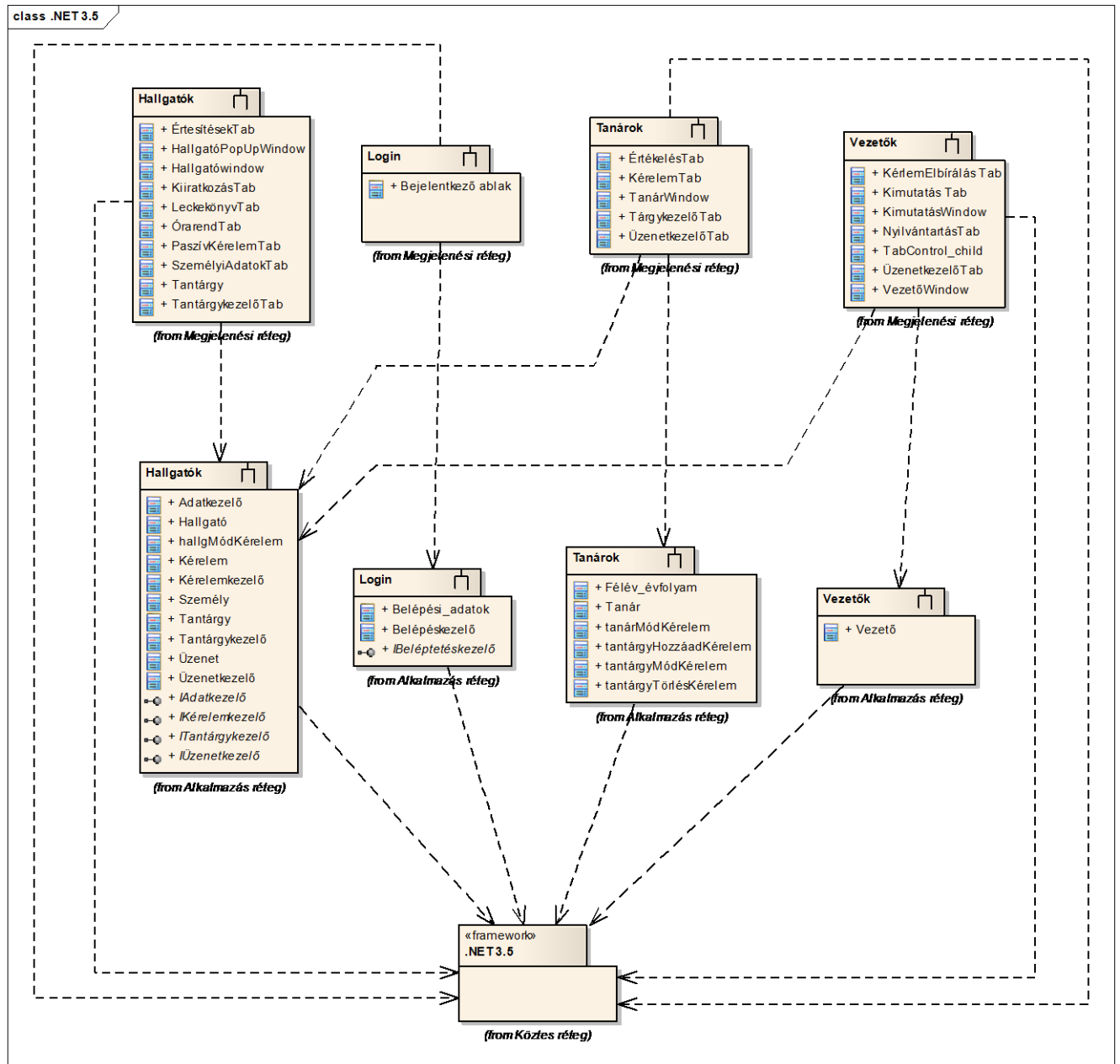
A szervert a Microsoft Windows Server 2008 R2 operációs rendszere működteti, a kliens oldali elvárás legalább a Microsoft Windows XP SP3. A program .NET 3.5-ös verziójú keretrendszerben íródott, ezért ennek mind szerver, mind kliens oldalon telepítve kell lennie.



A rendszer környezete

## A rendszer szerkezete

A tervezés során a szoftvert alrendszerekre bontottuk figyelembe véve a korábban meghatározott funkciókat és a modularizációt. Meghatároztuk azokat a külső elemeket és osztályszintű kapcsolatokat, amelyek szükségesek a szoftver implementálásához.



Alrendszerek és függőségeik

A szoftver felső, megjelenési szintjén található négy alrendszerbe lett csoportosítva, amelyek a felhasználói interfészek megjelenítéséhez szükséges osztályokat tartalmazzák figyelembe véve a különböző jogkörű felhasználók elvégezhető műveleteit.

A szoftver alkalmazási szintjén található alrendszerek a program működéséhez szükséges adatok tárolását, kezelését és hozzáférhetőségét biztosítják. A felső szinten lévő alrendszerek ezeket az itt lévő alrendszereket használják a működésükhöz.

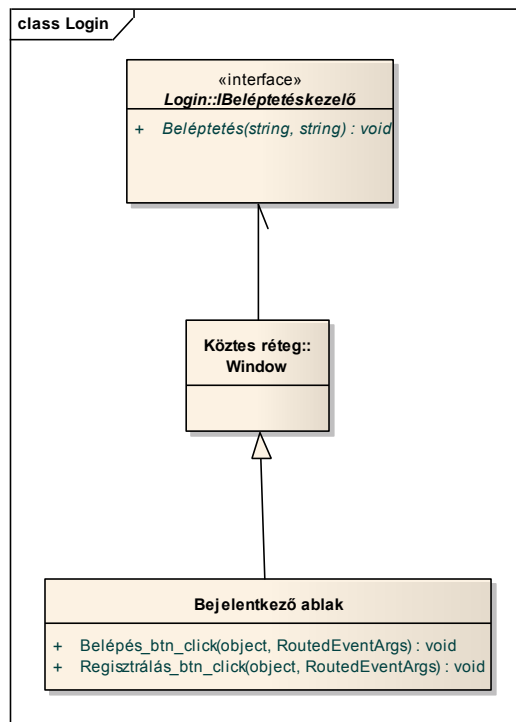
Végül a köztes szinten a .NET 3.5-ös keretrendszerét szimbolizáló csomag áll.

### Megjelenítési réteg:

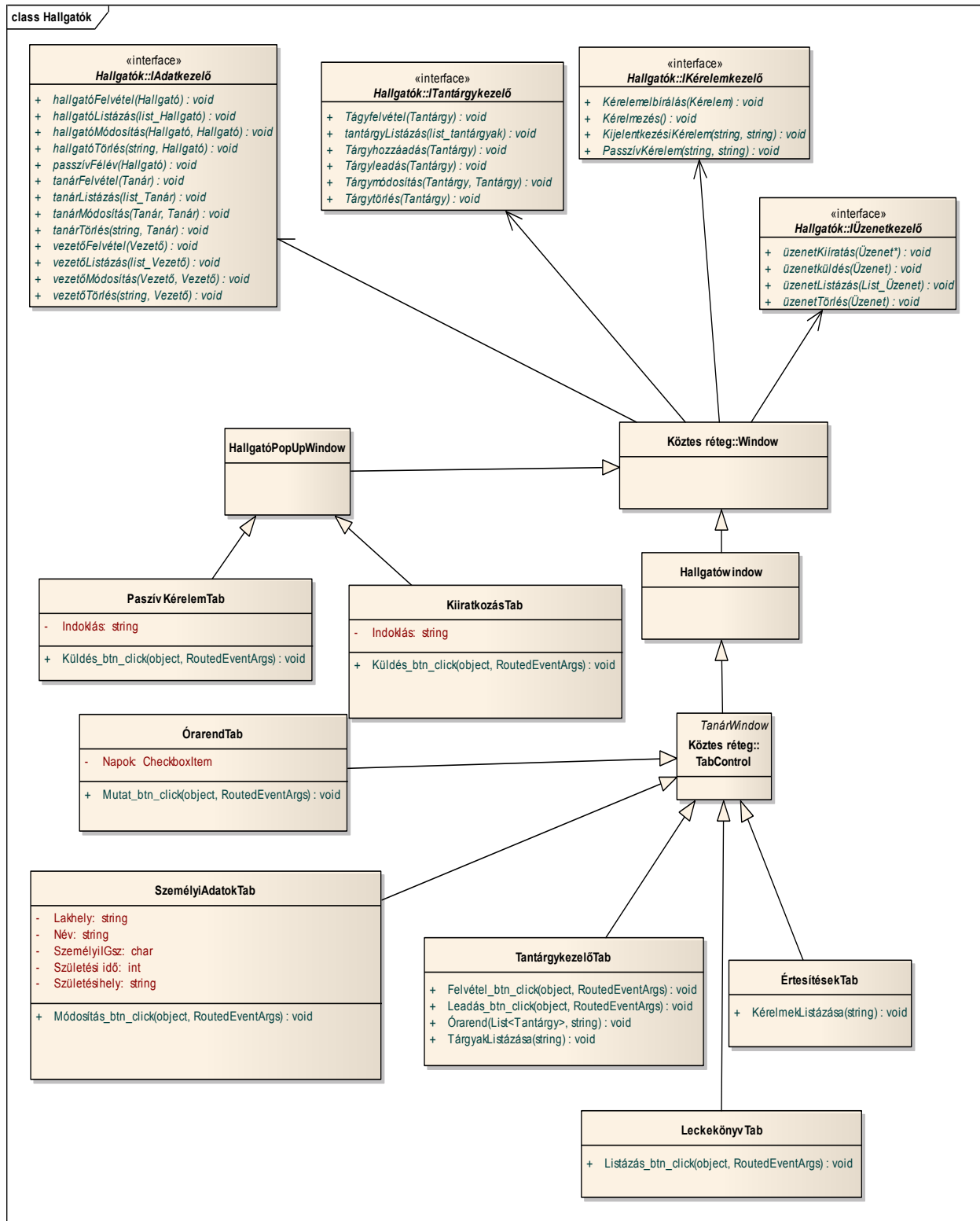
A login alrendszer egy bejelentkező képernyőt foglal magában, melyben lehetőségünk van bejelentkezni a rendszerbe, szükség esetén új vezetőt tudunk regisztrálni.

A hallgató alrendszer a hallgató számára elérhető form és azok tab feladatait szemlélteti.

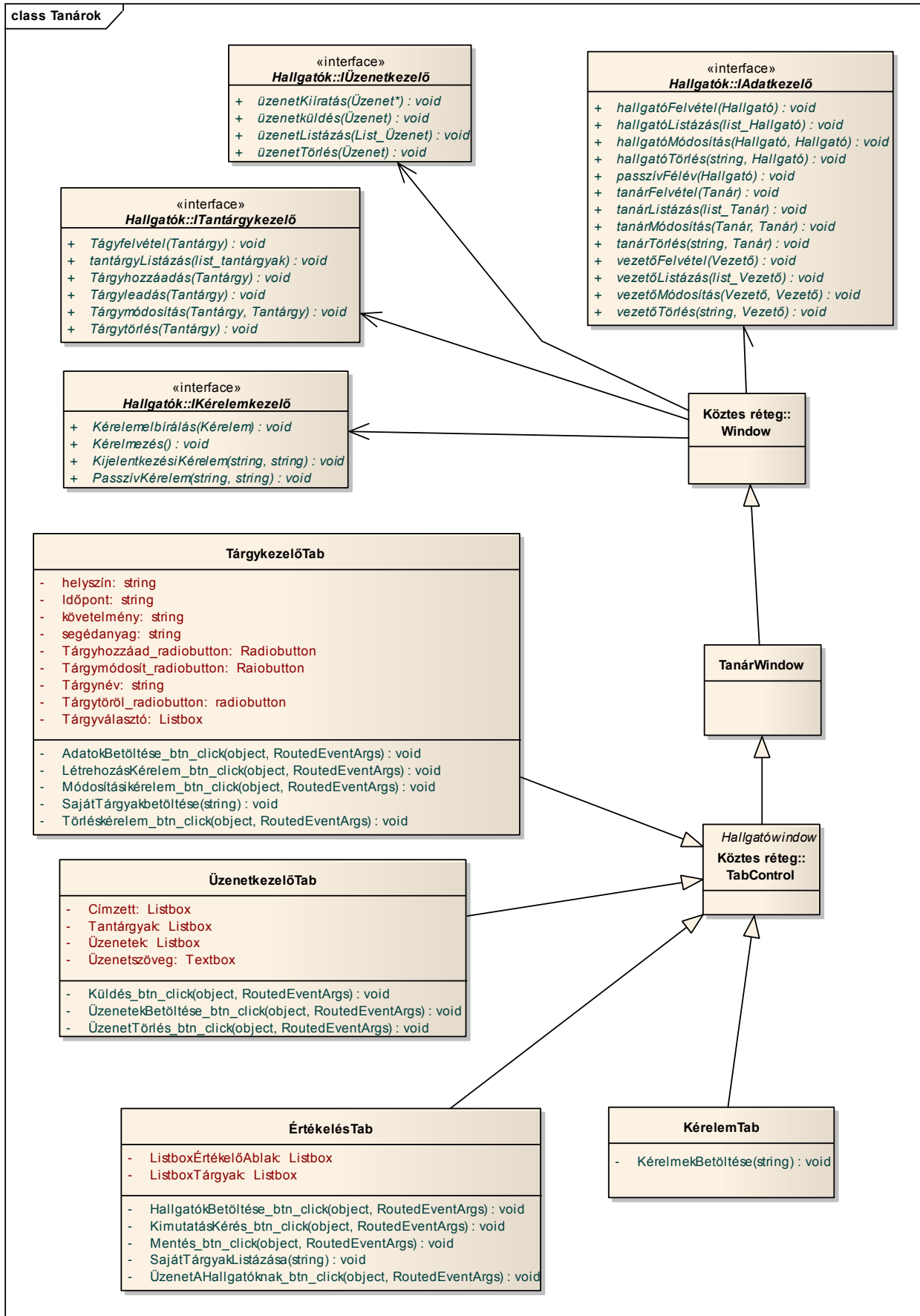
Hasonló szempontok szerint épül fel a vezetői és tanári alrendszer is, melyek az elvárt funkciók megvalósításának figyelembevételével szemlélteti a megjelenési réteg kapcsolódását az alkalmazásréteg funkcióihoz, úgy mint üzenetkezelés, tantárgykezelés, üzenetküldés és személyügyi adminisztráció.



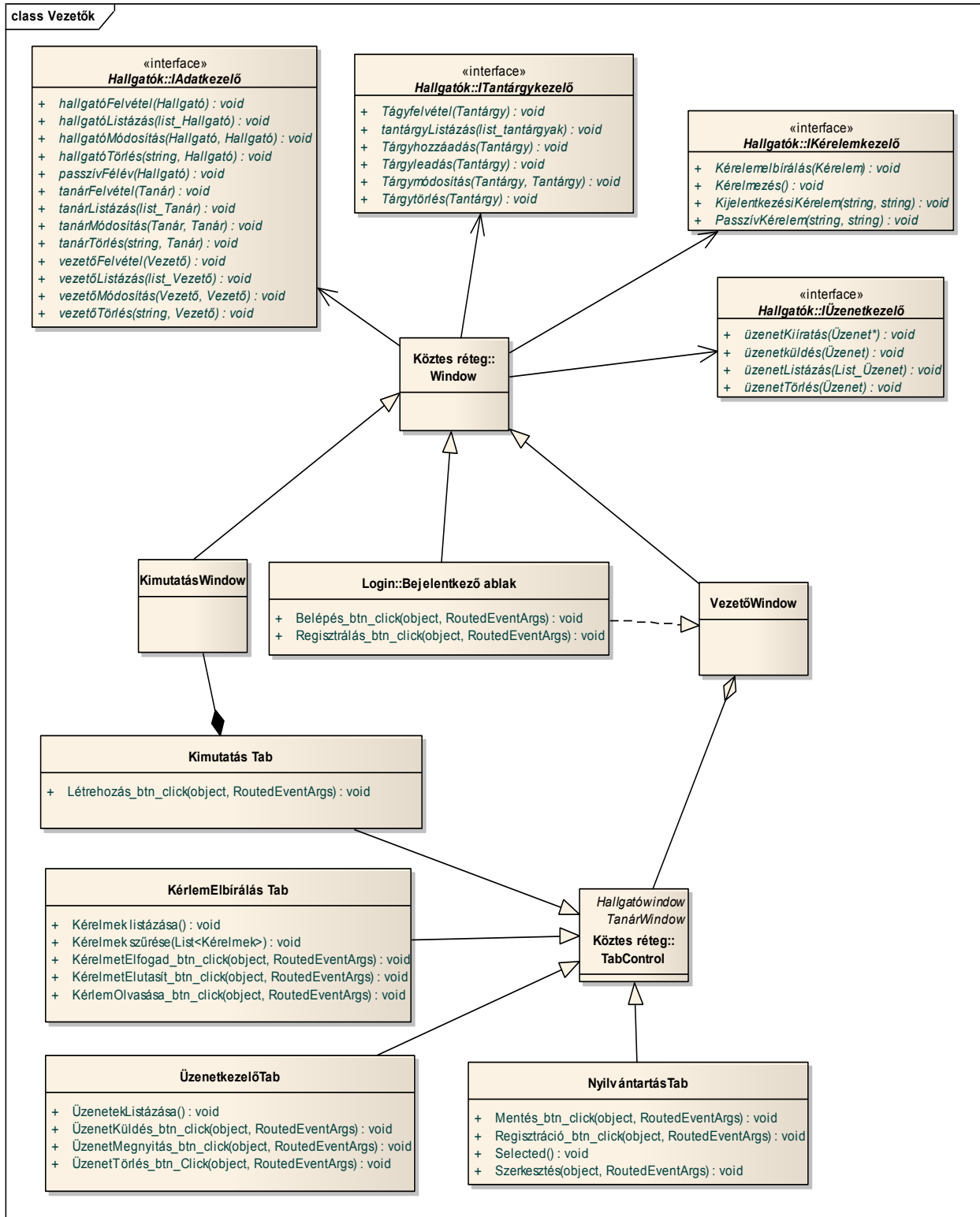
A login alrendszer



A hallgató alrendszer



A Tanárok alrendszer



A Vezetői alrendszer

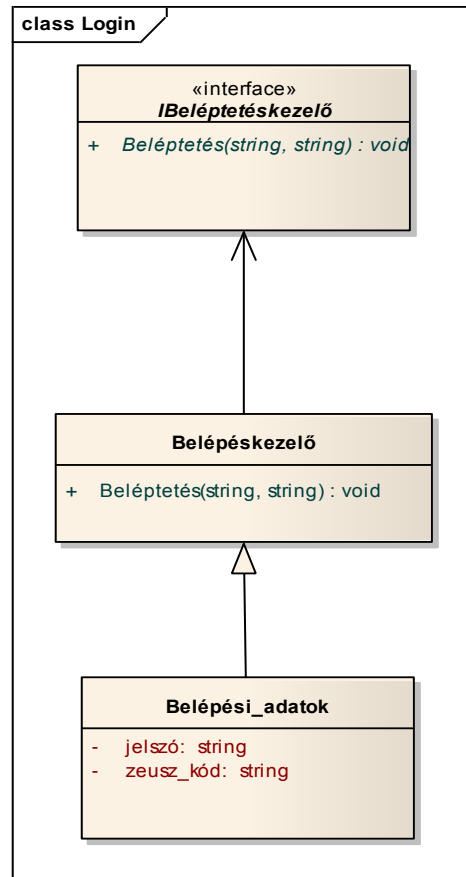


## Alkalmazási réteg:

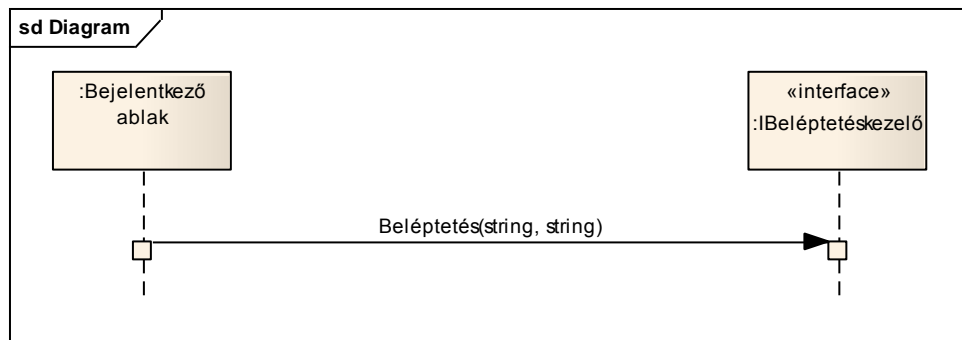
A rendszerelemek együttműködését, a közöttük végbemenő üzenetváltásokat és kapcsolatokat szemléltetik a következő ábrák.

### Login alrendszer:

Ez az alrendszer felelős a személyek belépésének biztosításáért.



A rendszerelemek együttműködését, a közöttük végbemenő üzenetváltásokat szemléltetik a következő ábrák.

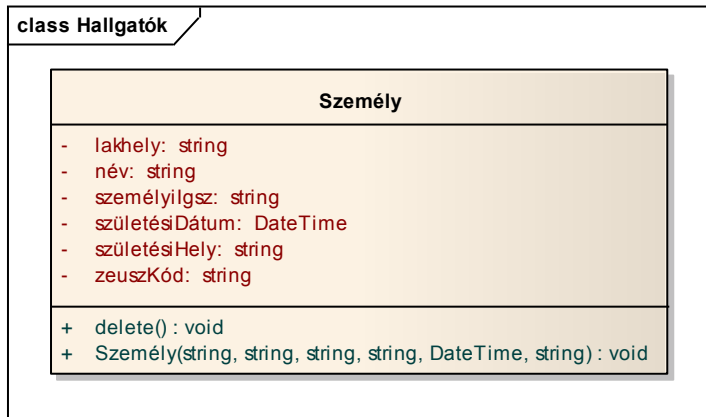


Belépés

### Hallgatók alrendszer:

Ez az alrendszer felelős a különböző tanulók adatainak, kimutatásainak és eredményeinek tárolásáért, kezeléséért és védelméért.

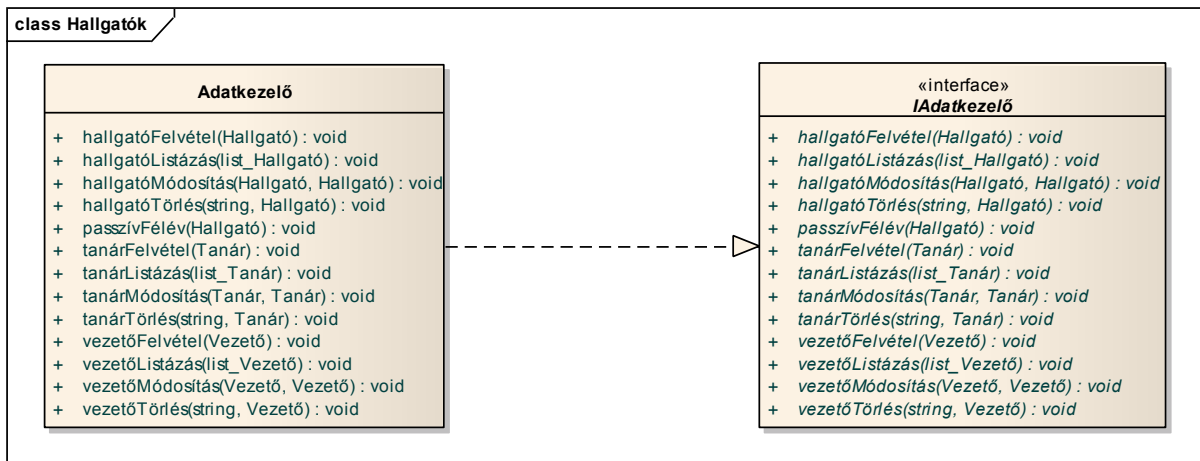
Személy osztály:



A személy osztály lesz a közös őse az eltárolandó felhasználóinknak, az ő funkcióit bővíti ki a Hallgató, Tanár és Vezető osztályok.

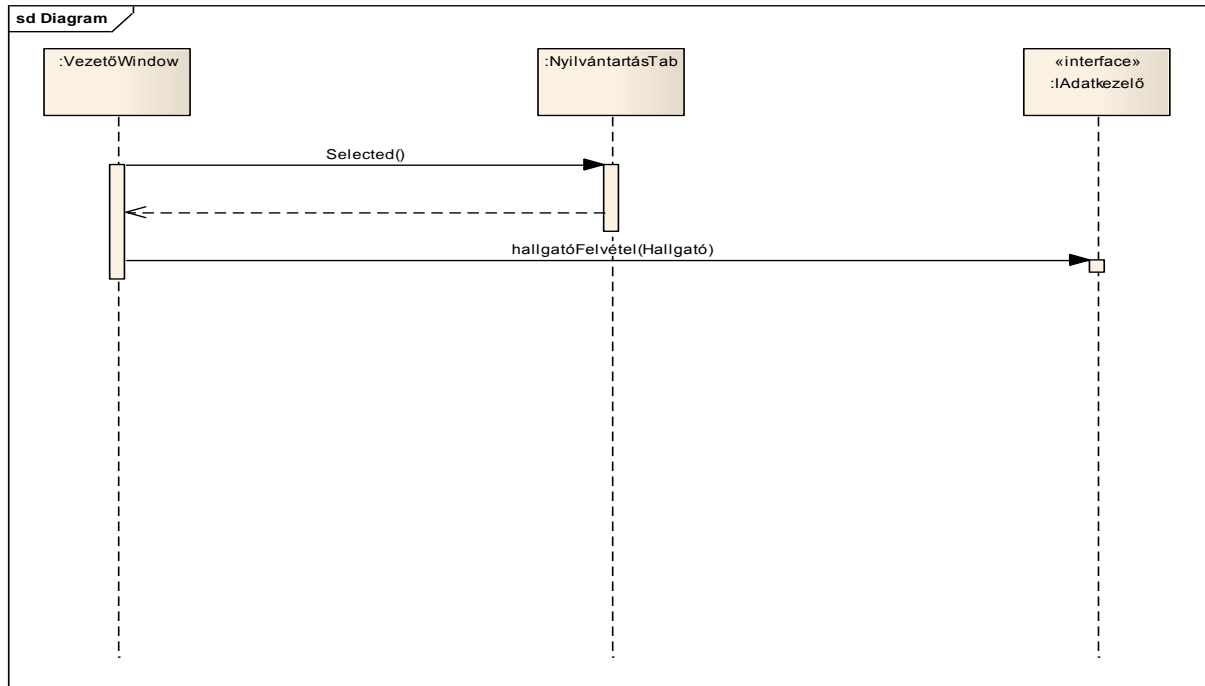
Létrehozáskor meg kell adni név, cím, személyi igazolványszám, születési dátum, születési hely és zeusz kódot.

Adatkezelő osztály és IAdatkezelő interface:

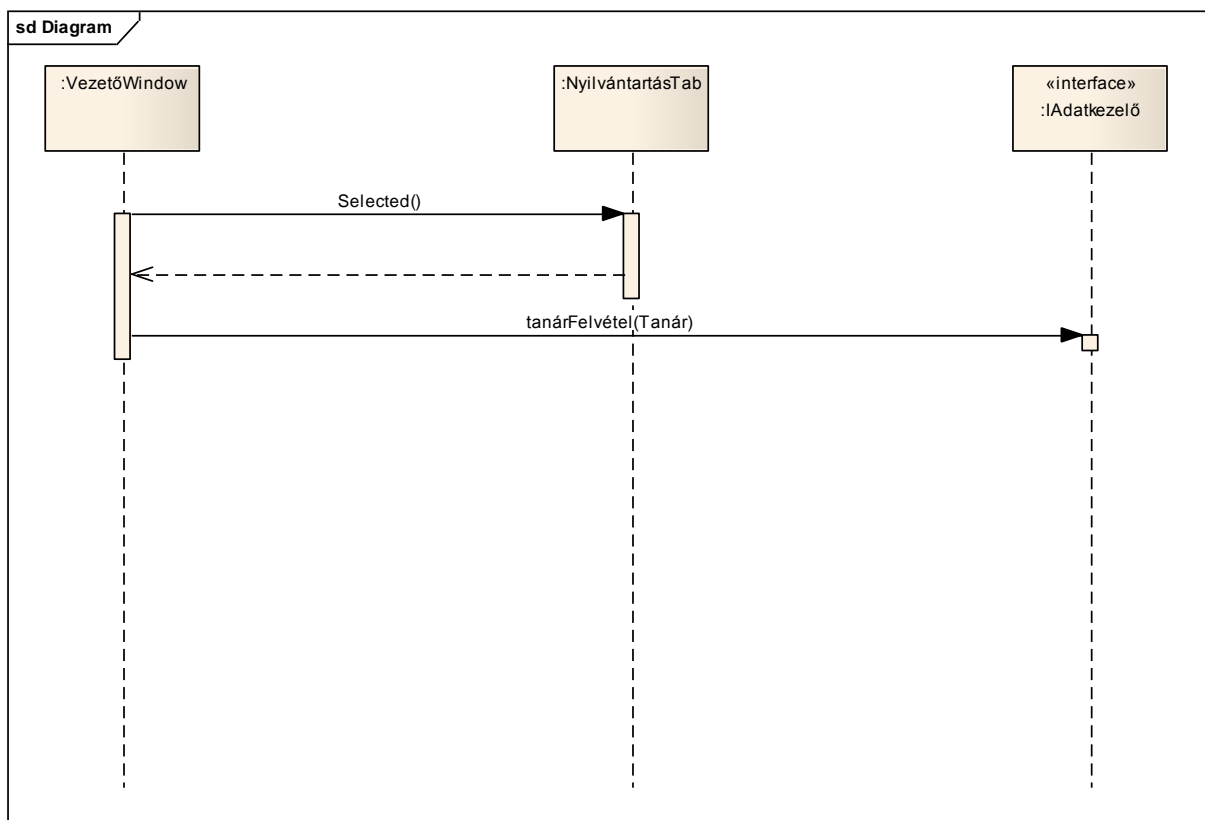


Az adatkezelő osztály segítségével hajtjuk végre a személyes adatokon történő változtatásokat, úgy mint létrehozás, módosítás és törlés.

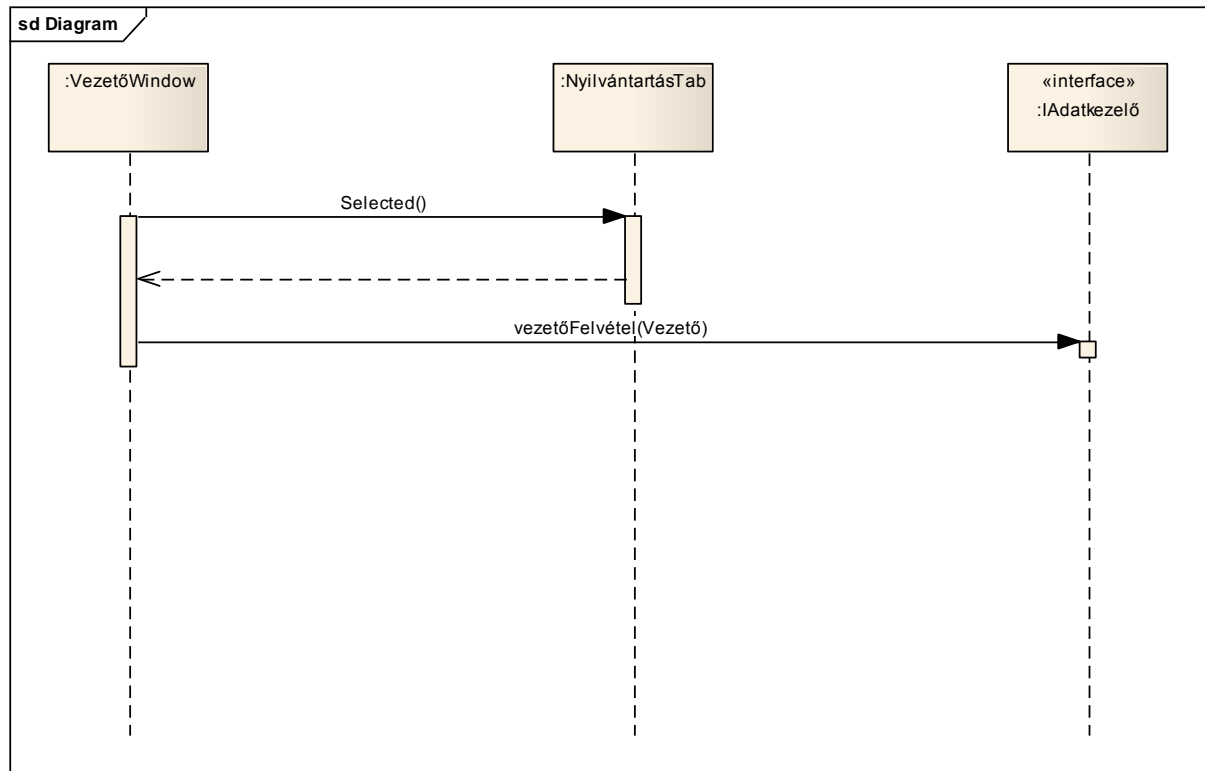
A `HallgatóFelvétel()`, `tanárFelvétel()` és `vezetőFelvétel()` metódusokat csak vezető jogkörű felhasználó használhatja, segítségükkel új felhasználó vehető fel a rendszerbe.



Hallgató felvétele (Vezető)

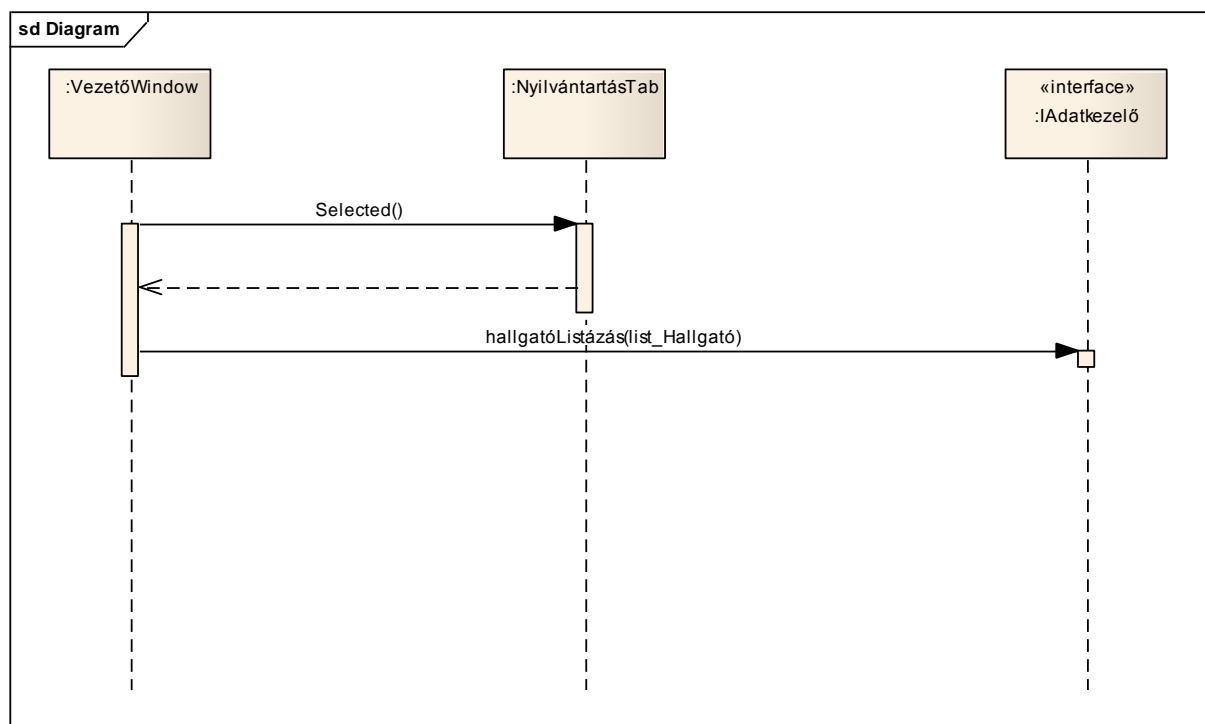


Tanár felvétele (Vezető)

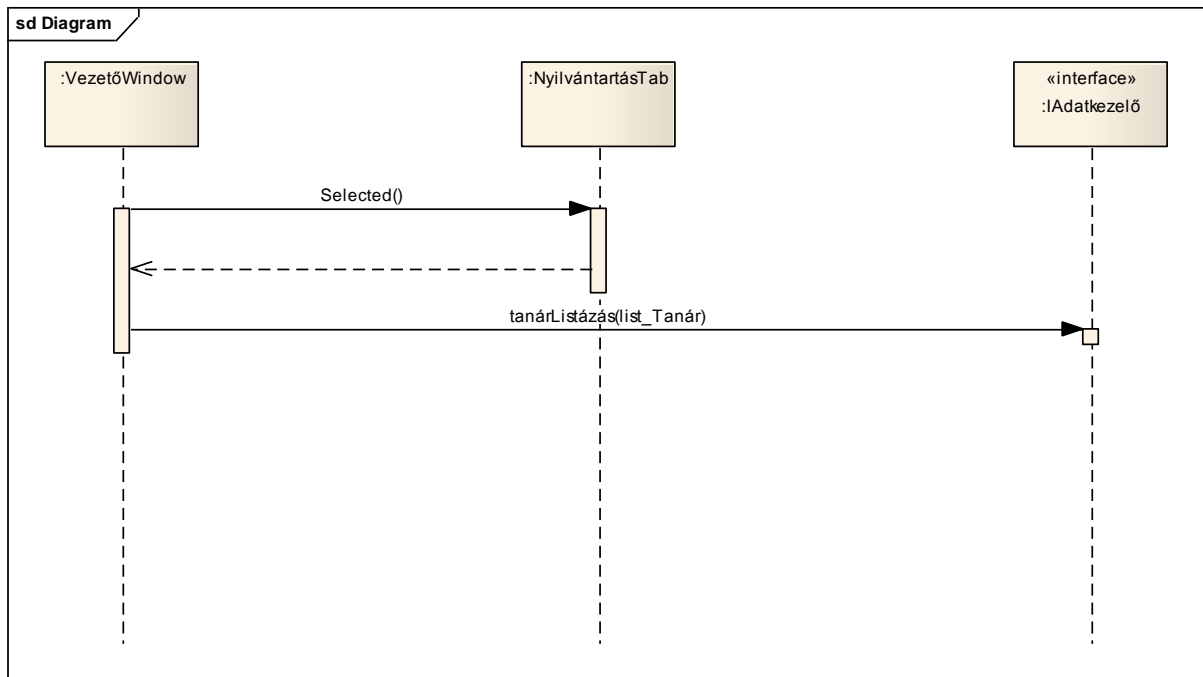


Vezető felvétele (Vezető)

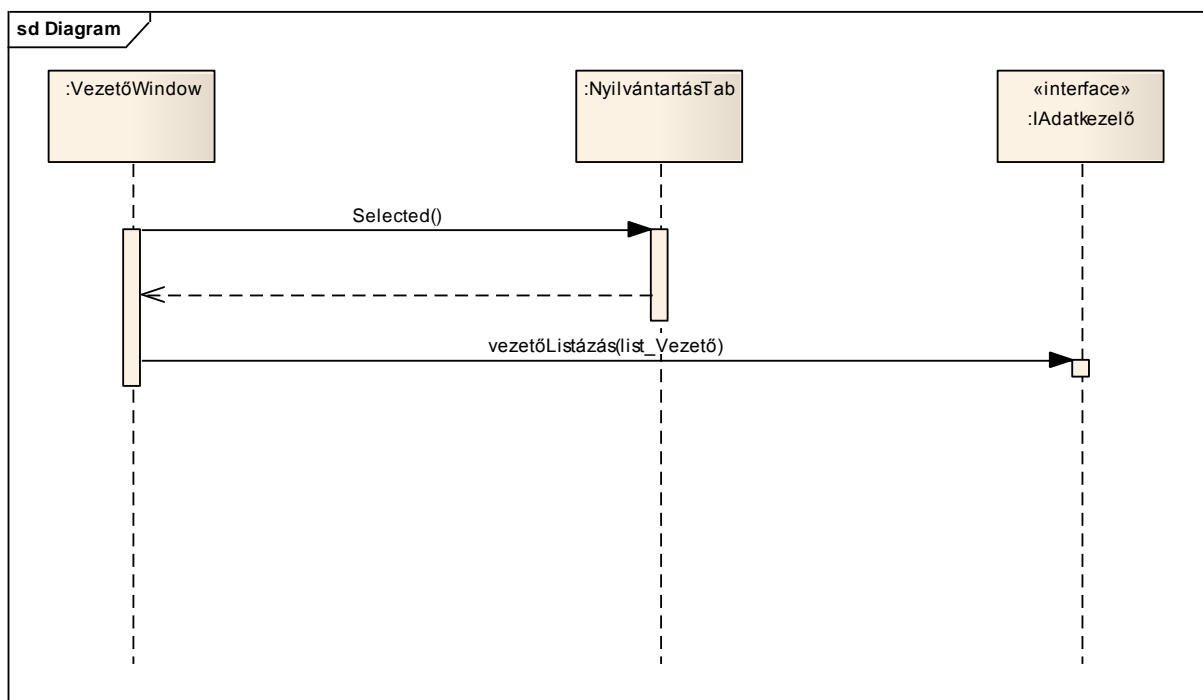
A könnyebb átláthatóság miatt szétbontottuk jogkörök szerint a teljes nyilvántartás listázását, ezért lett **vezetőListázás()**, **tanárListázás()** és **hallgatóListázás()**, melyek vizuálisan is három külön listába töltik be a kapott végeredményt.



Hallgatók listázása (Vezető)

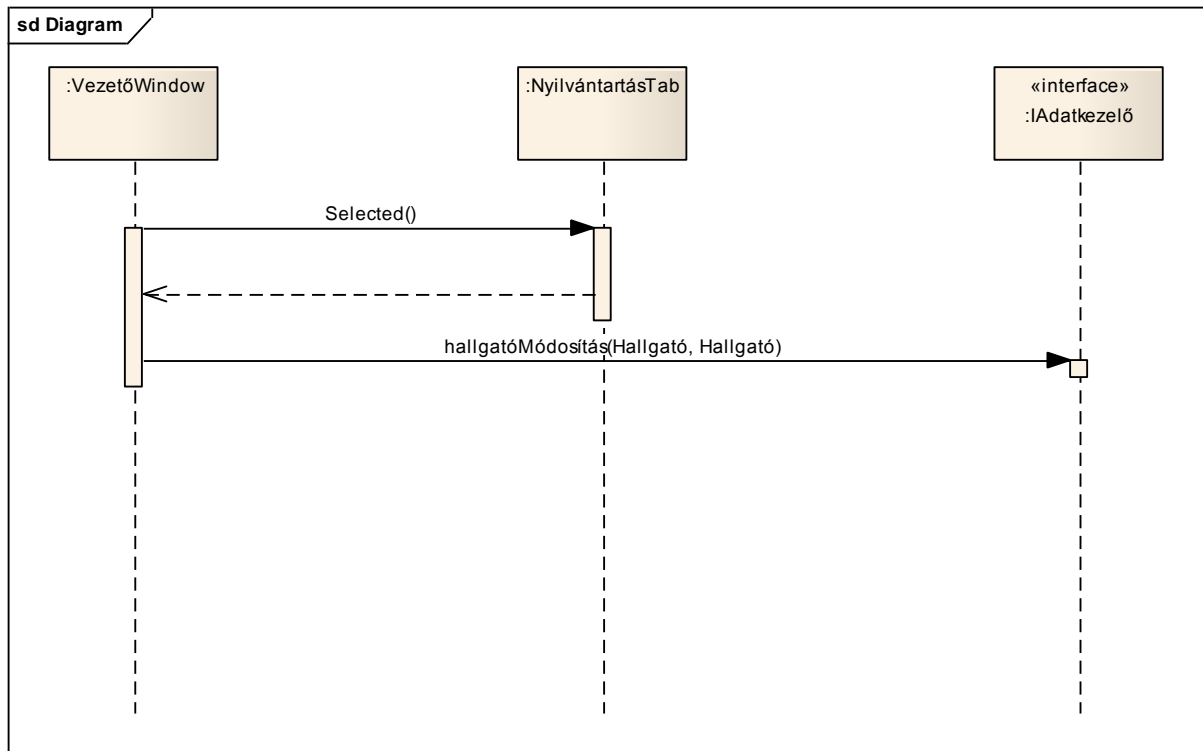


Tanár listázása (Vezető)

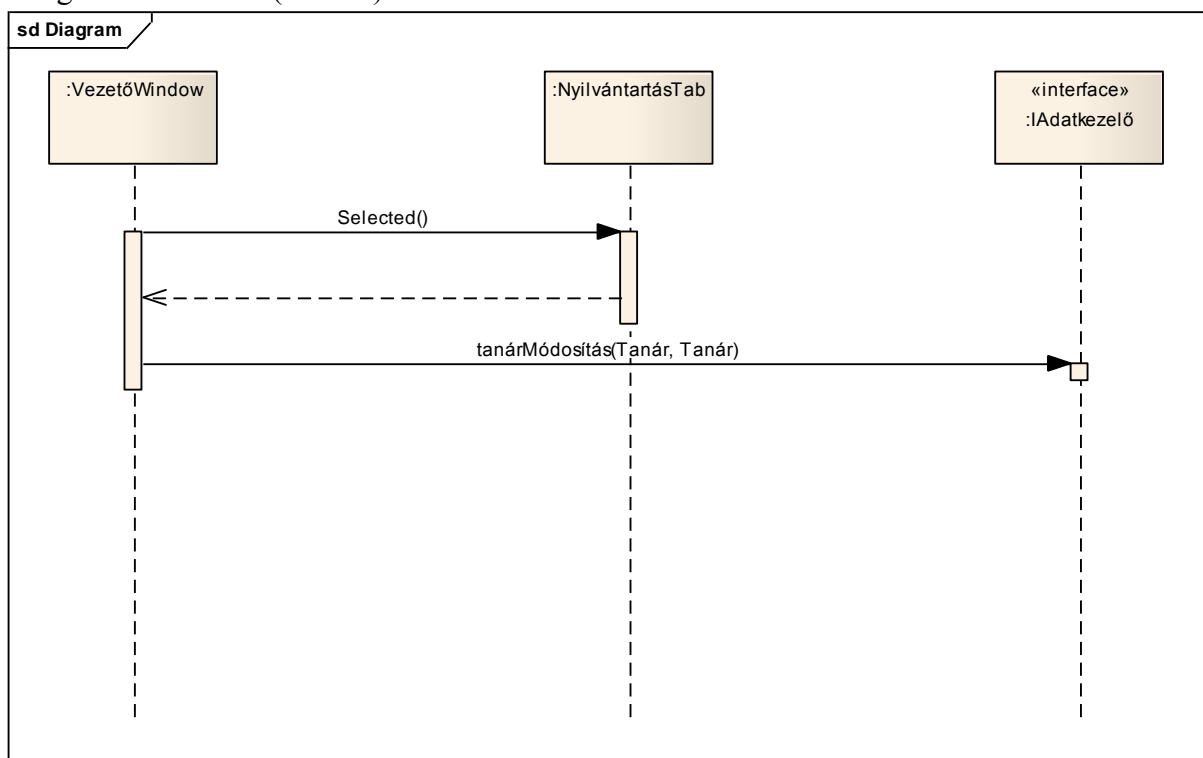


Vezető listázása (Vezető)

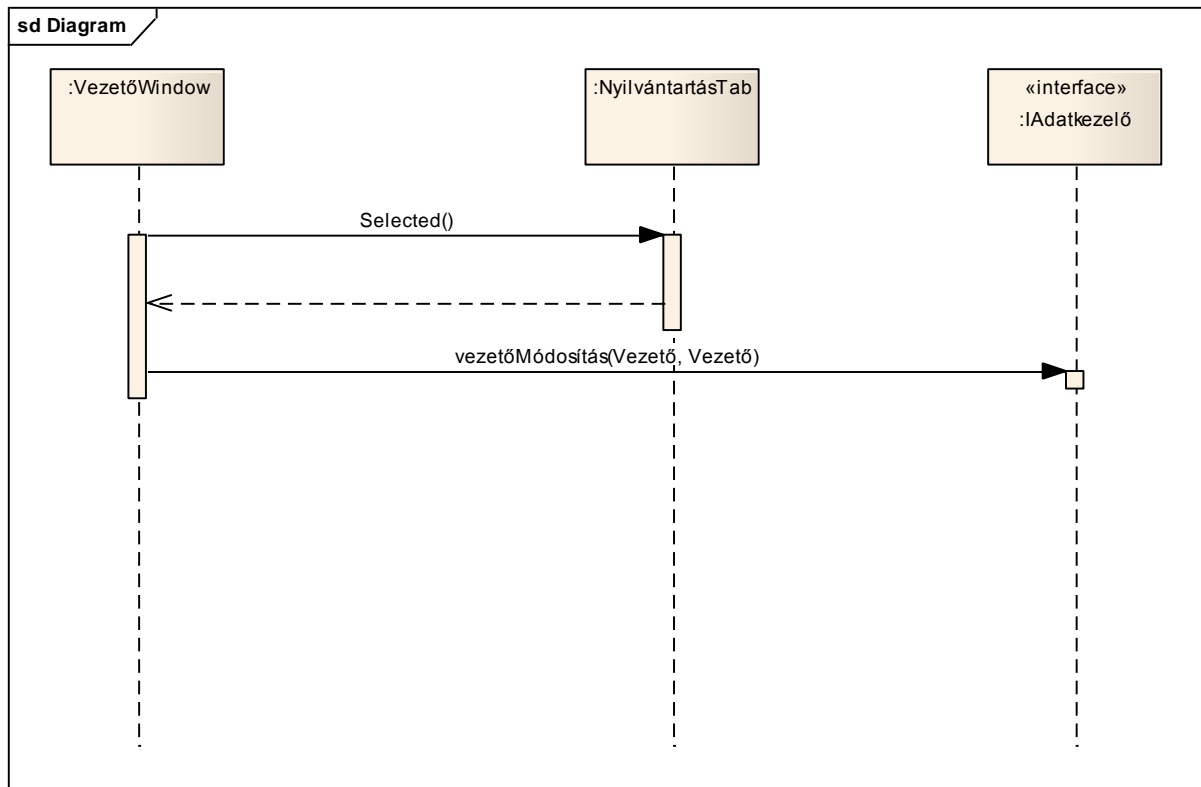
A hallgatóMódosítás(), tanárMódosítás() és vezetőMódosítás() hasonló megkötésekkel működik, mint a létrehozás, itt azonban paraméterként meg kell adni, hogy mit, mire akarunk változtatni.



Hallgató módosítása (Vezető)

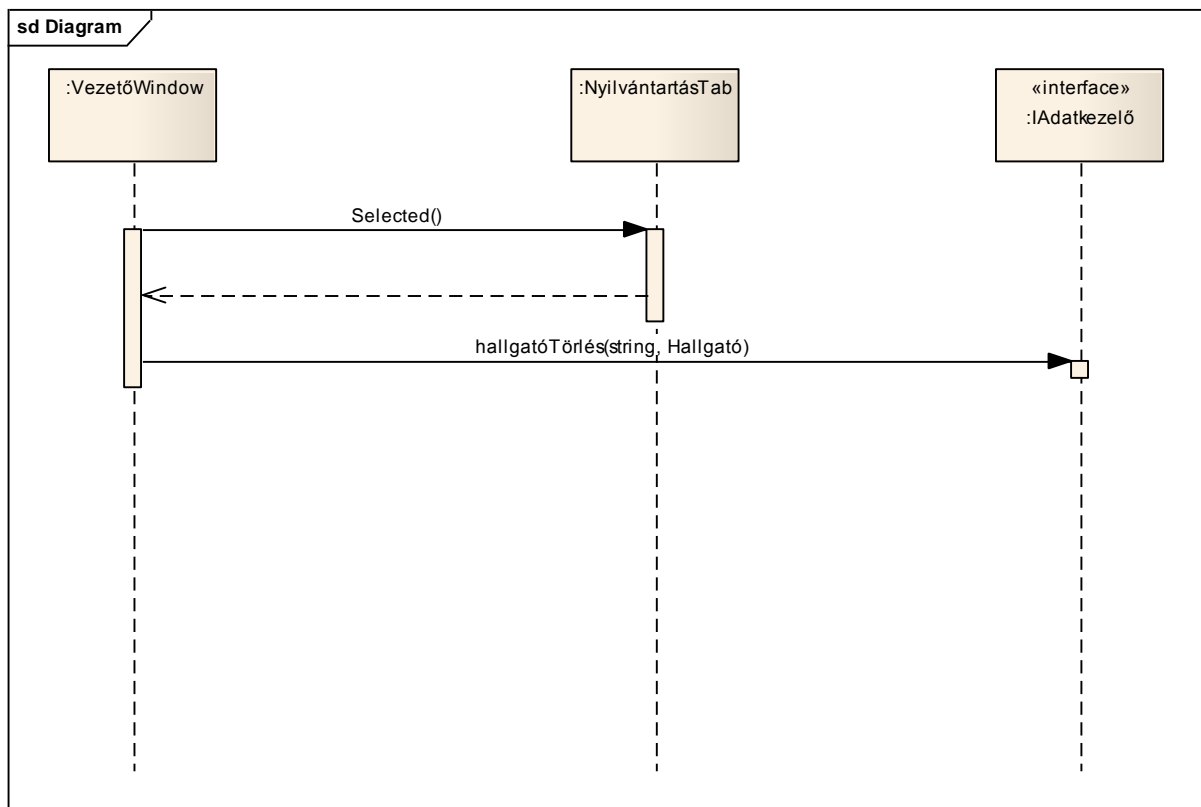


Tanár módosítása (Vezető)

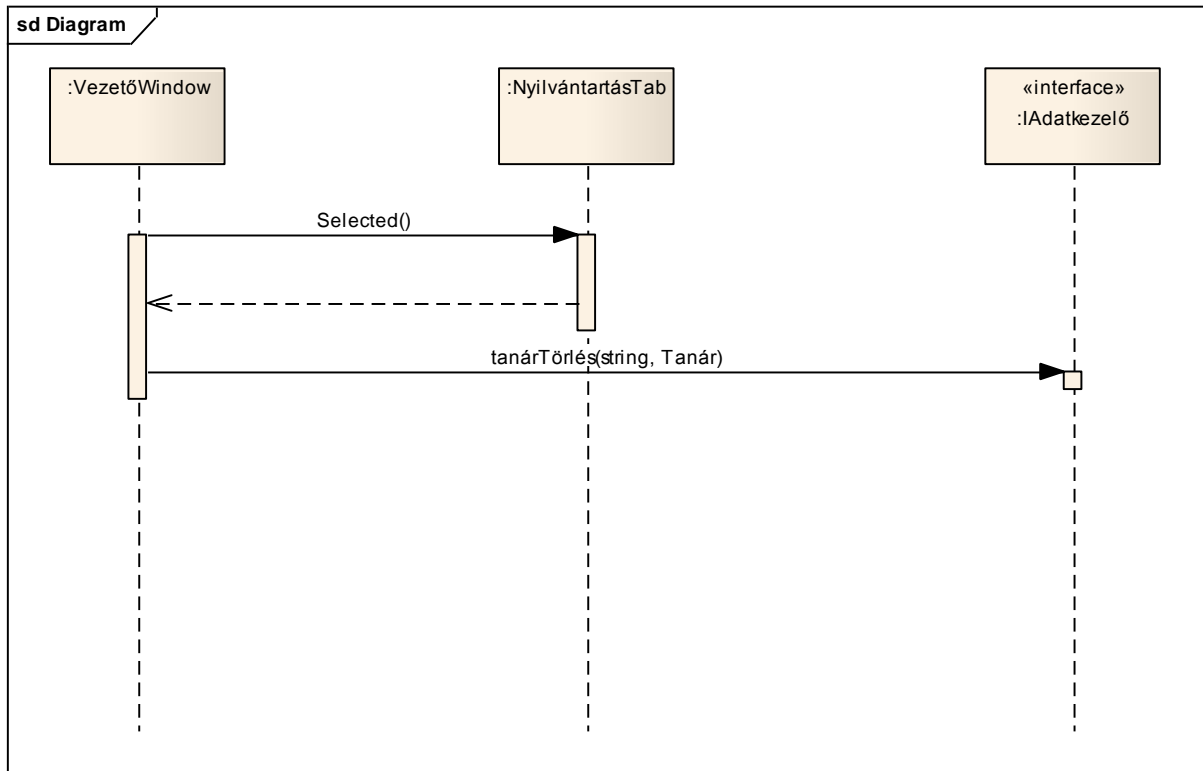


Vezető módosítása (Vezető)

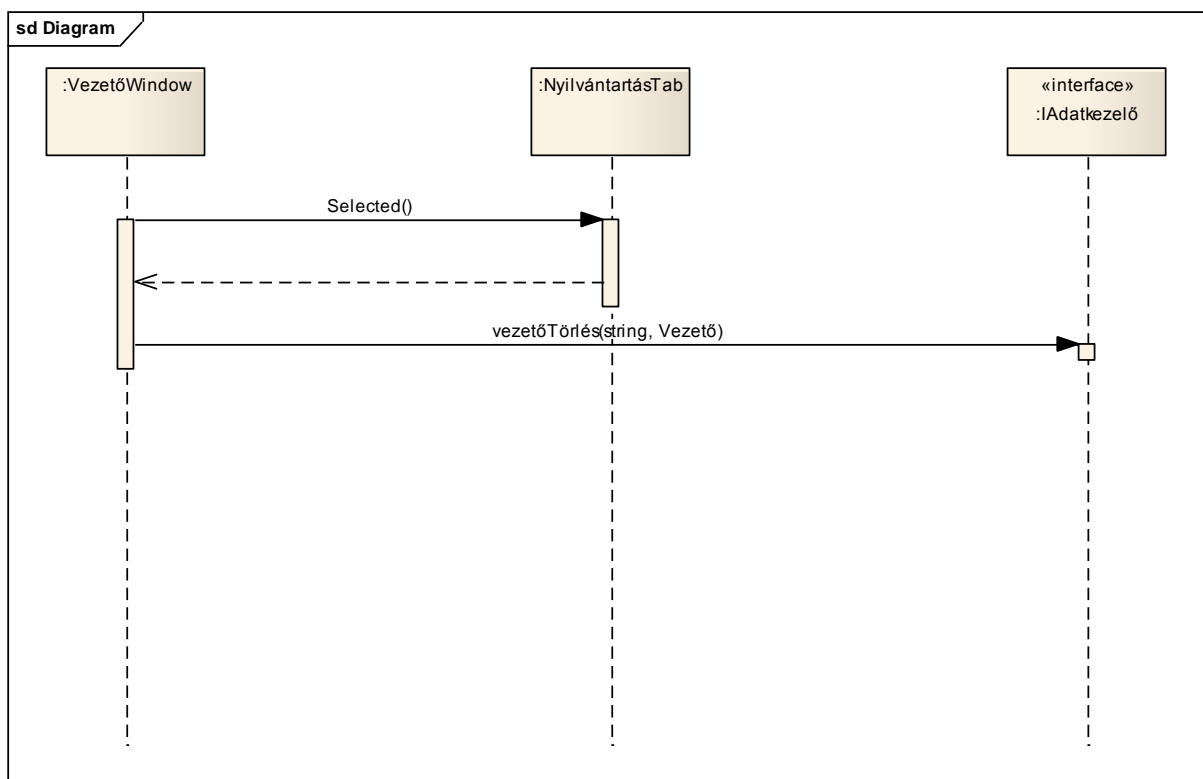
A törlés\*() is szintén vezetői jogkört igényelnek, ezek a metódusok fizikailag nem törlik a felhasználókat, csupán archiválják őket.



Hallgató törlése (Vezető)



Tanár törlése (Vezető)



Vezető törlése (Vezető)

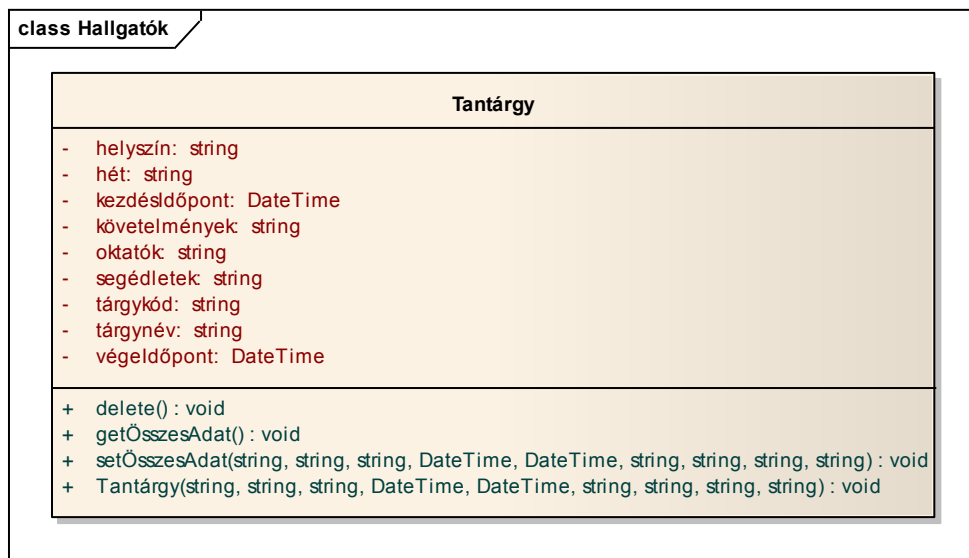


Hallgató osztály:



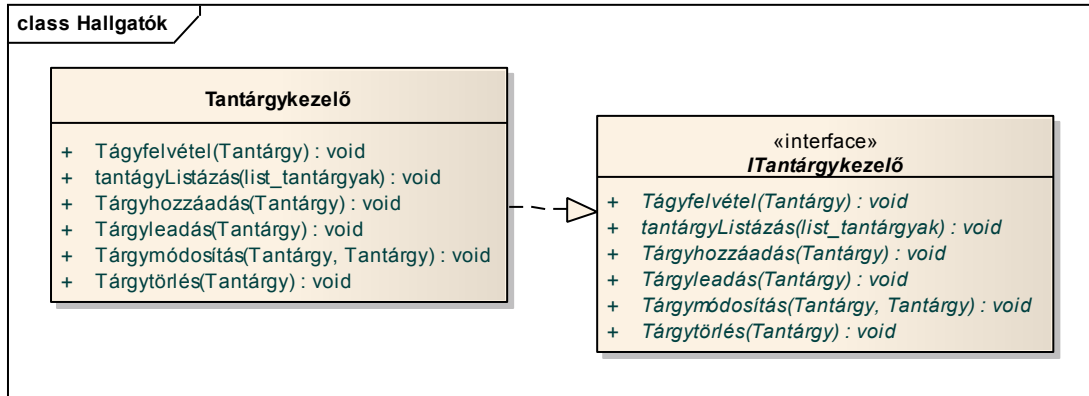
A hallgató osztály objektumai tárolják a hallgatók felvett tárgyait, tanulmányi státuszukat, öröklök a Személy osztály attribútumait és kibővítik egy aktív-státusz figyelő értékkel, a felvett tárgyak listájával és az iskola elvégzésére utaló értékkel.

Tantárgy osztály:



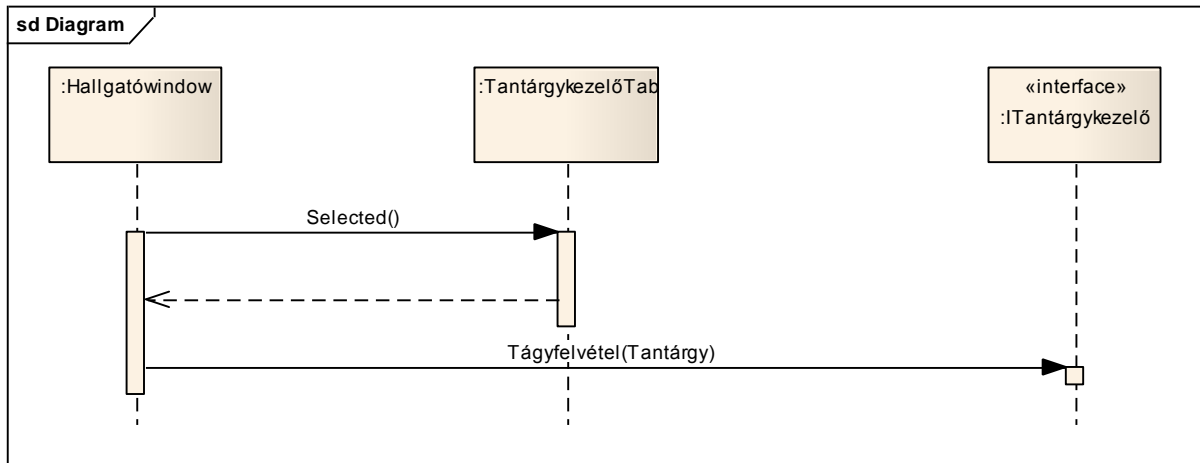
Az osztály objektumai segítségével tárolhatjuk el az iskola tantárgyait, listázhatjuk őket és módosíthatjuk is a tartalmukat.

Tantárgykezelő osztály és ITantárgykezelő interface:



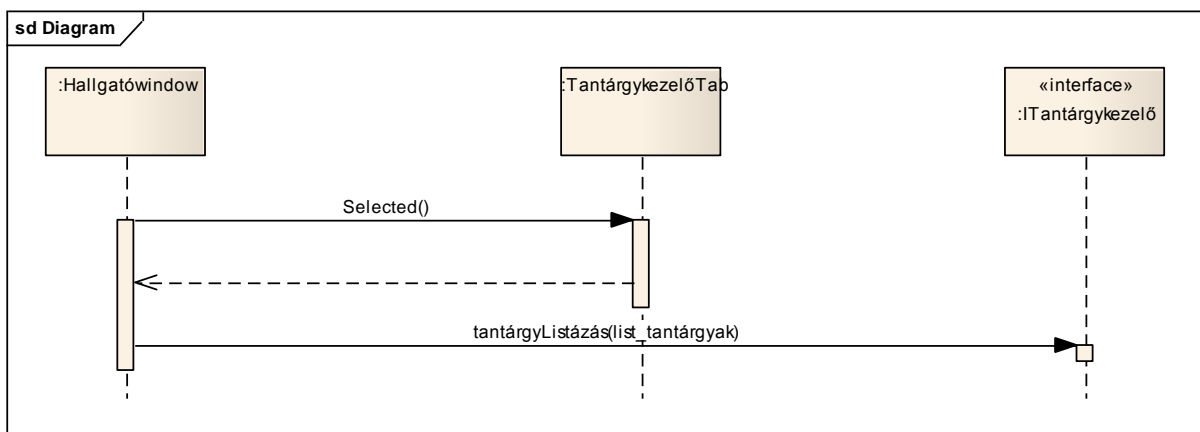
A kezelőosztályon keresztül hajthatjuk végre a tárgyfelvételt, a tárgyak listázását, az új tárgy hozzáadását, a tárgyleadást, a tárgymódosítást és a törlést.

A Tanárgyfelvétel() a paraméterként kapott tárgyat próbálja meg regisztrálni az aktuális hallgatóhoz.



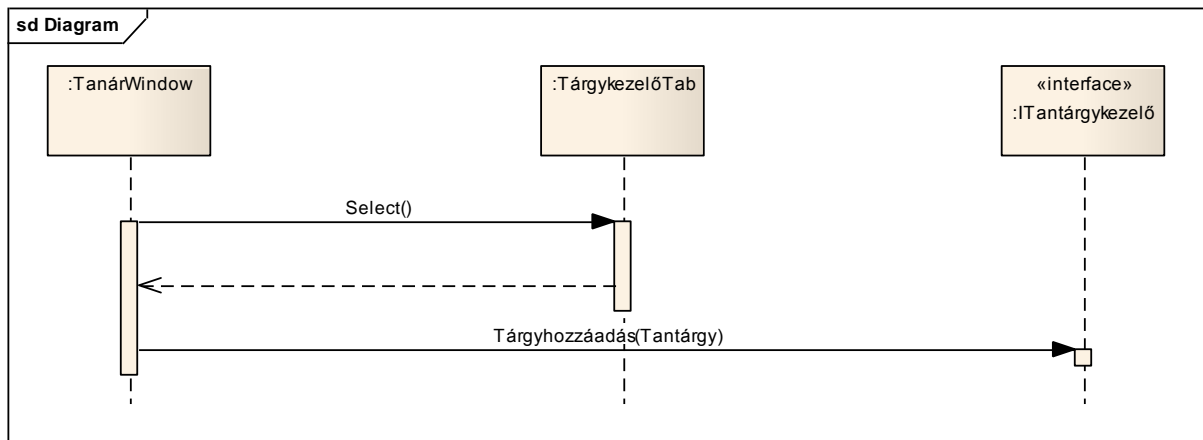
Tárgyfelvétel (Hallgató)

A TantárgyListázás() visszaadja a paraméterként kapott Tantárgy típusú lista elemeinek részleteit.



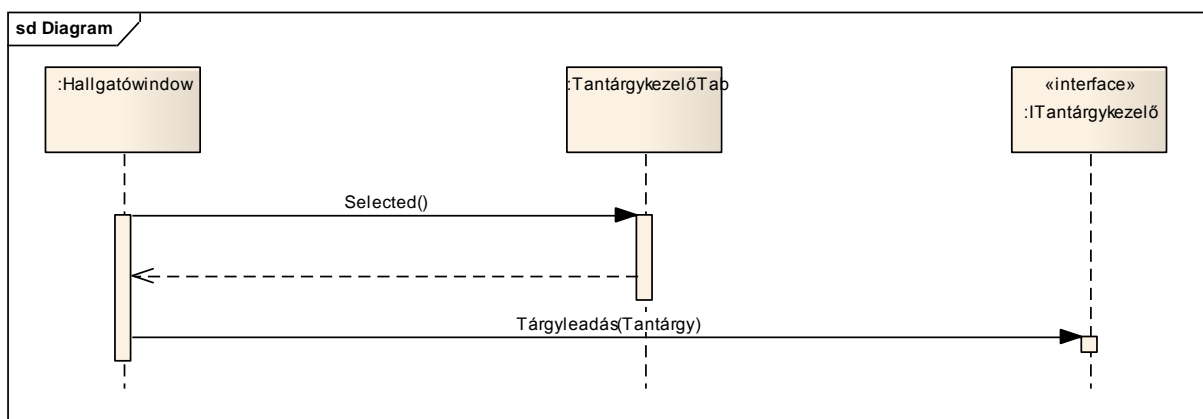
Tantárgy listázása (Hallgató)

A Tárgyhozzáadás() művelet a paraméterként kapott új tárgyat próbálja meg hozzáadni a rendszerhez.



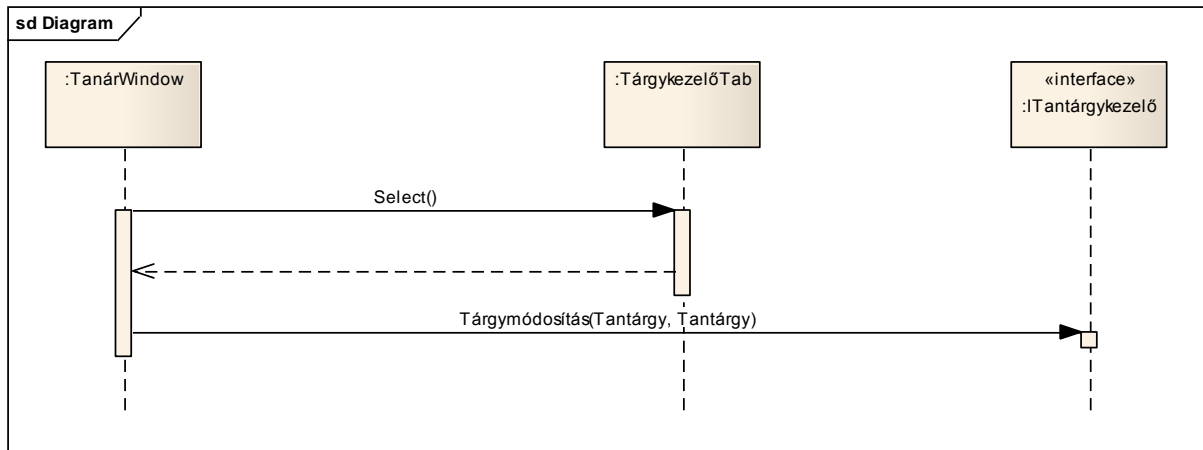
Tárgy hozzáadása (Tanár)

A Tárgyleadás() folyamat bontja a kapcsolatot a tantárgy és a hallgató között, lejelentkezünk róla.



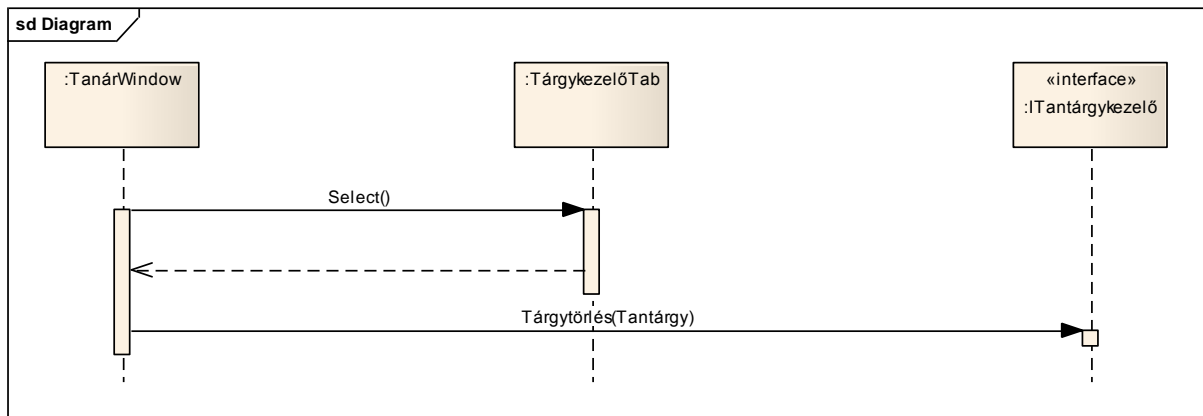
Tárgy leadása (Hallgató)

A Tárgymódosítás() egy tanári funkció, segítségével a már meglévő tárgyak adatait szerkeszthetjük.



Tárgy módosítása (Tanár)

A Tárgytörlés() metódus a már meglévő tárgyakat képes törölni, hallgatónál nem elérhető ilyen funkció.



Tárgy törlése (Tanár)

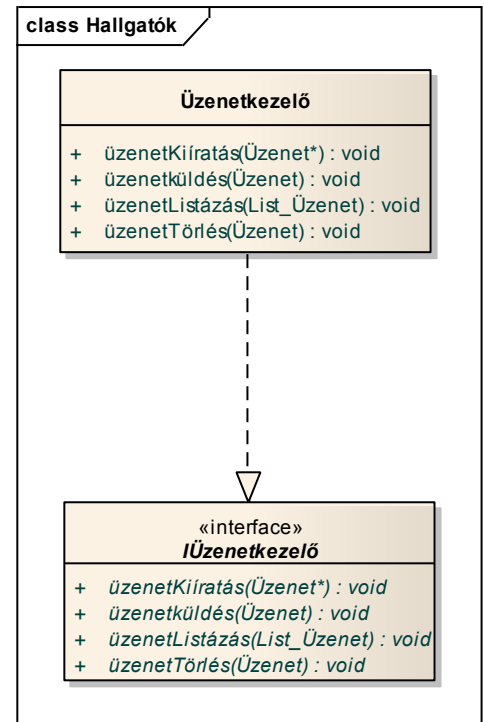
Üzenet osztály:



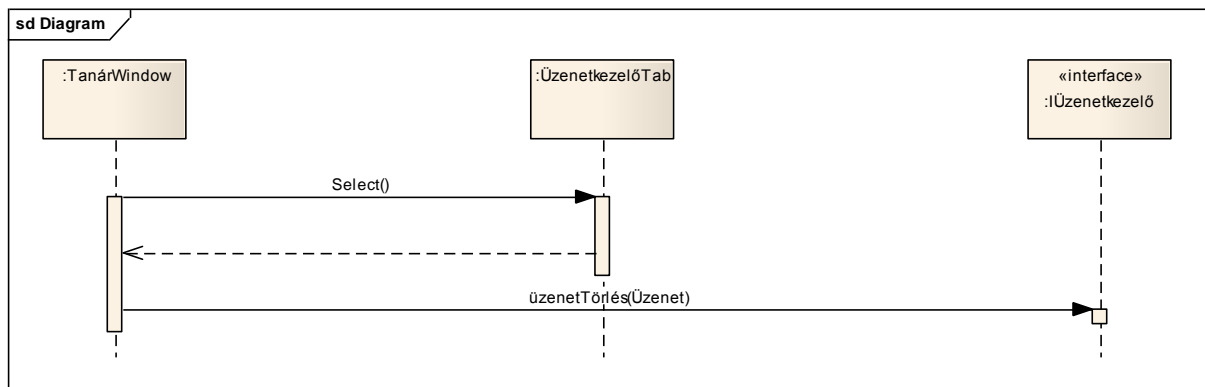
Az üzenet osztály segítségével tároljuk a felhasználók egymás közt küldött üzeneteit. Lekérdezhető a számosságuk, törölhetőek és kezelik az olvasatlan-olvasott státuszt.

Üzenetkezelő és IÜzenetkezelő interface:

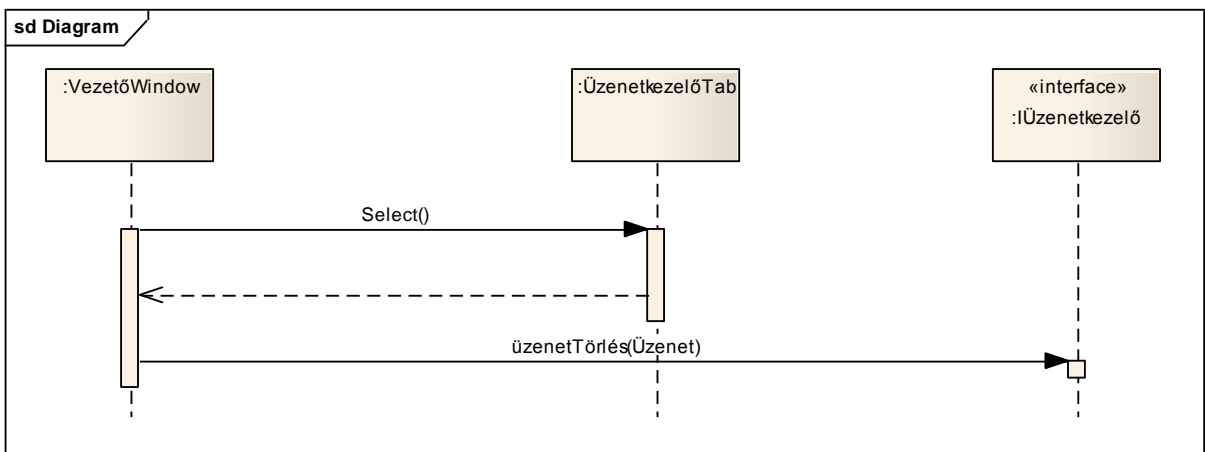
Az Üzenetkezelővel valósíthatjuk meg az ÜzenetKíratást(), az ÜzenetKüldést(), az ÜzenetListázást() és az ÜzenetTörlést().



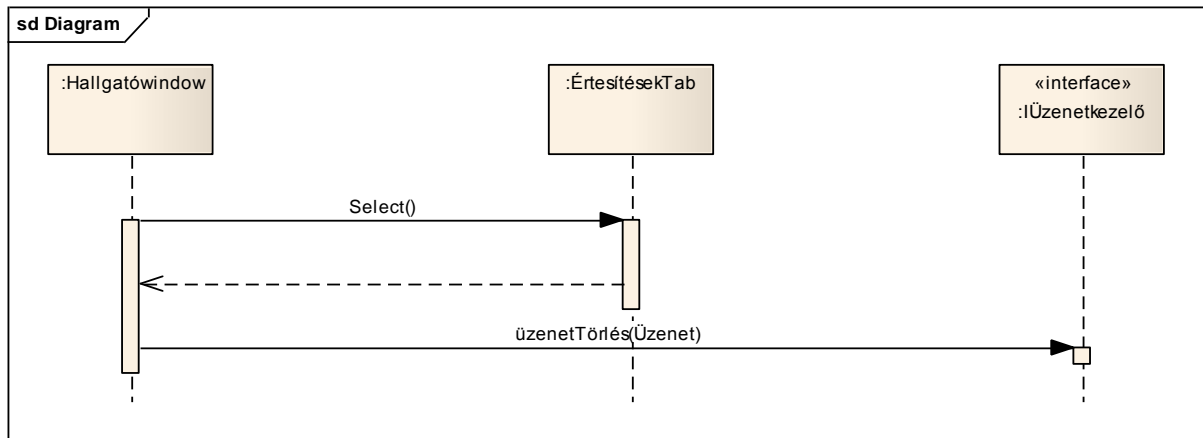
Az ÜzenetTörlés() interfészműveletnek paraméterként egy üzenetet megadva törölni tudjuk a felhasználói listából. A tanári és vezetői megvalósítás a feladatok sorrendiségét tekintve megegyezik.



Üzenet törlés (Tanár)

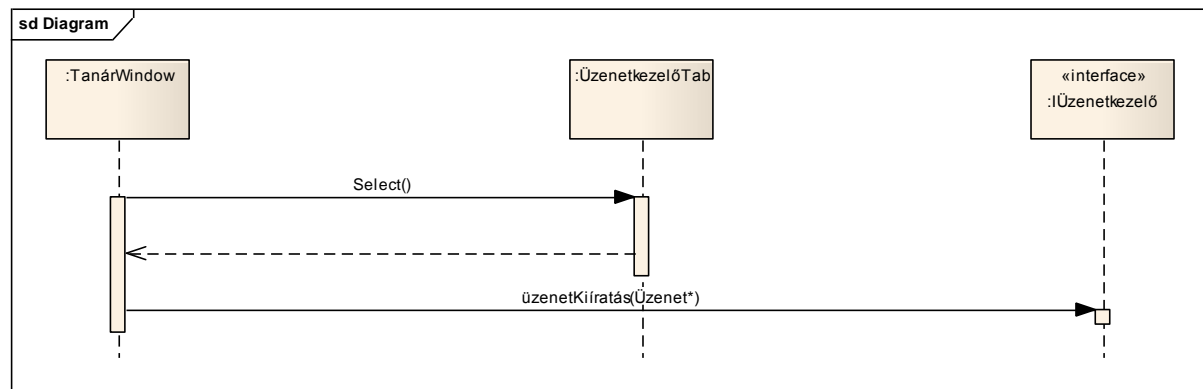


Üzenet törlés (Vezető)

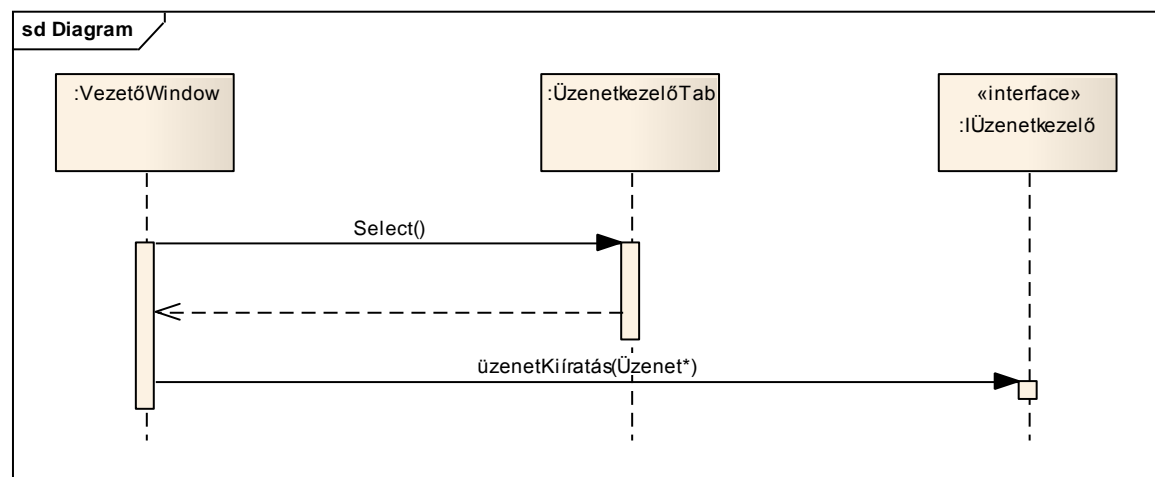


Üzenet Törlés (hallgató)

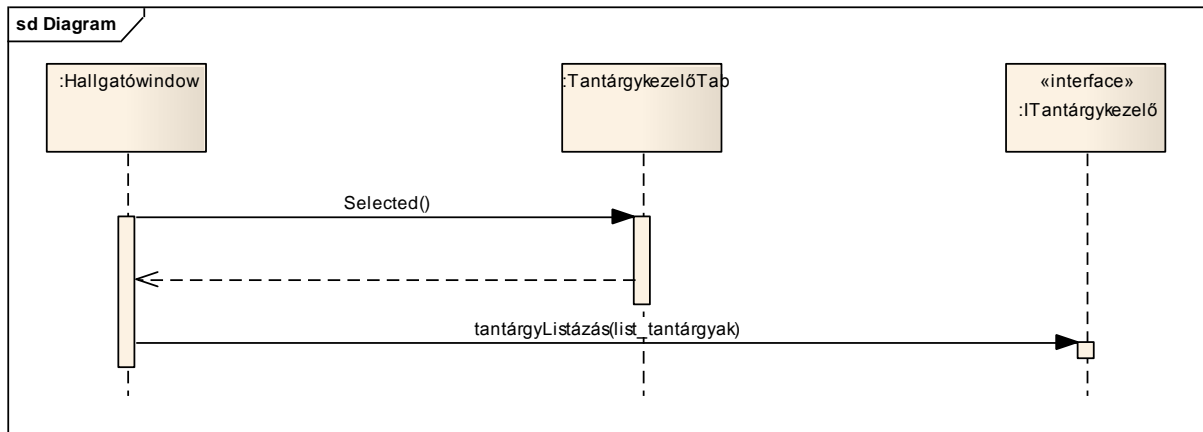
Az `ÜzenetKiíratás()` metódus a kijelölt üzeneteket nyitja meg egy felugró ablakban. Szekvenciális megvalósítása tanárnál és vezetőnél megegyeznek.



Üzenet kiíratás (Tanár)

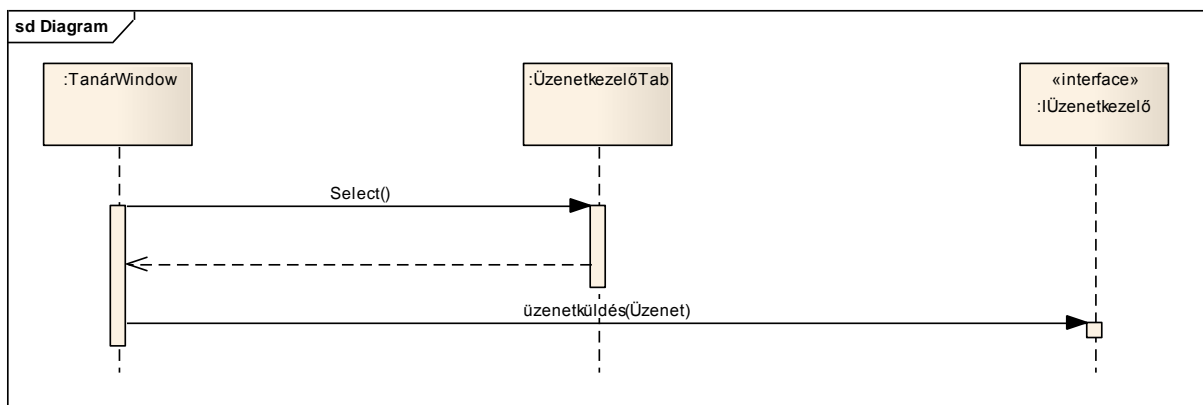


Üzenet kiíratása (Vezető)

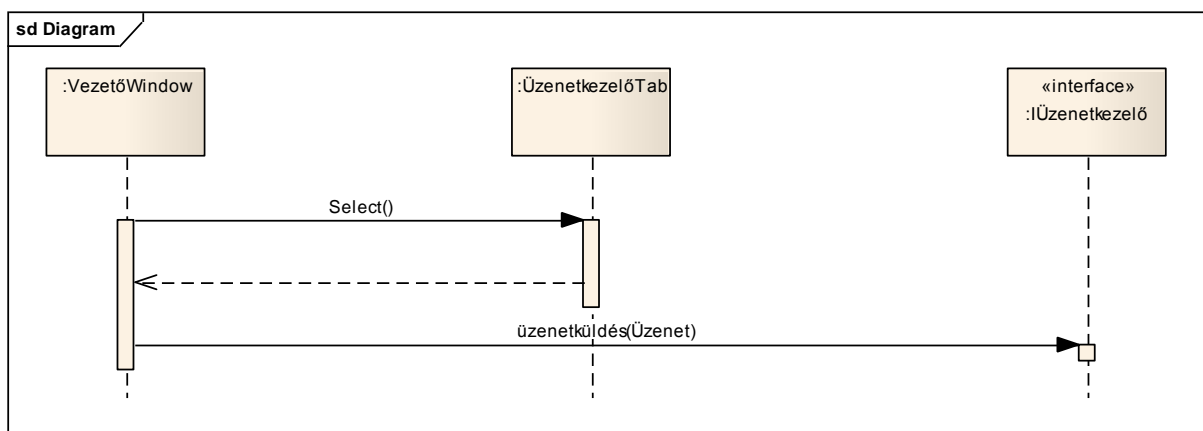


Üzenet kiíratás (Hallgató)

Az ÜzenetKüldés()-nek egy Üzenet típusú paramétert átadva tudunk küldeni tényleges üzenetet más felhasználóknak. Szekvencia szinten a vezető és tanár üzenetküldése megyezik, a hallgató nem tud üzenetet küldeni, csak fogadni.

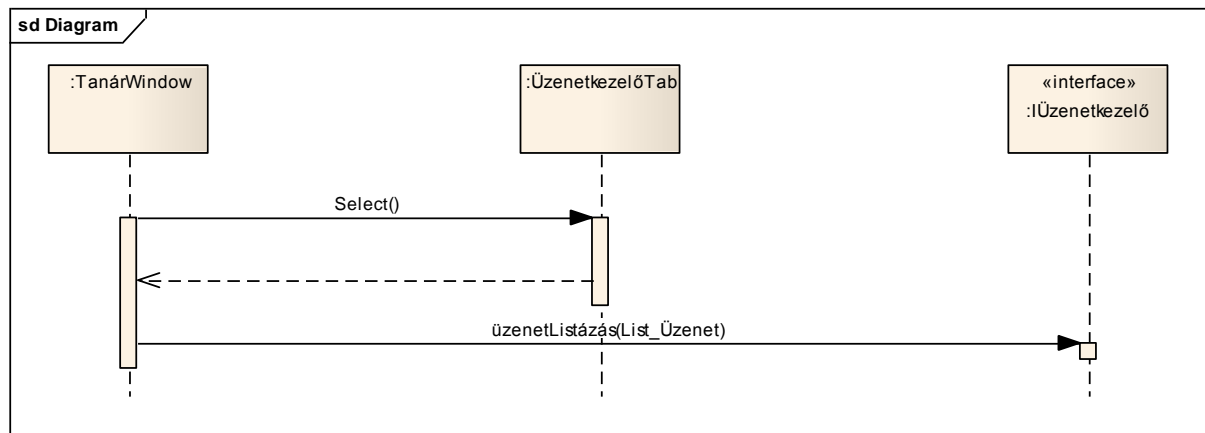


Üzenet küldés (Tanár)

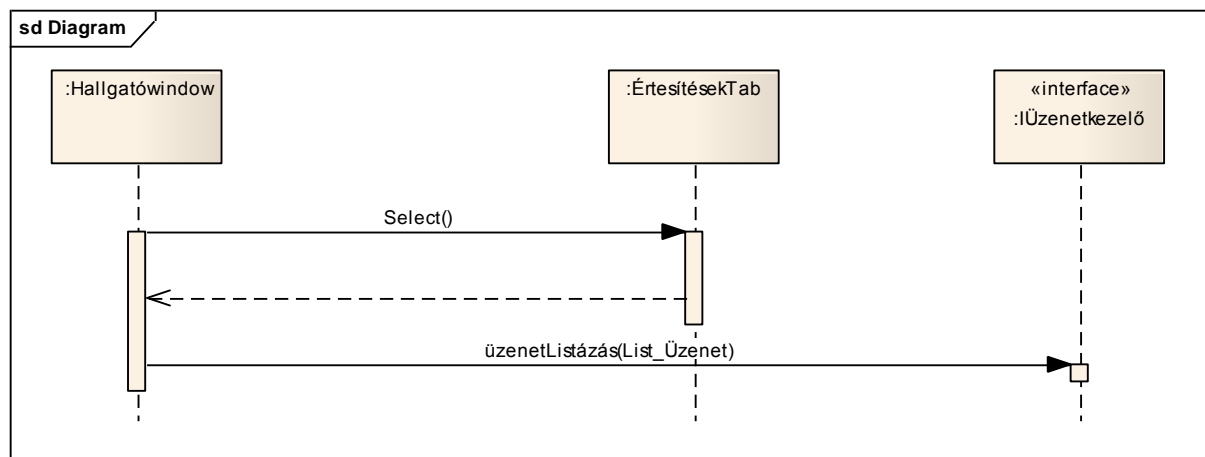


Üzenet küldés (Vezető)

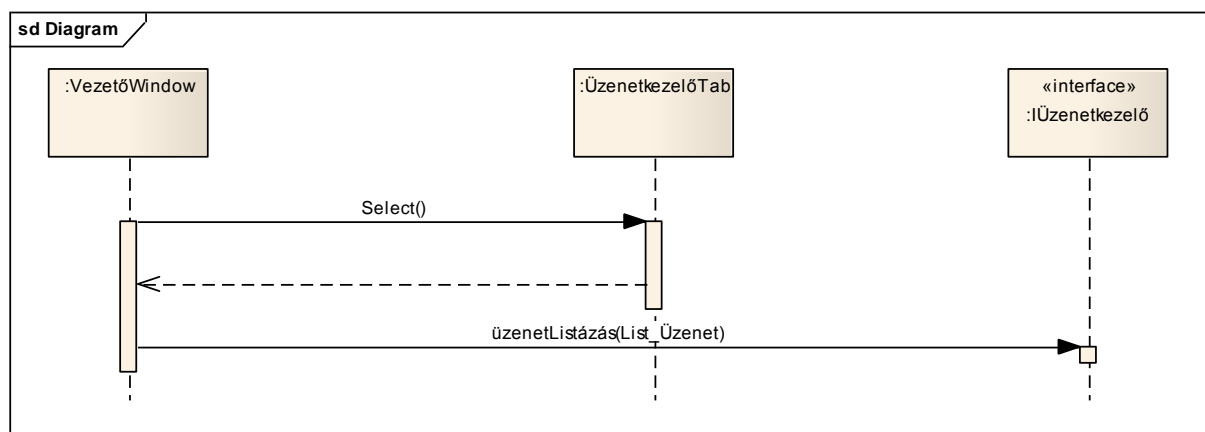
Az ÜzenetListázás() kilistázza a felhasználónak a hozzá tartozó üzeneteket.



Üzenet listázás (Tanár)



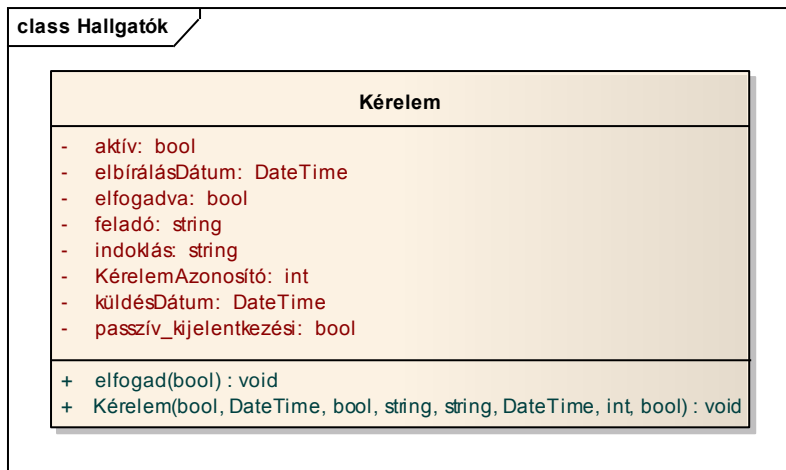
Üzenet listázás (Hallgató)



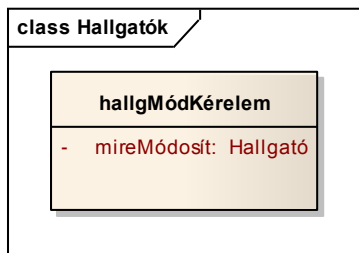
Üzenet listázás (Vezető)



Kérelem osztály:

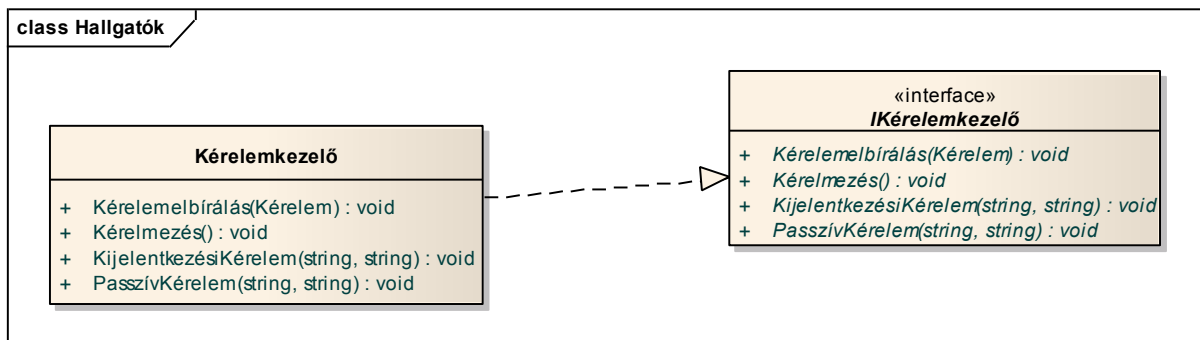


A kérelem osztály a vezetés felé küldhető, döntést igénylő műveletek kérelmeinek a tárolására szolgál.



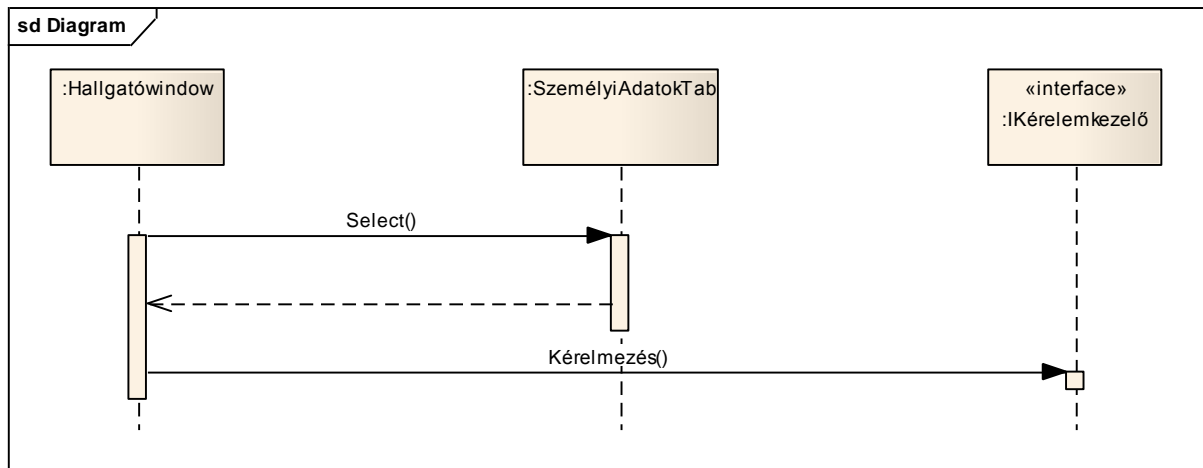
A hallgMódKérelem a Kérelem osztály leszármazottja, segítségével kérhetik a hallgatók saját adataik módosítását.

Kérelemkezelő osztály és IKérelemkezelő interfész

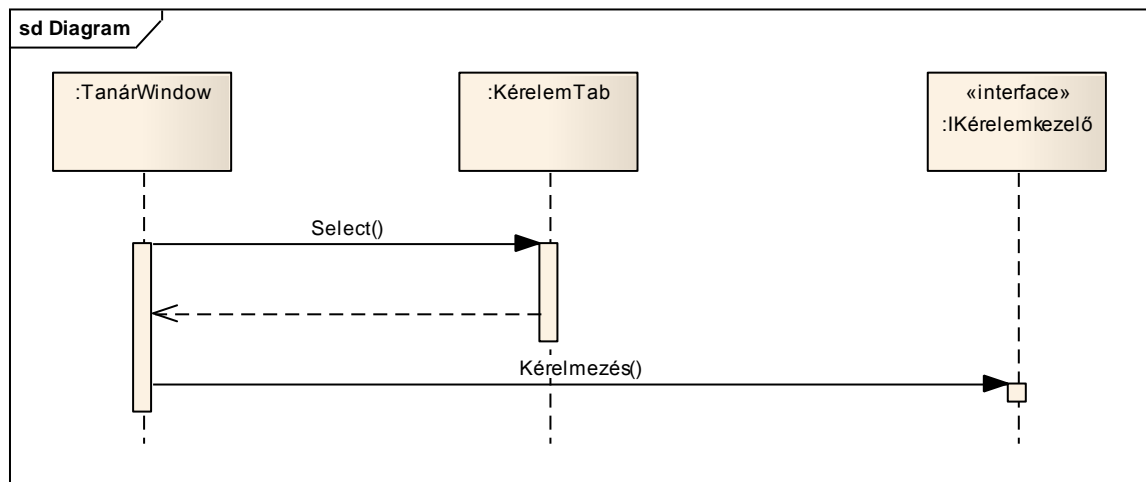


A Kerelemkezelő osztállyal valósíthatjuk meg a Kerelemelbírálás() folyamatát, a kérelmezés benyújtását és hallgatói oldalról a kijelentkezési vagy passzív kérelem kezelését.

A Kérelmezés() indításakor a tanár vagy a hallgató egy olyan folyamatot indít amihez vezetői jóváhagyás szükséges.

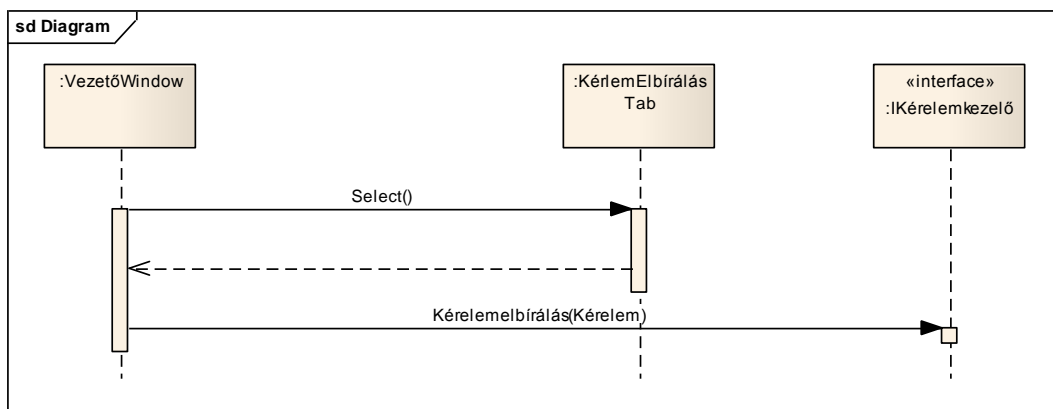


Kérelmezés(Hallgató)



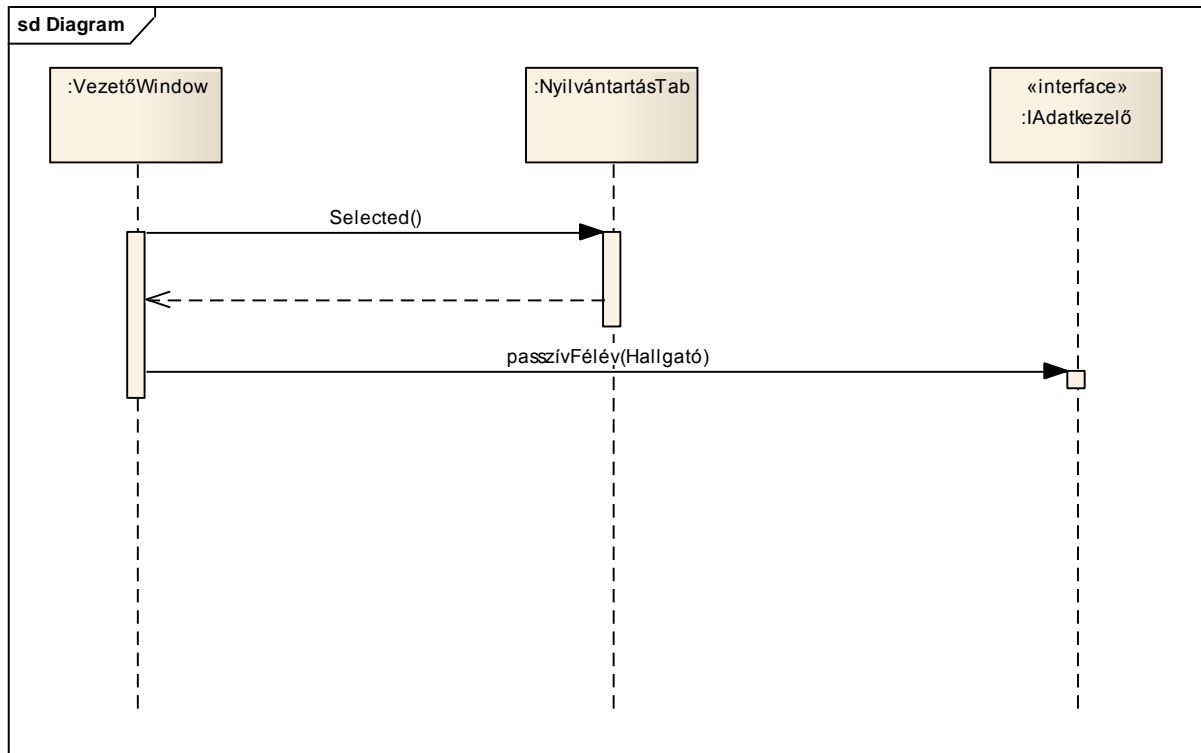
Kérelmezés (Tanár)

A Kérelmelbírálás() művelet a vezetőnél használható, döntés születik, hogy jóváhagyjuk vagy elutasítjuk a kérelmet.



Kérelmek elbírálás (Vezető)

A PasszívFélévKérelem() metódust a hallgató hívja meg, megfelelő indoklással passzív félévre szeretne menni. A folyamat szekvenciája megegyezik egy bármely más általános igénylésével, a vezetői oldalról folyamat a alábbi ábrát követi.



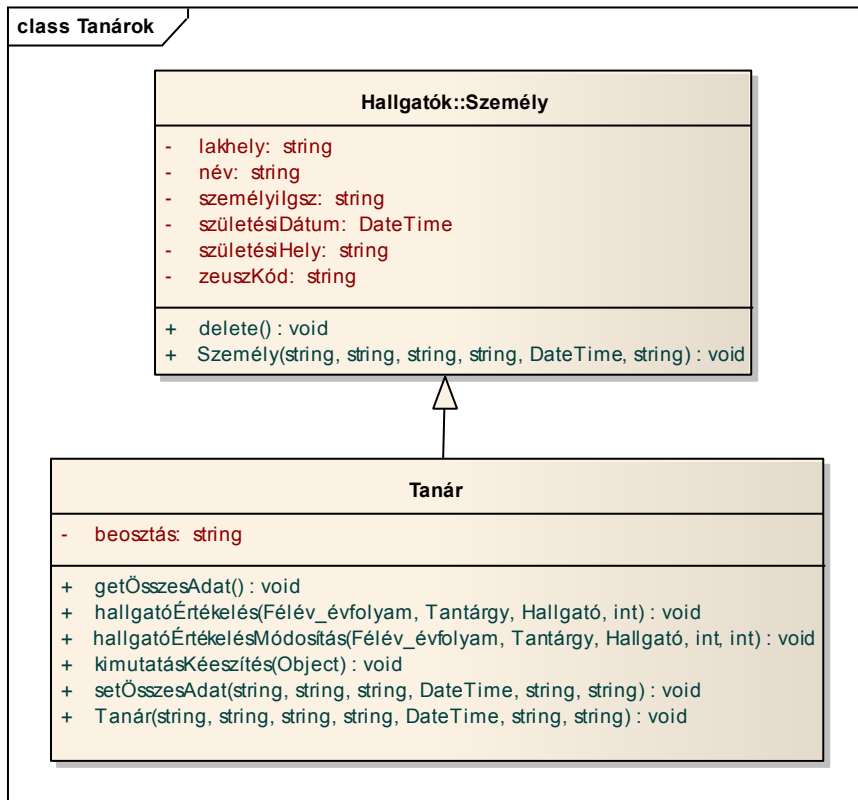
Passzív kérelem(Vezető)

KijeltekzésiKérelem()-nél a hallgató meg is szüntetné a jogviszonyt, szekvencia szinten megegyezik a PasszívKérelemmel.

### Tanár alrendszer:

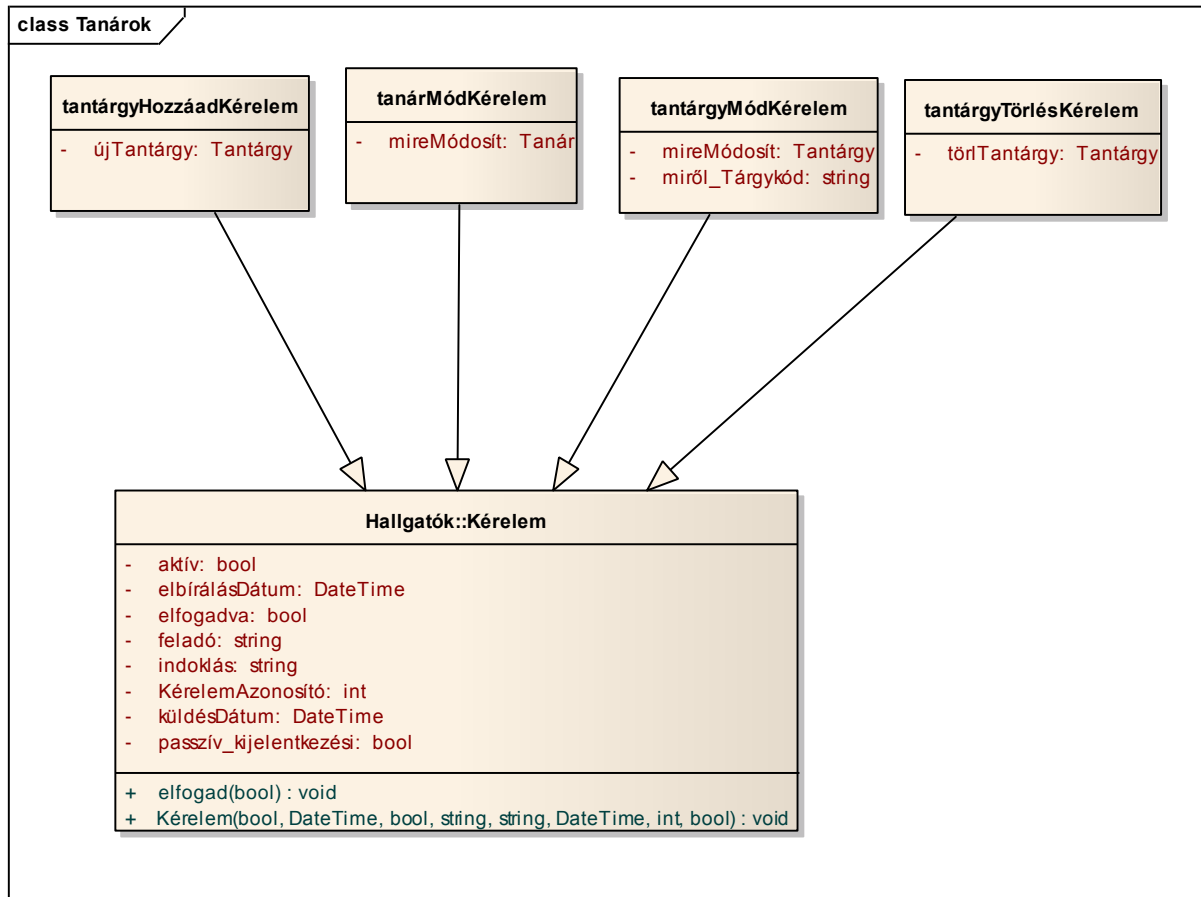
A tanárok által használt osztályok és metódusok itt vannak megvalósítva, jellemzően a Hallgató réteg osztályait egészítik ki 1-1 speciális osztállyal, metódussal.

Tanár osztály:



Szintén a Személy osztályból öröklődik, bővítve az adattagokat egy string típusú beosztás mezővel, illetve megjelenik a hallgatóÉrtékelés(), hallgatóÉrtékelésMódosítás(), kimutatásKészítés() és Összesadat() metódusok, melyek tantárgyak adminisztrálásához és statisztikák kimutatásához kellenek.

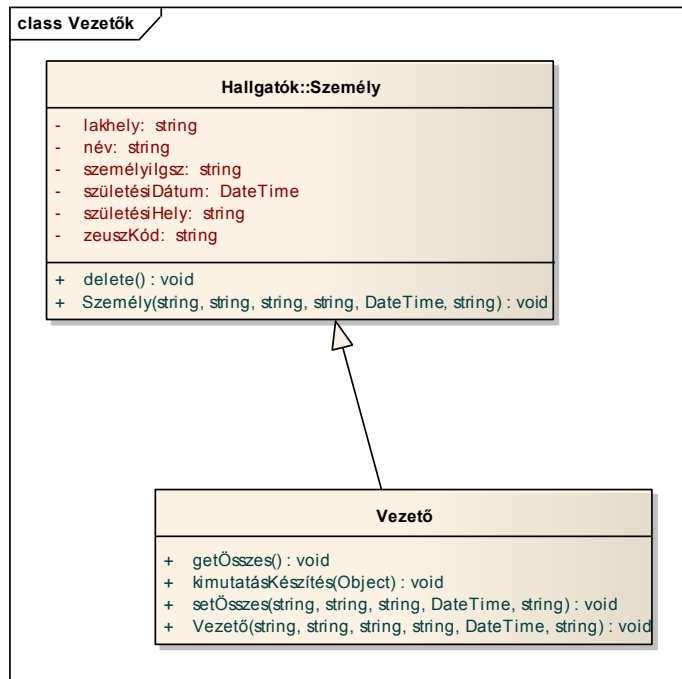
Kérelem osztály:



A kérelem osztály kiegészül Több leszármazottal, úgy mint tantárgyhozzáadKérelem(), TanárMódKérelem() melyek a tanár saját személyes adatait módosítja, tantárgyMódKérelem(), ami a tárgyak adatainak módosításához szükséges, valamint a tárgyTörlésKérelem().

## Vezető alrendszer:

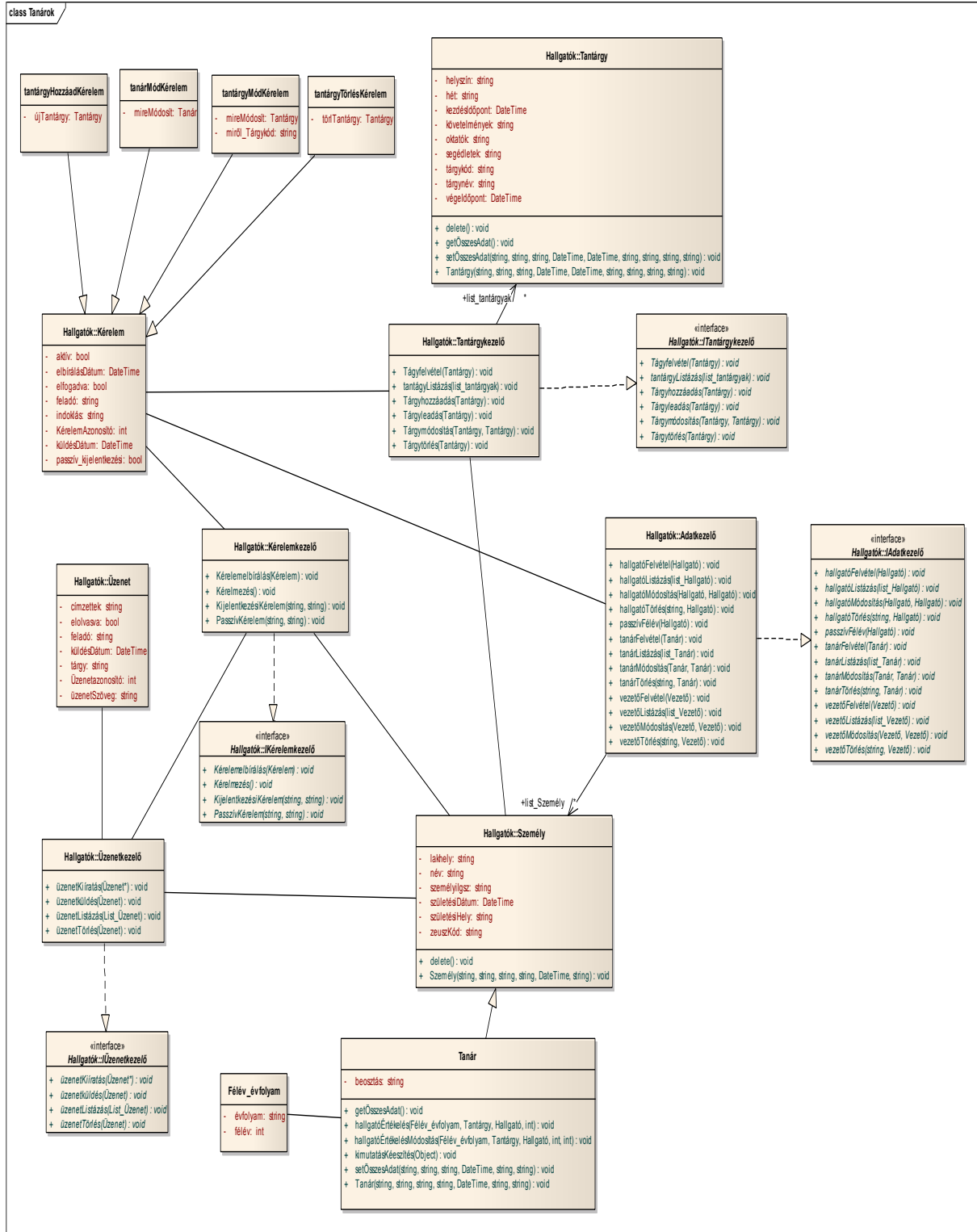
Vezető osztály:



A vezetés egyesít mindent, amit eddig a hallgatóknál és tanároknál felvettünk. A vezető osztály segítségével tároljuk el az egyes vezetőket. Metódusai a személy osztályhoz képest bővültek egy KimutatásKészítés()-sel, ami különböző szempontok szerint tudja kimutatni az iskola eddigi történetét.



Tanárok alrendszer:



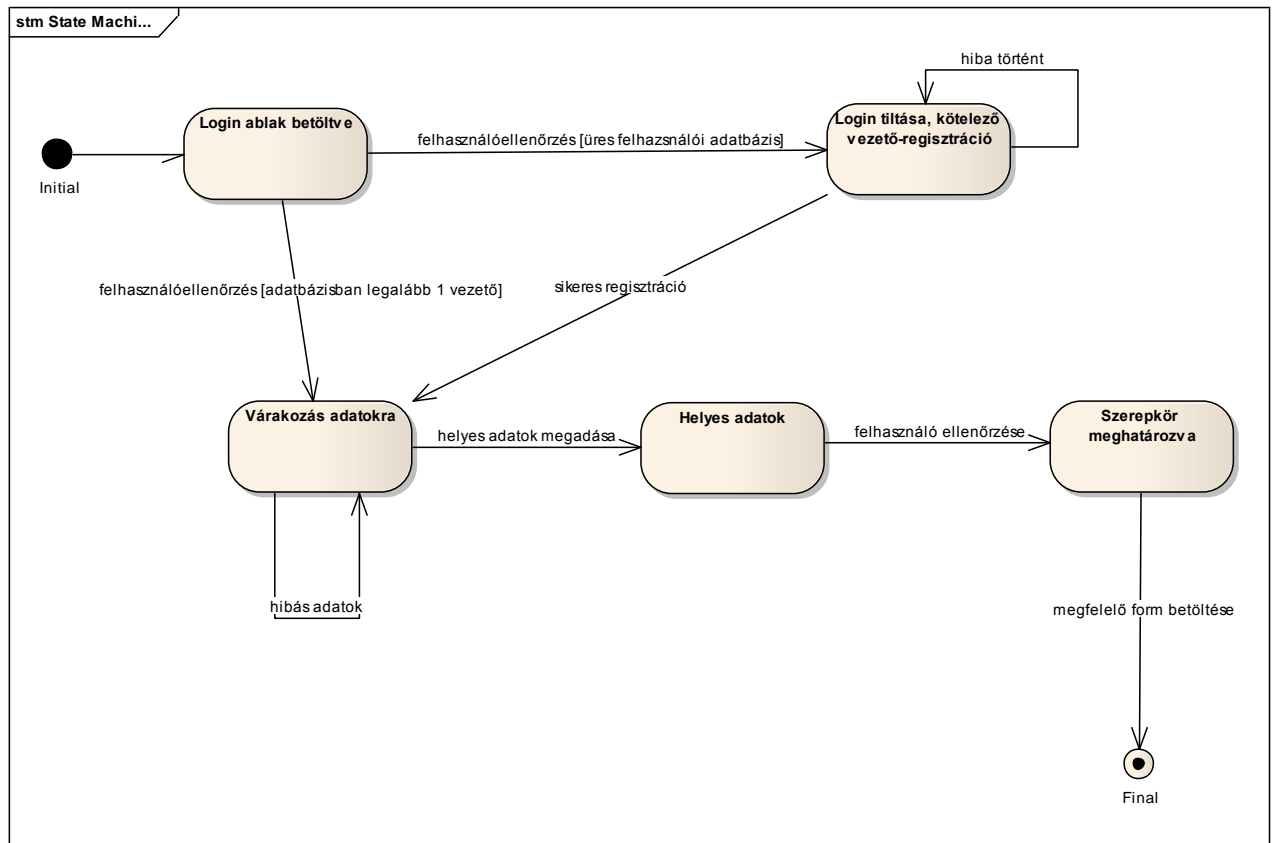




## Állapot diagramok

A programban jól elhatárolható állapotokat az alábbi osztályok, felületek vehetnek fel:

### 1. Megjelenítési réteg – Bejelentkező ablak

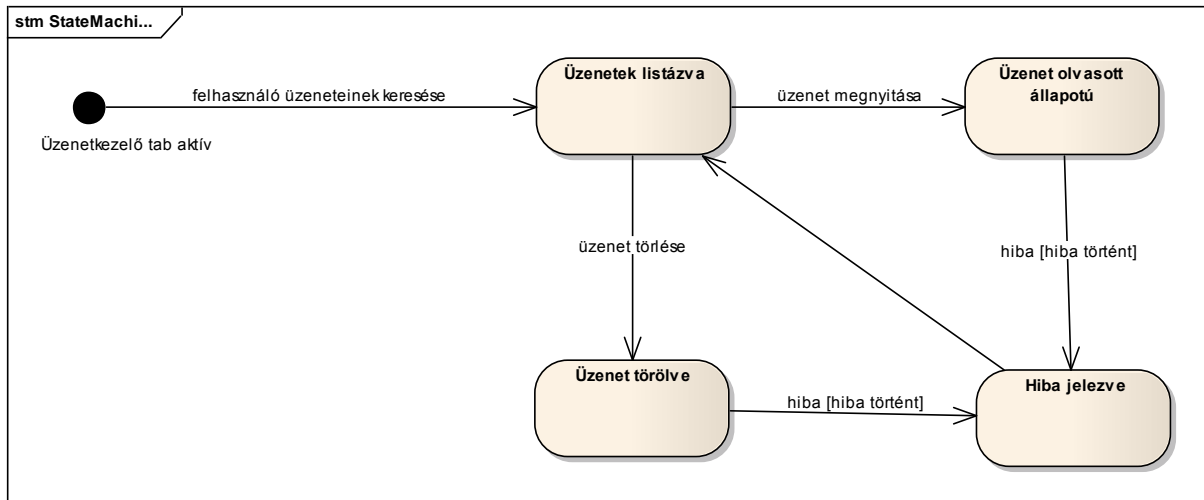


A bejelentkezőablak állapotai

A program elindítása után betölt a bejelentkező ablak, majd ebből az állapotból elindul egy automatikus keresés, ami ha talál már felvett vezető (létezik legalább egy vezető jogkörű felhasználó), átléptet az „Adatok várakozása” állapotra. Abban az esetben, ha nincs még regisztrált vezető letiltódik a felhasználónév és jelszó bevitel, csak egy gomb lesz aktív, amivel regisztrálhatjuk a legelső vezető. Bármilyen hiba esetén visszatérünk a „Login tiltása, kötelező vezető-regisztráció” állapotba, ellenkező esetben (sikeres volt a folyamat), továbblépünk a „Várakozás adatokra” állapothoz, melyben már nem látszódik regisztráló gomb és aktívak a beviteli mezők.

Hibás adatok megadásakor a felület nem enged tovább, érvényes felhasználónév és jelszó esetén továbbmegyünk a „helyes adatok állapotba”, melyből ismét automatikusan, lekérdezzük a bejelentkezett felhasználó jogkörét. Amint megjön az eredmény betöltjük a jogkörhöz tartozó formot és bezárjuk a bejelentkező ablakot.

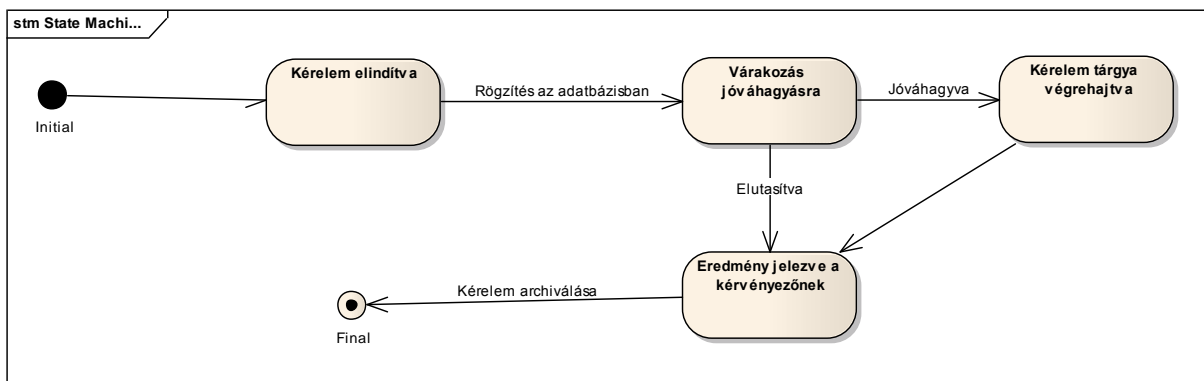
2. Alkalmazás réteg – Üzenetkezelő osztály



Az üzenetek felvehető állapotai

Amint rálépünk a formok üzenetkezelő tabjaira elkezdjük betölteni a felhasználóhoz tartozó üzeneteket, amint elkészült és kiíródott a lista, belépünk az „Üzenetek listázva” állapotba, ahonnan két lehetőségünk nyílik. Amennyiben törölni akarunk egy levelet biztonsági okokból az adatbázisból nem törlődik, de a státusza töröltre módosul, így többet nem lesz listázva a felhasználónál. Egy új üzenet megnyitásakor a státusza olvasottá válik és vizuálisan el is különítődik a többi olvasatlan levéltől, úgy ahogy egy mindennapi levelezéskezelőtől ezt elvárhatjuk.

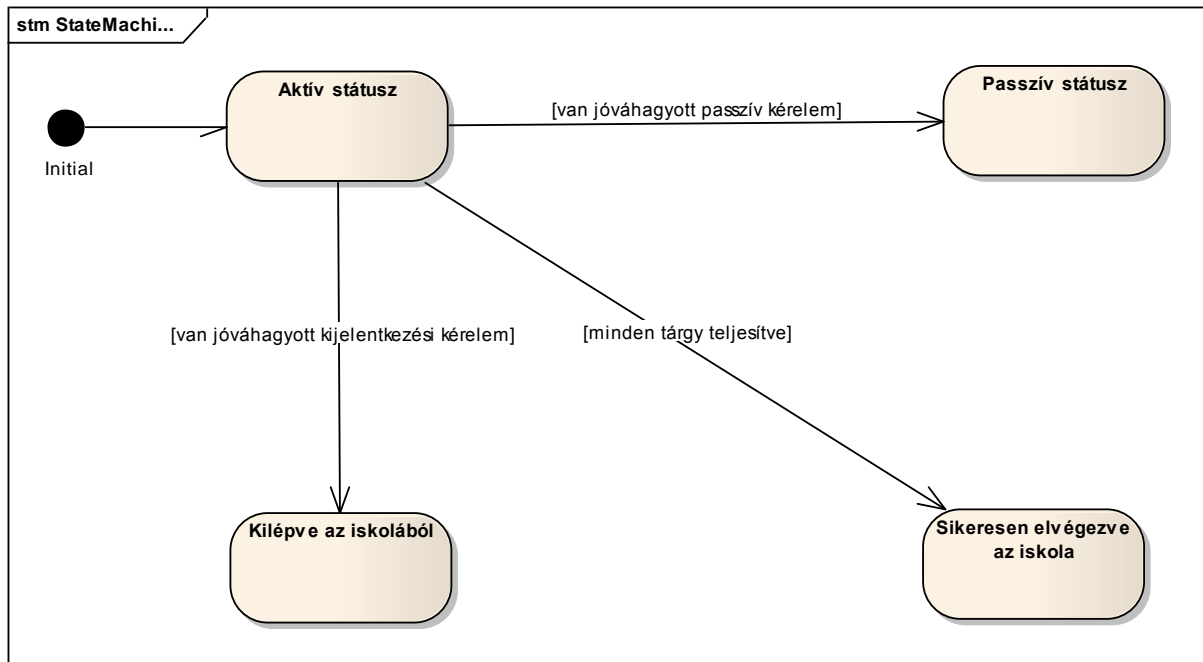
3. Alkalmazás réteg – Kérelemkezelő



Kérelmek kezelésének állapotai

Az elindított kérelmet először rögzítjük az adatbázisban, majd elküldjük a vezetőknek jóváhagyásra. Pozitív elbírálás esetén végrehajtjuk a kérelemhez fűződő változtatásokat, majd jelezzük a döntés eredményét a kérelmezőnek. Elutasítás esetén szintén csak értesítjük a kérelmezőt.

## 4. Alkalmazás réteg – Hallgató osztály



A hallgató lehetséges státuszai

Alapértelmezésként minden hallgató aktív státusszal indul, de kérelmezés útján elérhető számukra a kijelentkezett státusz, ekkor megszűnik a jogviszony, a hallgató nem lesz többé tagja az intézménynek, adatai archiválódnak. Élhetnek passzív félévvel, melyben a jogviszony tovább él, de tárgyakat nem vehet fel. Abban az esetben, ha minden tárgyat sikerül teljesítenie, a hallgató sikeresen elvégezte az iskolát.

## Használati útmutató

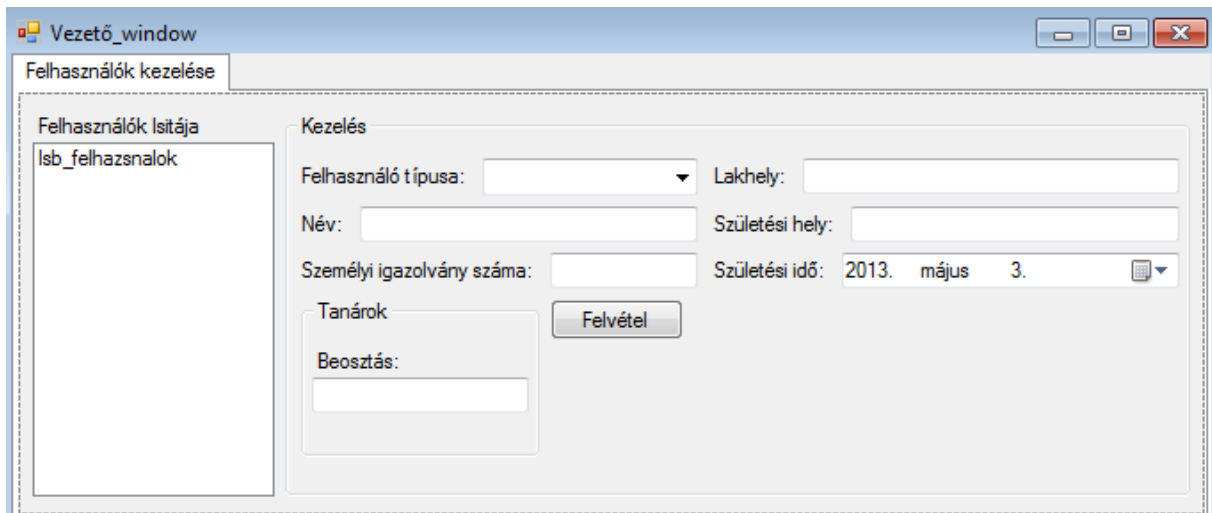
A program egy login ablakkal indul, melyben lehetőségünk van megadni a kapott Zeuszkódot és jelszavat. A prototípus xml-be dolgozik, nem szükséges semmilyen külső program telepítése, az eltárolt hallgatókat a Hallgató.xml, a tanárokat a Tanár.xml és a vezetőket a Vezető.xml-ben menti. A tesztelés idejére létre lett hozva mindegyik típusból egy tesztfelhasználó, név szerint

- teszth                                      A hallgatók tesztfelhasználója
- teszt                                        A tanárok tesztfelhasználója
- tesztv                                      A vezetők felhasználója

Minden generált felhasználónak egységesen „passw1234” a jelszava, a későbbiekben ezt mindenki szabadon módosíthatja.

Sikeres belépés esetén jogkörünktől függően más-más felület jelenik meg.

Vezetőként az alábbi felület fogad bennünket:



Vezetői ablak

A prototípusban jelenleg a vezetői funkciók közül az adatkezelés aktív, ennek segítségével vehetünk fel embereket az intézménybe, tanárok esetén mindez még kiegészíthető a közvetlen beosztással.

A tanári jogkörrel az alábbi prototípusfelületet láthatjuk:

Tanár\_window

Tantárgyak kezelése

Tanított tantárgyak:

lsb\_tantargyak

Kezelés

Tárgy kódja:

Tantárgy neve:

Hét:

Terem:

Kezdés és befejezés ideje:

0 0

0 0

Követelmény:

Segédlet:

Tantárgy felvétele

Tantárgy módosítása

Tantárgy törlése

Oktatók

lsb\_oktatok

Hozzáad

Töröl

A tárgyat oktatják:

lsb\_tanarok

Tanári felület

A tanároknál a tárgykezelés funkció lett megvalósítva, a bejelentkezett tanár létrehozhatja saját óráit, módosíthatja, törölheti őket. A jövőben itt lehetőség van a vezetői kontrol beiktatására, jóváhagyások alkalmazásával. Bal oldalt láthatjuk a tanár saját tantárgyait, arra kattintva megjelenik minden lényeges információ a tárgyról. Jobb oldalt pedig a jövőben plusz oktatókat is rendelhetünk a tantárgyakhoz, jelen verzióban egy tárggyhoz még csak egy tanár adható hozzá.

Végül, ha hallgatóként lépünk be az alábbi ablakok jelennek meg:

The screenshot shows a web application window with the title "Belépve teszthallgató néven." (Logged in as test student). The window has two tabs: "Adatok kezelése" (Data management) and "Tantárgyak kezelése" (Subjects management). The "Adatok kezelése" tab is active, showing a form for "Személyi adatok megváltoztatása" (Change personal data). The form contains the following fields and controls:

- Zeusz kód: teszth
- Név: teszthallgató
- Személyi igazolvány szám: asdf423
- Lakhely: nagy Város
- Születési hely: kis város
- Születési dátum: 2013. május 1.
- Jelszó módosítás: passw1234
- Jelszó módosítás button
- Küldés button

Below the form, there is a "Státusz" (Status) section:

- Jelenlegi hallgató státusz: aktív
- ☒ Passzív kérelem 1 félévre
- ☐ Kijelentkezés az iskolából
- Kérelem elküldése button

Hallgatói felület - adatkezelés

Itt lehetősége van a hallgatónak módosítani személyes adatait, melyek azonnal el is tárolódnak.

Szintén ezen a felületen van lehetősége a hallgatóknak saját jelszavuk megváltoztatására is, melyet a következő bejelentkezéskor más használhatnak is.

A prototípus jövőjét hivatott ábrázolni a passzív- és kijelentkezési kérelem küldés, ennek segítségével egy kattintásra lesz csak szükség, ha a hallgató jelezni szeretné szándékát ha passzív félévre menne vagy pedig végleg elhagyná az intézményt.

A következő felület a hallgatói tárgykezelés:

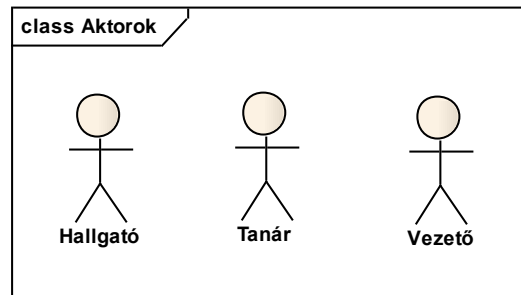
Hallgatói felület - tárgykezelés

A prototípus limitációi miatt még nem áll módunkban kipróbálni a tényleges tárgyfelvevételt, de láthatjuk, hogy a kevésbé gyakorlott felhasználók számára is áttekinthető struktúrában, bal oldalt a számára felvehető és jobb oldalt a felvett tárgyak helyezkednek el. A tárgyra kattintva pedig megjelenik minden hallgatók számára fontos információ a kurzusról.

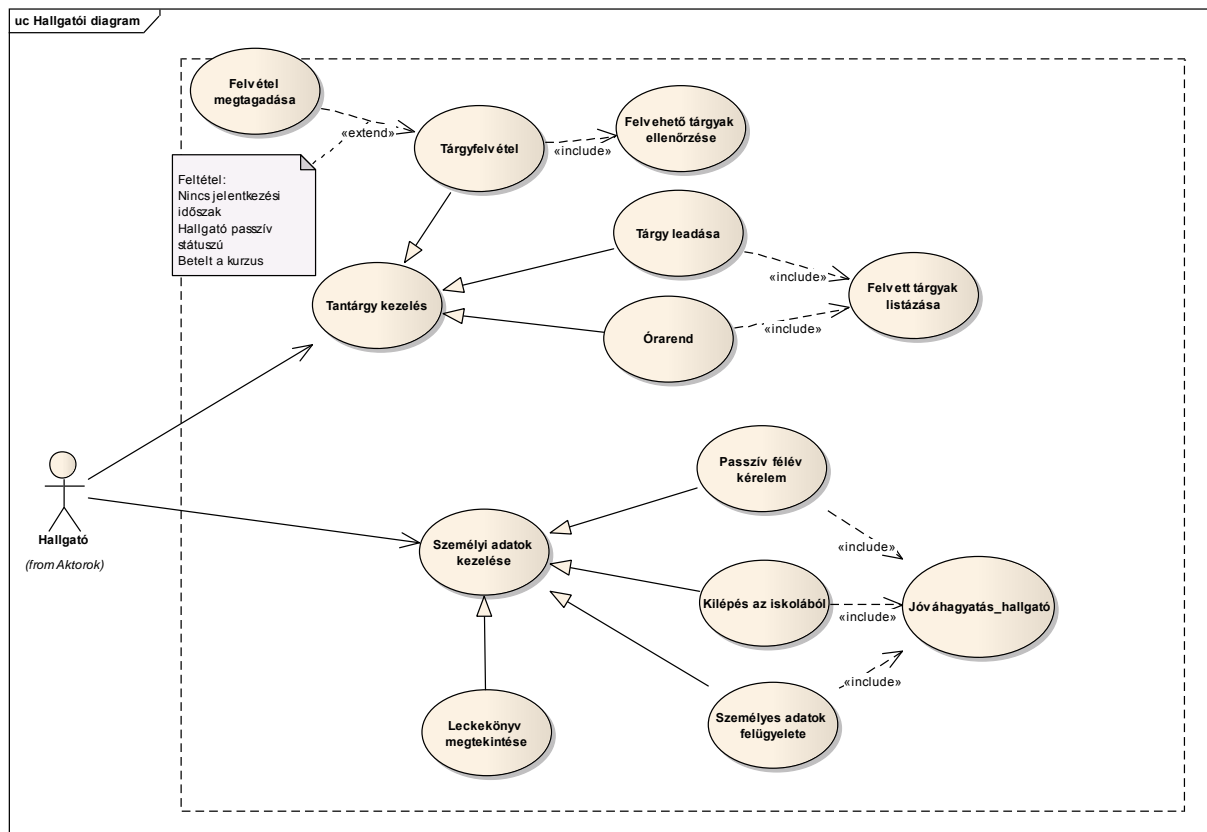


## 1. Melléklet Használati eset modell

A tervezett szoftver rendszer felhasználói három csoportba sorolhatók, úgymint: Hallgatók, Tanárok, Vezetők. Ezek a felhasználói csoportok a használati eset modellben aktorokként jelennek meg. Ezeket az aktorokat szemlélteti a következő ábra:



A rendszer mindegyik felhasználó csoport számára más és más funkciókat biztosít. Az egyes aktorok és az általuk igénybe vehető rendszer funkciók kapcsolatát szemléltetik az alábbi ábrák.



A „Tantárgy kezelés”, „Személyi adatok kezelése” használati esetek különböző rendszer-funkciók általánosításai, ezekből származnak a különböző alapfunkciók: „Tárgyfelvétel”,

## Szoftvertechnológia gyakorlat – OE-NIK – 2012/2013/2

„Tárgy leadása”, „Órarend”, „Passzív félév kérelem”, „Kilépés az iskolából”, „Személyes adatok felügyelete”, „Leckekönyv megtekintése”.

A „Felvétel megtagadása”, „Felvehető tárgyak ellenőrzése”, „Felvett tárgyak listázása” és a „Jóváhagyatás\_hallgató” olyan rész-funkciók, amelyek különféle adatokat ellenőriznek az egyes alapfunkciók során.

### Tárgyfelvétel

1. Bejelentkezés
2. Felvehető tárgyak listázása
3. Kiválasztott tárgyra jelentkezés

### Tárgy leadása

1. Bejelentkezés
2. Felvett tárgyak listázása
3. Leadandó tárgy kiválasztása
4. Leadás

### Órarend

1. Félév kiválasztása
2. Felvett tárgyakból órarend listázása

### Passzív félév kérelem

1. Bejelentkezés
2. Személyes adatok menü kiválasztása
3. Passziváltatás jelzése
4. Indoklás
5. Kérelem benyújtása

### Kilépés az iskolából

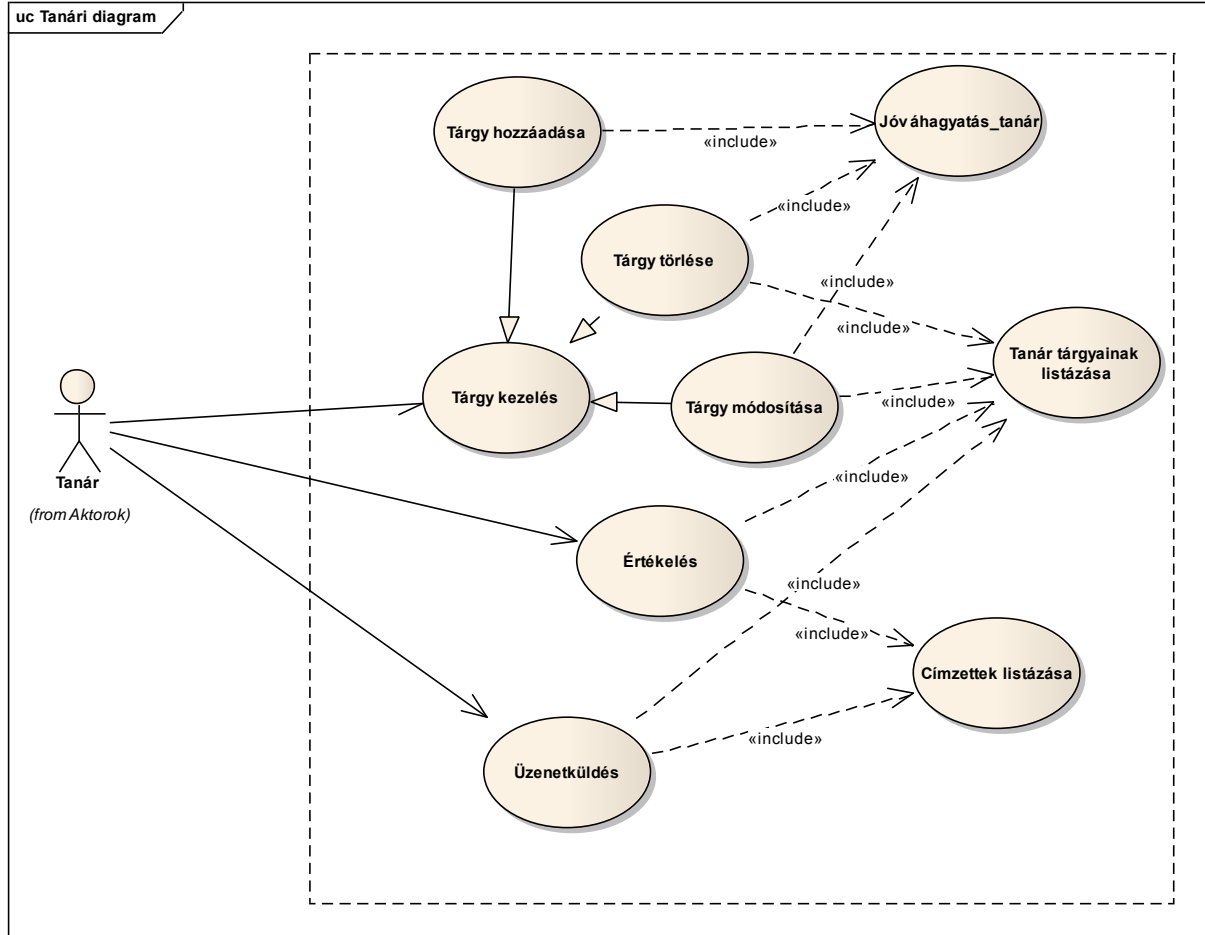
1. Bejelentkezés
2. Személyes adatok menü kiválasztása
3. Indoklás
4. Kérelem elküldése

### Személyes adatok felügyelete

1. Bejelentkezés
2. Személyes adatok menü kiválasztása
3. Kívánt adat módosítása, indoklása
4. Mentés, küldés jóváhagyásra

### Leckekönyv megtekintése

1. Bejelentkezés
2. Leckekönyv menü kiválasztása
3. Kívánt félév kiválasztása
4. Listázás



A „Tantárgy kezelés” olyan használati eset ami pár rendszer-funkció általánosítása, ebből származik a „Tárgy hozzáadása”, „Tárgy törlése”, „Tárgy módosítása” alapfunkciók. További alapfunkció még az „Értékelés” és az „Üzenetküldés”.

A „Tanár tárgyainak listázása”, „Címzettek listázása” és a „Jóváhagyás\_tanár” olyan rész-funkciók, amelyek különféle adatokat ellenőriznek az egyes alapfunkciók során.

### Tárgy hozzáadása

1. Bejelentkezés
2. Tárgy hozzáadása menü kiválasztása
3. Tárgyadatok megadása (terem, időpont, létszám)
4. Kérelem továbbítása jóváhagyásra

## Tárgy törlése

1. Bejelentkezés
2. Saját tárgyak listázása
3. Tárgy kiválasztása
4. Törlési kérelem + indoklás jelzése

## Tárgy módosítása

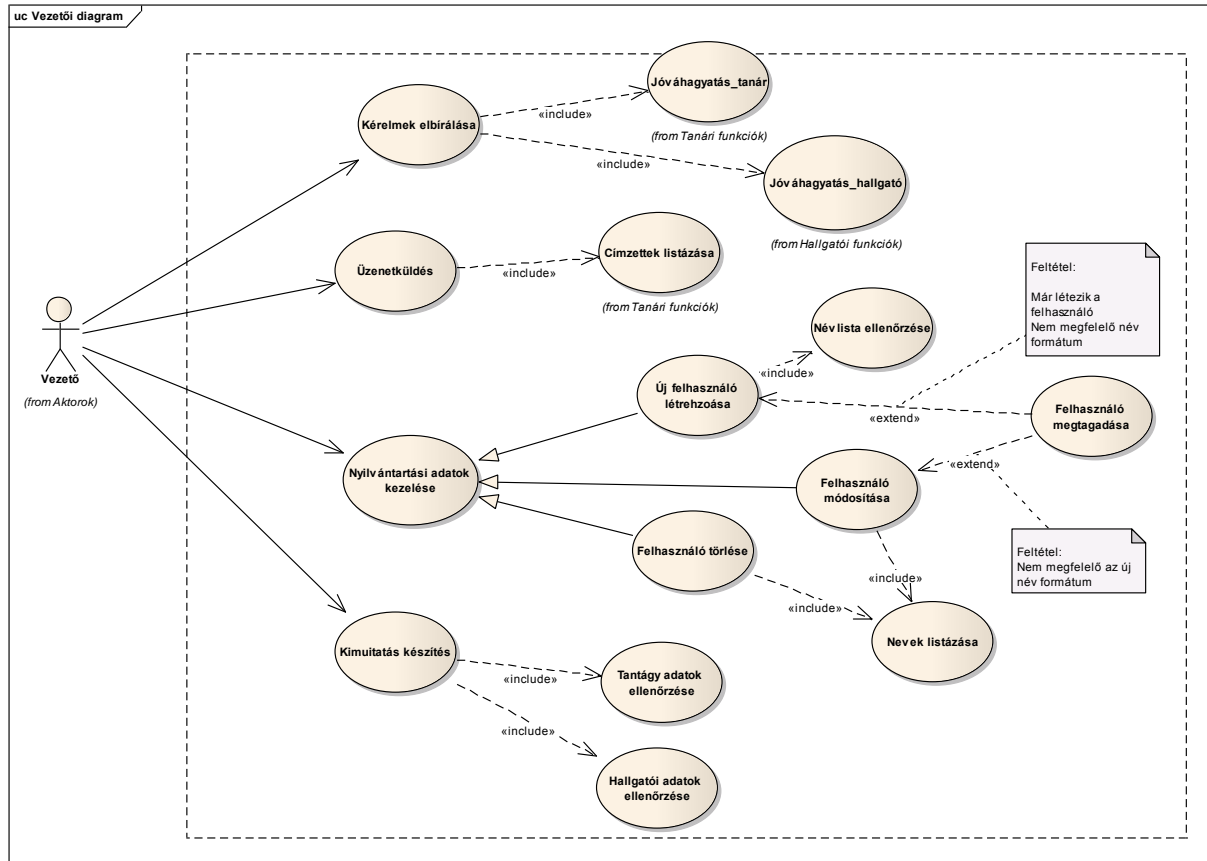
1. Bejelentkezés
2. Saját tárgyak listázása
3. Tárgy kiválasztása
4. Adatok módosítása
5. Igény elküldése jóváhagyásra

## Értékelés

1. Bejelentkezés
2. Saját tárgyak listázása
3. Kursus hallgatóinak listázása
4. Érdemjegyek beírása
5. Mentés

## Üzenetküldés

1. Saját kurzusokra járó hallgatók listázása
2. Hallgató(k) kiválasztása
3. Üzenet írása
4. Küldés



A „Nyilvántartási adatok kezelése” olyan használati eset ami pár rendszer-funkció általánosítása, ebből származik a „Új felhasználó létrehozása”, „Felhasználó módosítása”, „Felhasználó törlése” alapfunkciók. További alapfunkció még az „Kérelmek elbírálása”, az „Üzenetküldés”, „Kimutatás készítés”.

A „Jóváhagyás\_hallgató”, „Jóváhagyás\_tanár”, „Névlista ellenőrzése”, „Felhasználó megadása”, „Nevek listázása”, „Tantárgy adatok ellenőrzése” és a „Hallgatói adatok ellenőrzése” olyan rész-funkciók, amelyek különféle adatokat ellenőriznek az egyes alapfunkciók során.

### Új felhasználó létrehozása

1. Bejelentkezés
2. Nyilvántartási adatok fül
3. Új felhasználó hozzáadása
4. Felhasználó adatainak megadása
5. Regisztrálás

### Felhasználó módosítása

1. Bejelentkezés
2. Nyilvántartási adatok fül
3. Felhasználók listázása
4. Kívánt felhasználó kiválasztása
5. Szerkesztés gomb

6. Adatok módosítása
7. Mentés gomb

### Felhasználó törlése

1. Bejelentkezés
2. Nyilvántartási adatok fül
3. Felhasználó kiválasztása
4. Felhasználó törlése gomb

### Kérelem elbírálása

1. Bejelentkezés
2. Jóváhagyásra váró menü
3. Hallgatói és Tanári kérelmek listázása
4. Jóváhagyás/elutasítás

### Üzenetküldés

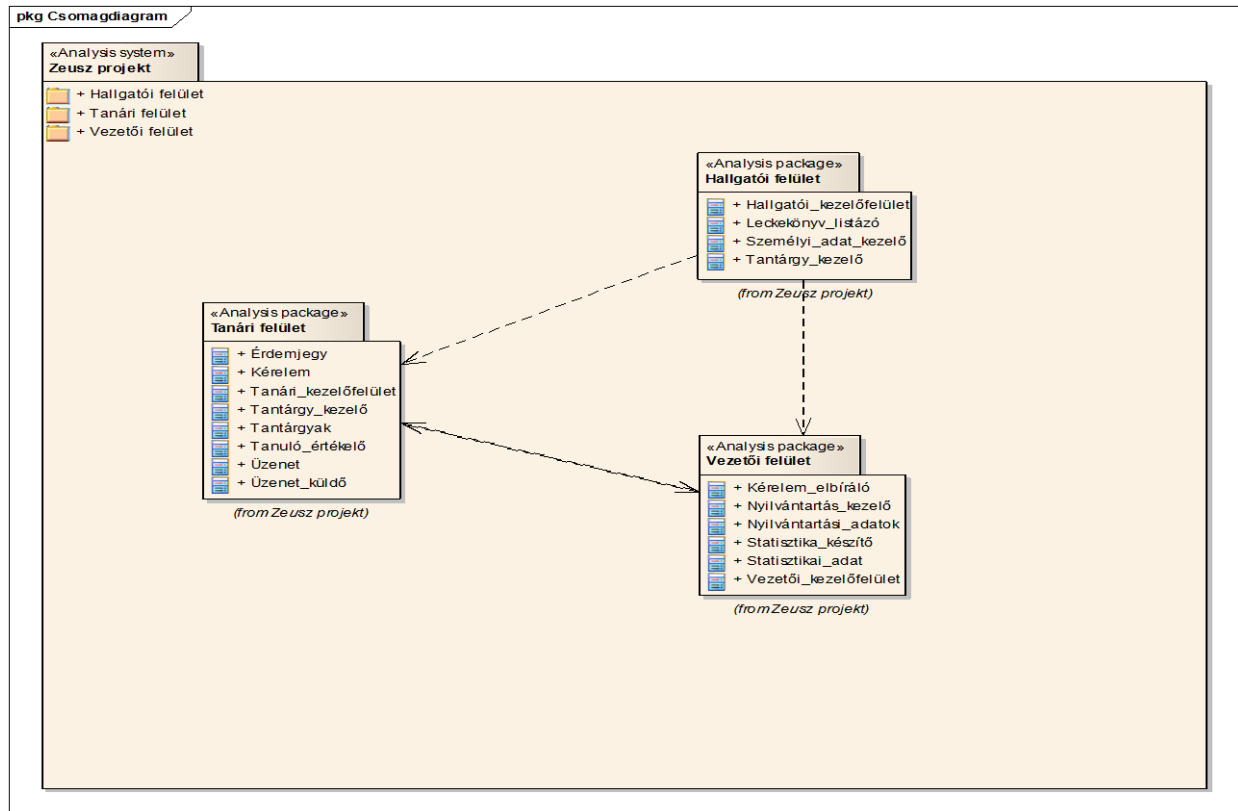
1. Bejelentkezés
2. Üzenet küldése menü
3. Címzett kiválasztása
4. Üzenet megírása
5. Elküldés

### Kimutatás készítés

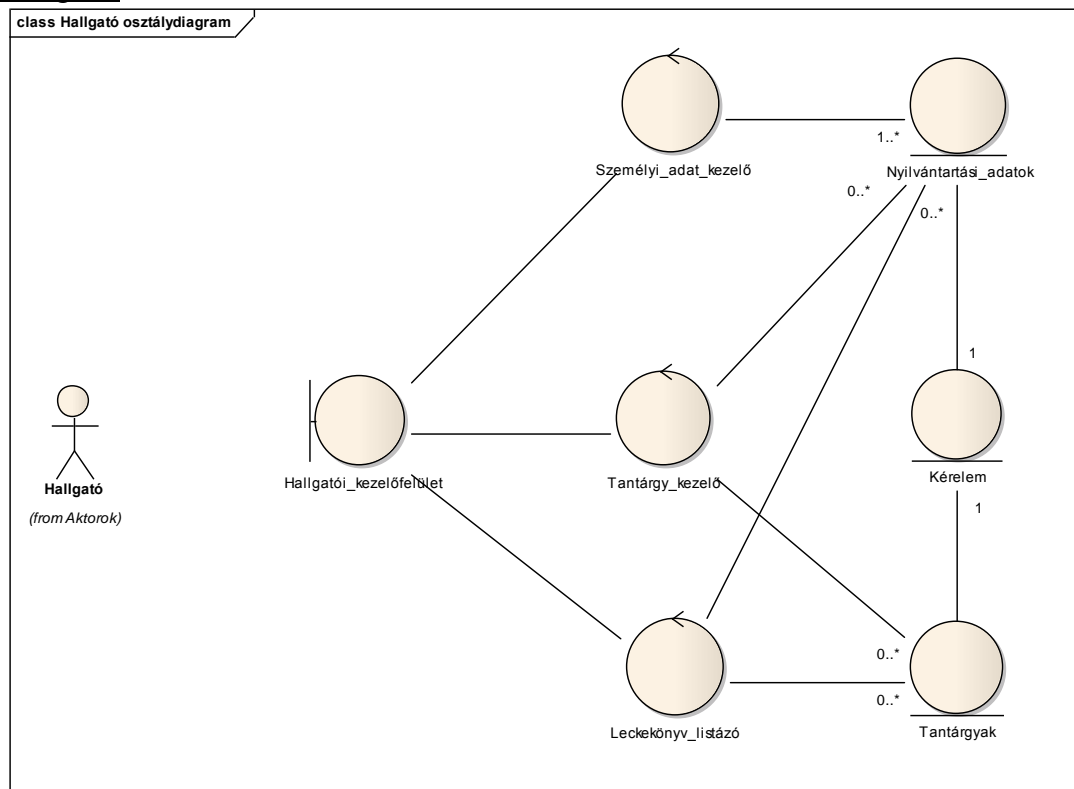
1. Bejelentkezés
2. Időszak kiválasztása
3. Tantárgy(ak) és/vagy hallgató(k) kijelölése
4. Eredmények, teljesítési mutatók, statisztikák..etc listázása

## 2.Melléklet Analízis modell

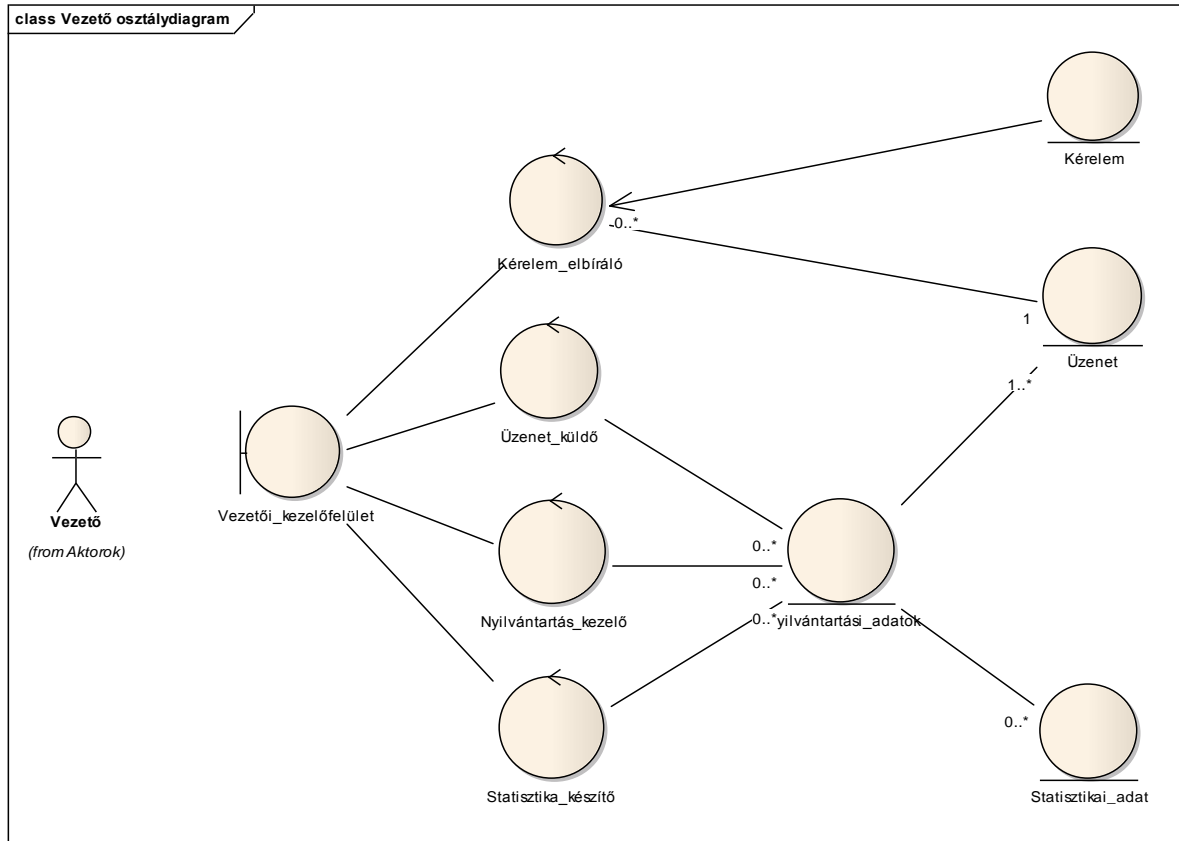
### A rendszer szerkezete:



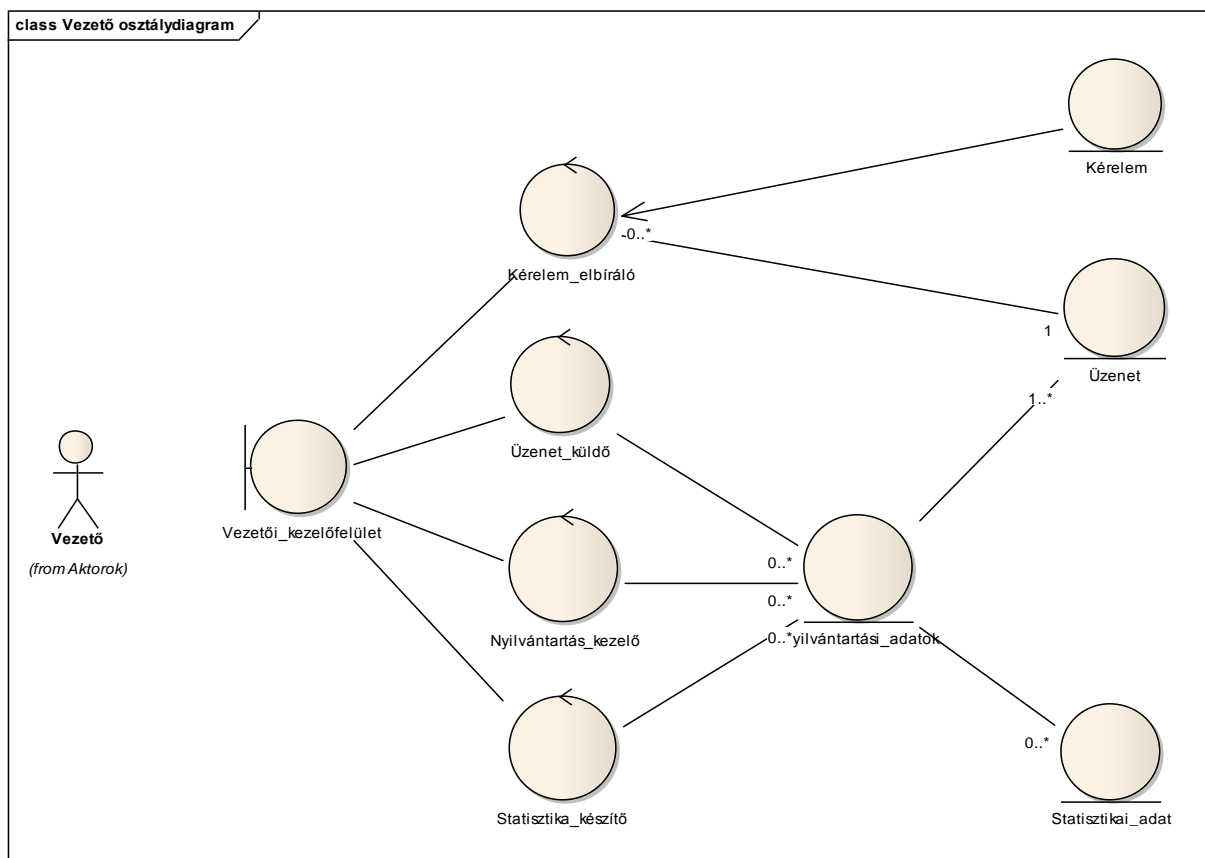
### Hallgató:



Tanár:



Vezető:



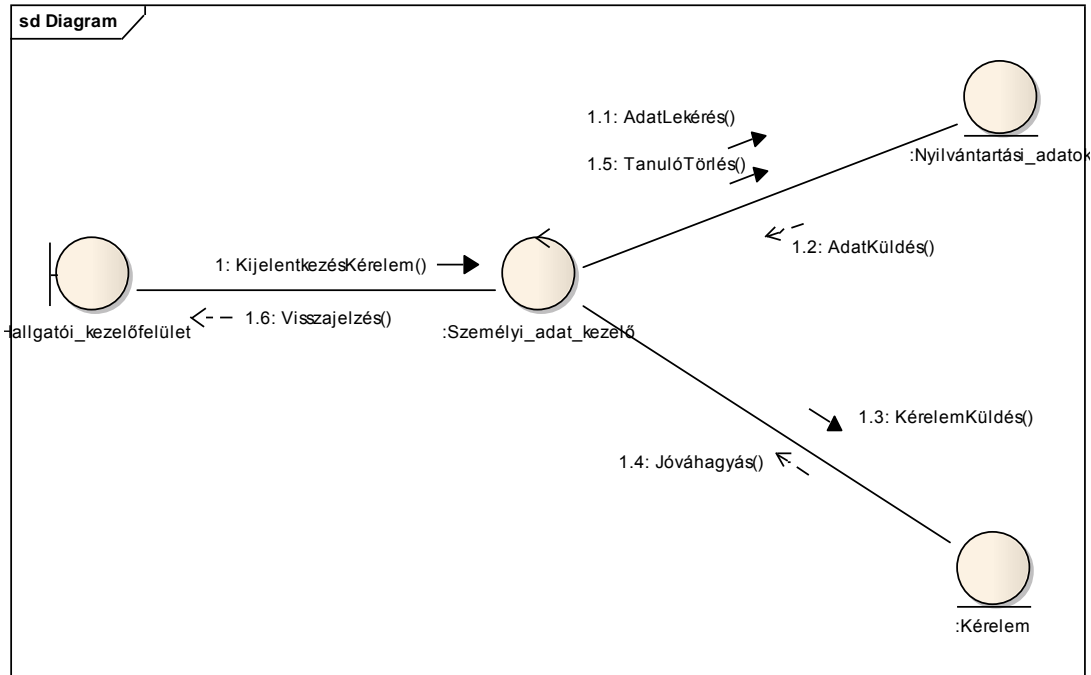


Az előző ábrákon látható szerkezeti elemek segítségével modellezzük a rendszer működését az egyes alapfunkciók biztosítása érdekében.

### Kommunikációs diagram

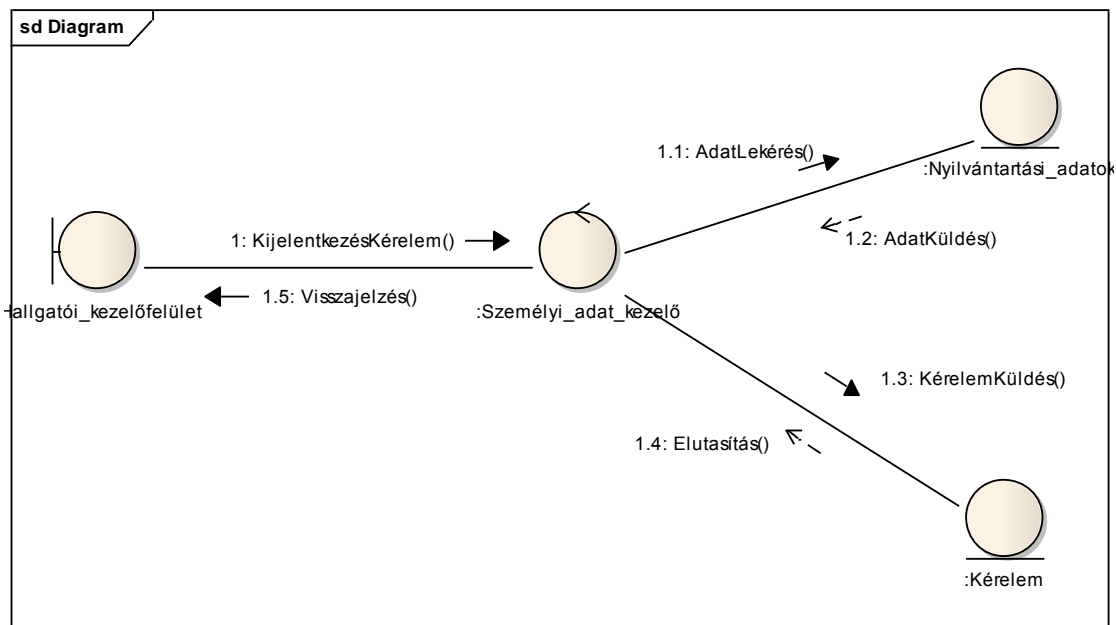
Hallgató kijelentkezési kérelem:

Normál működés:



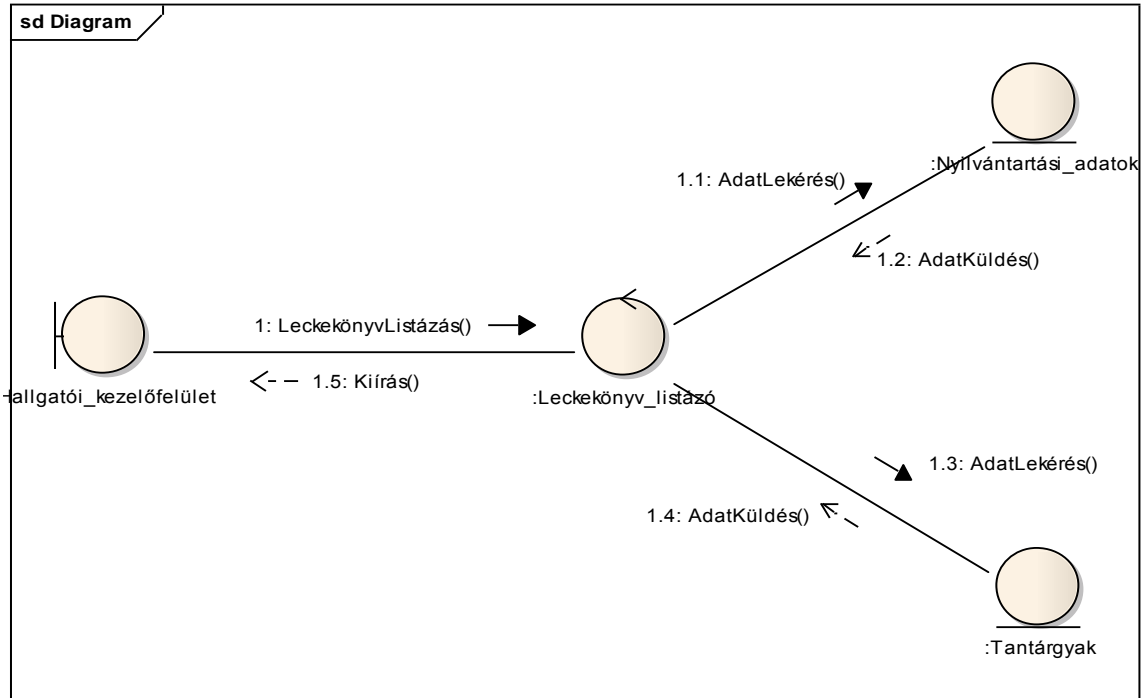
A hallgató elküldi az iskolából való kijelentkezés kérelmét, ezt feldolgozza a Személy adat kezelő, rögzíti az adatbázisban, majd a kérelmet elküldi jóváhagyatásra. Pozitív eredmény esetén a kezelő törölteti a hallgatót és visszajelez a hallgatónak.

Elutasított működés:



Itt elutasítják a hallgató kérelmét, ezért a Kérelem kezelőből érkező választ rögtön küldjük tovább a hallgatónak.

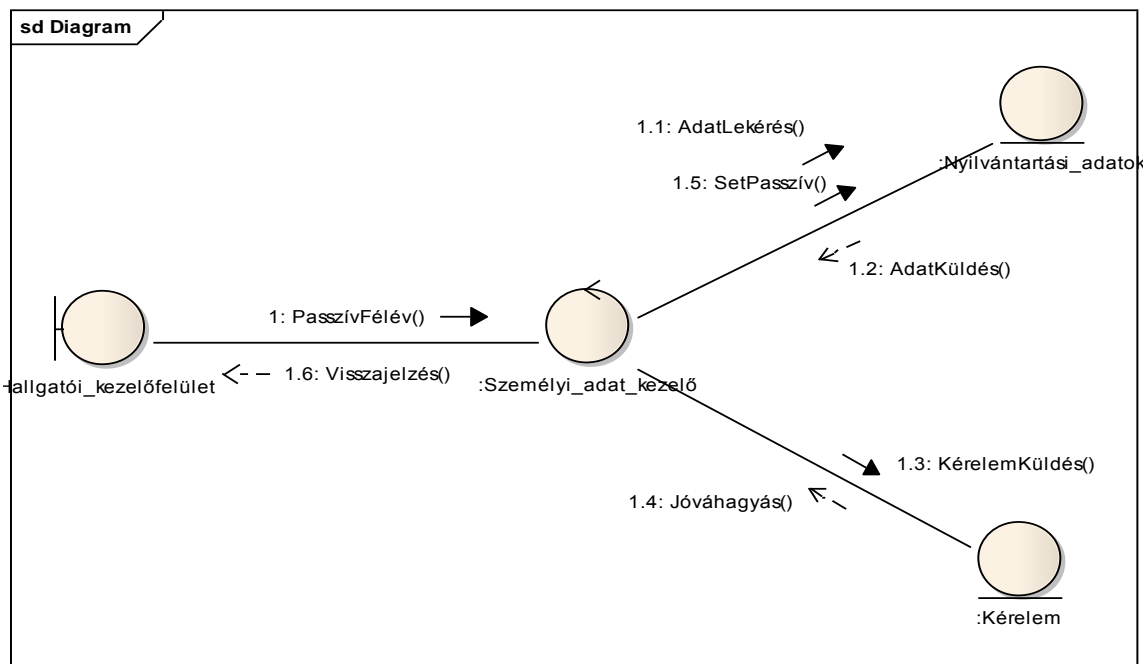
## Hallgató Leckekönyv lekérése:



A hallgató elindítja a felületén a folyamatot, a leckekönyv listázó ellenőrzi a felhasználót, bekéri a státuszát, majd ezek segítségével lekérdezi a hallgató által felvett tárgyakat. A visszakapott adatokat szerkeszti és megjeleníti a felületen.

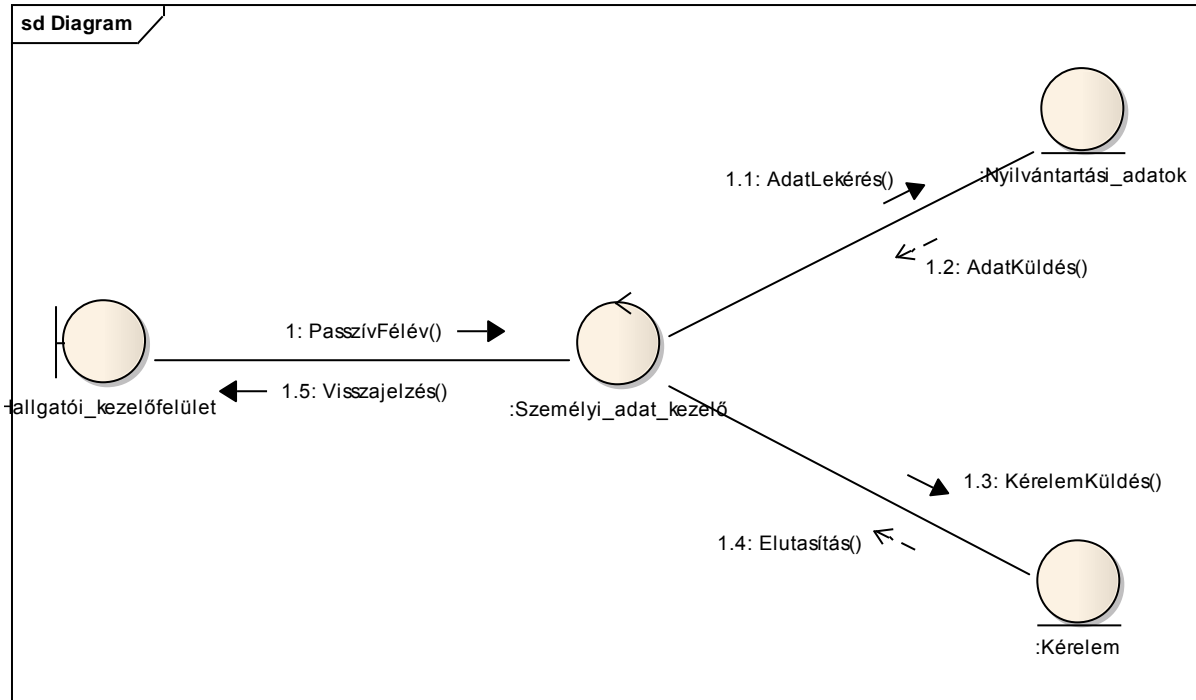
## Hallgató Passzív félév kérelem:

Normál működés



A személyi adat kezelő ellenőrzi a hallgatót az adatbázisban, a visszakapott értékek segítségével elküldi a kérelmet vezetői jóváhagyásra. Pozitív eredmény esetén beállítja az adatbázisban a hallgató státuszát passzívra, majd értesíti a hallgatót.

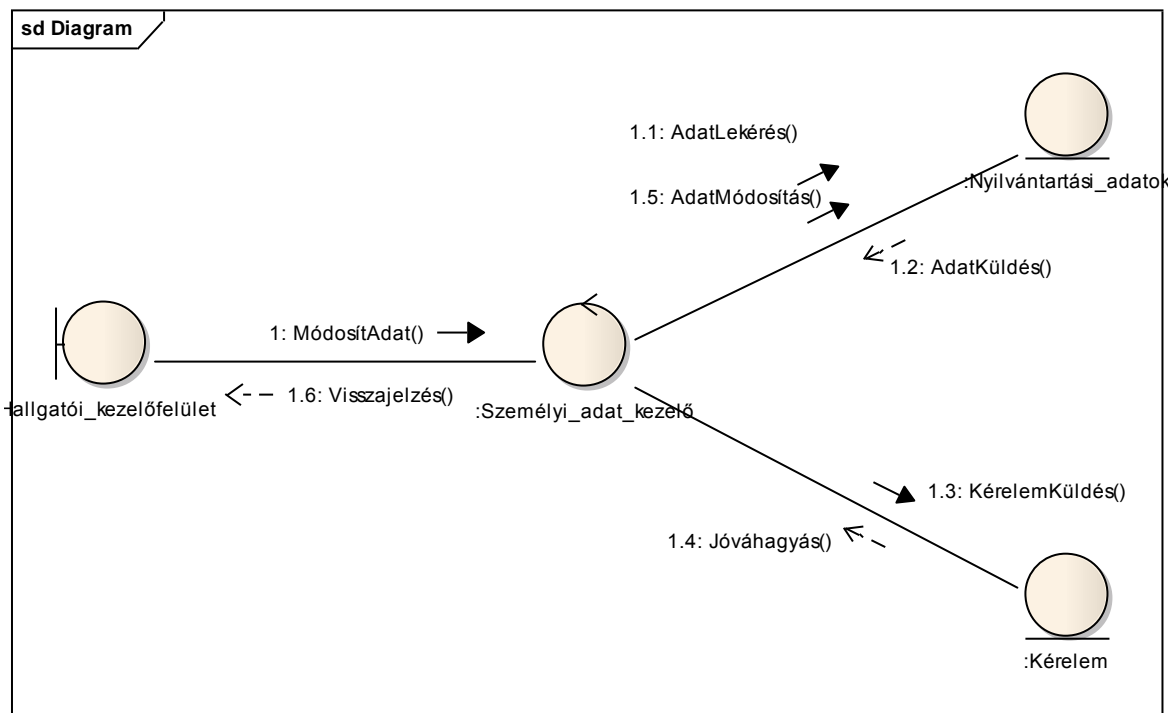
## Elutasított működés



Elutasítás esetén a kezelő a vezetői döntés után rögtön értesíti a hallgatót az eredményről.

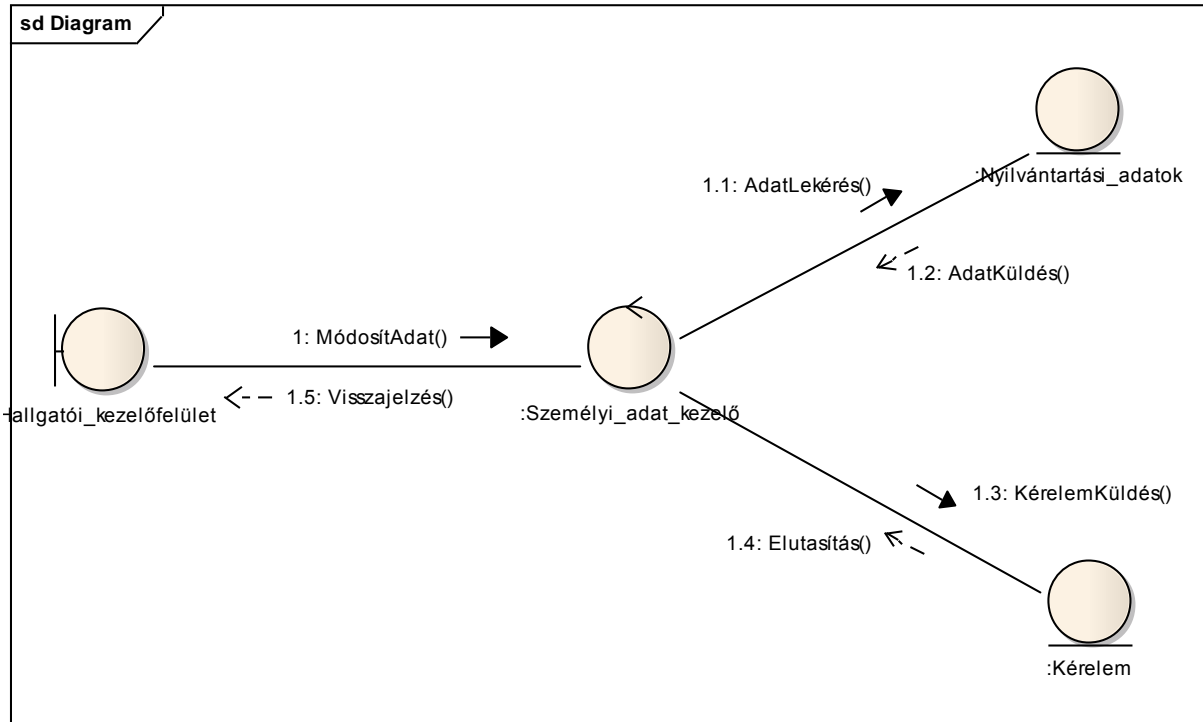
## Hallgató Személyes adat módosítása

### Normál működés



A hallgató a módosítandó adatait elküldi jóváhagyásra, a személyi adat kezelő ellenőrzi a hallgatót, majd továbbküldi jóváhagyásra. Pozitív döntéskor a kezelő módosítja az adatbázisban a kért adatokat és visszajelez a hallgatónak.

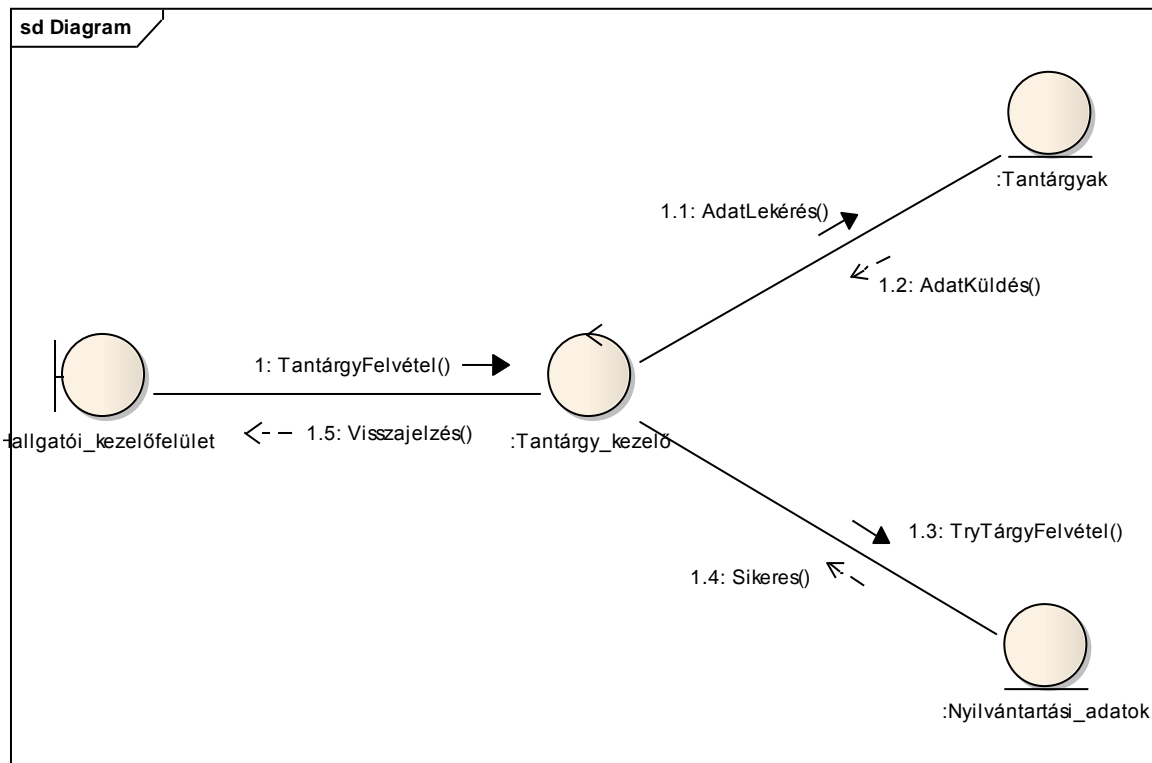
Elutasított működés



A negatív elbírálás esetén a kezelő jelzi a felhasználónak az eredményt.

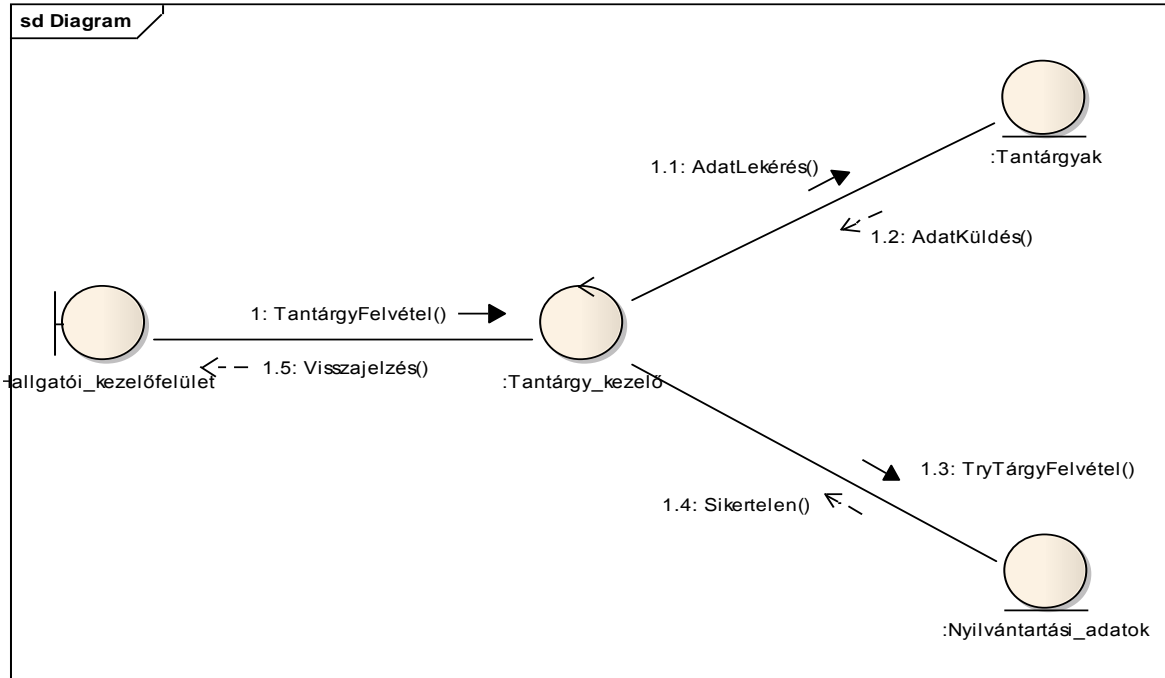
## Hallgató tárgyfelvétel

Normál működés



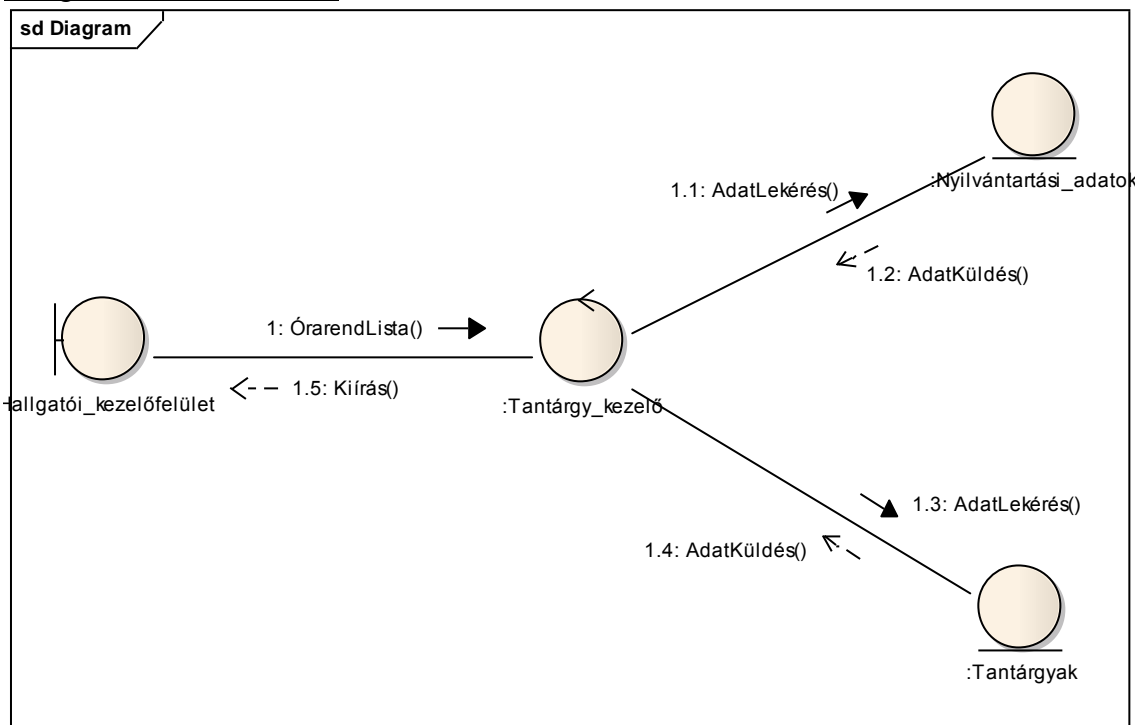
A hallgató elindítja a tárgyfelvételt, a tantárgy kezelő ellenőrzi a tárgyakat, majd megpróbálja azokat rögzíteni a hallgatóhoz. Sikeres felvétel esetén a kezelő visszajelzi a felvételt a hallgatónak.

Elutasított működés



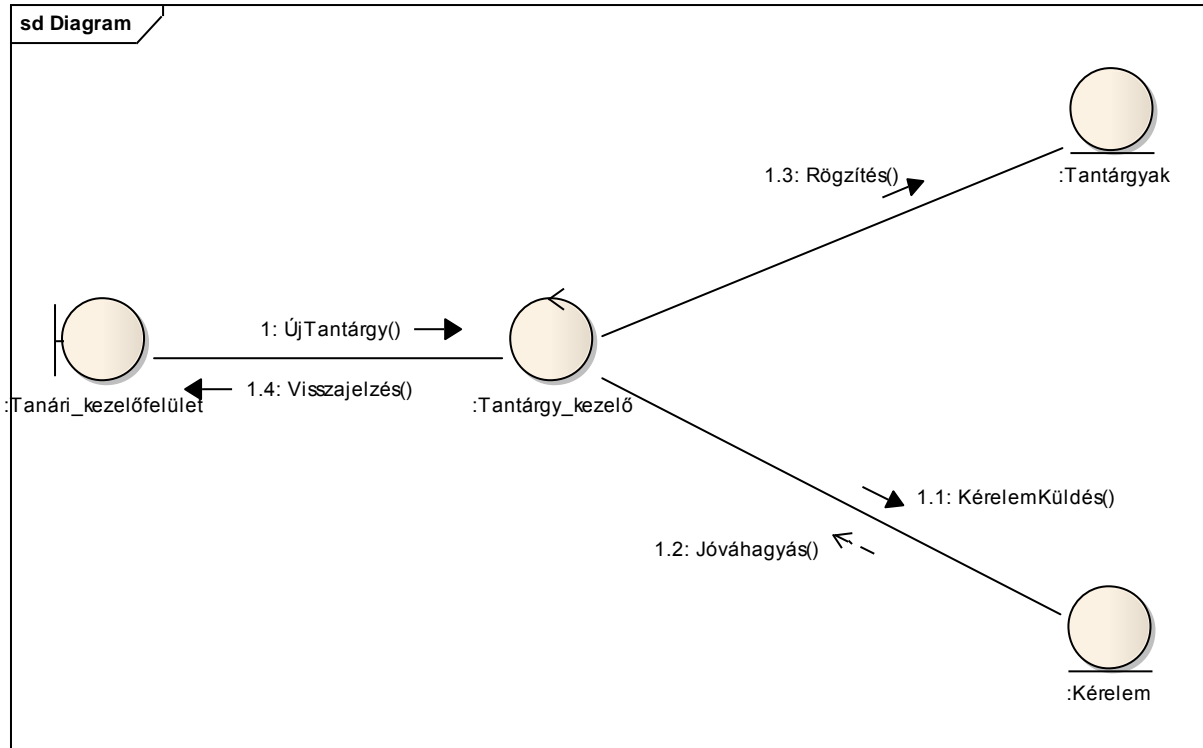
Hiba történik a tárgy hallgatóhoz rendelése közben, ezért a kezelő visszajelzi a hibát a hallgatónak.

Hallgató Órarend kiírás:



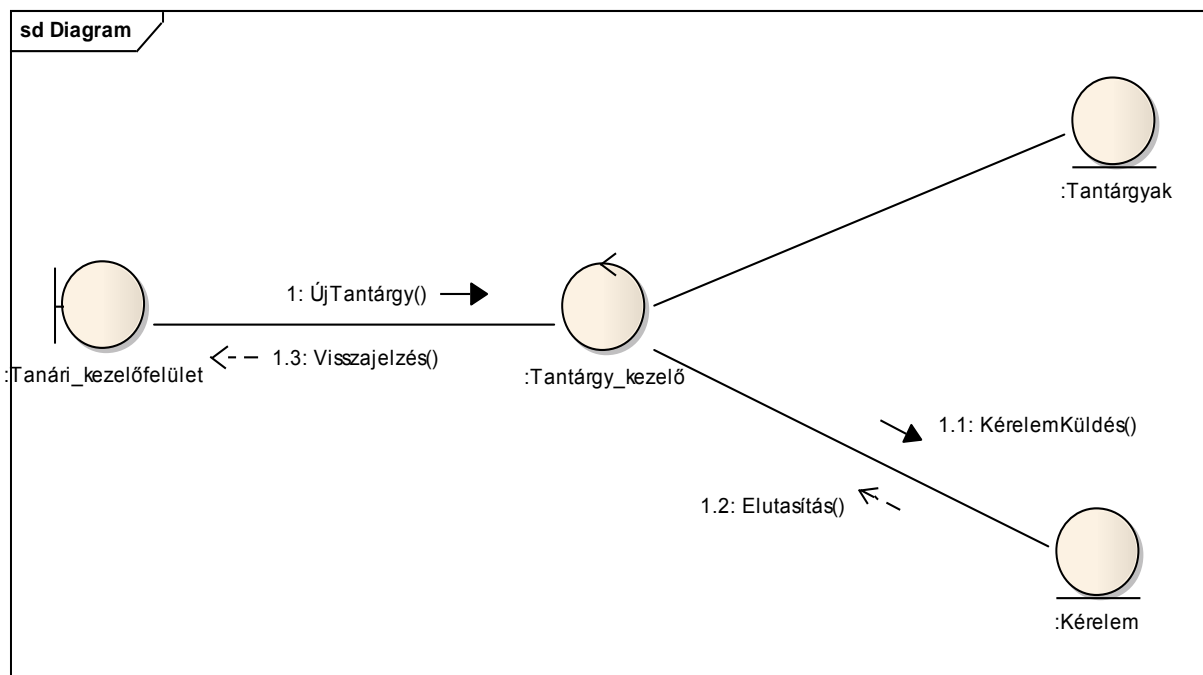
A tantárgykezelő ellenőrzi a hallgatót, majd lekéri a hallgató érintett tárgyait, végül formázva kiírja a felületre.

## Tanár Tantárgyfelvétel: Normális működés



A tanár az új tárgy adatait elküldi, a tantárgykezelő továbbítja jóváhagyásra a vezetéshez. Pozitív válasz esetén beírja a tárgy adatait a tantárgy adatbázisba és visszajelez a felvétel sikerességéről.

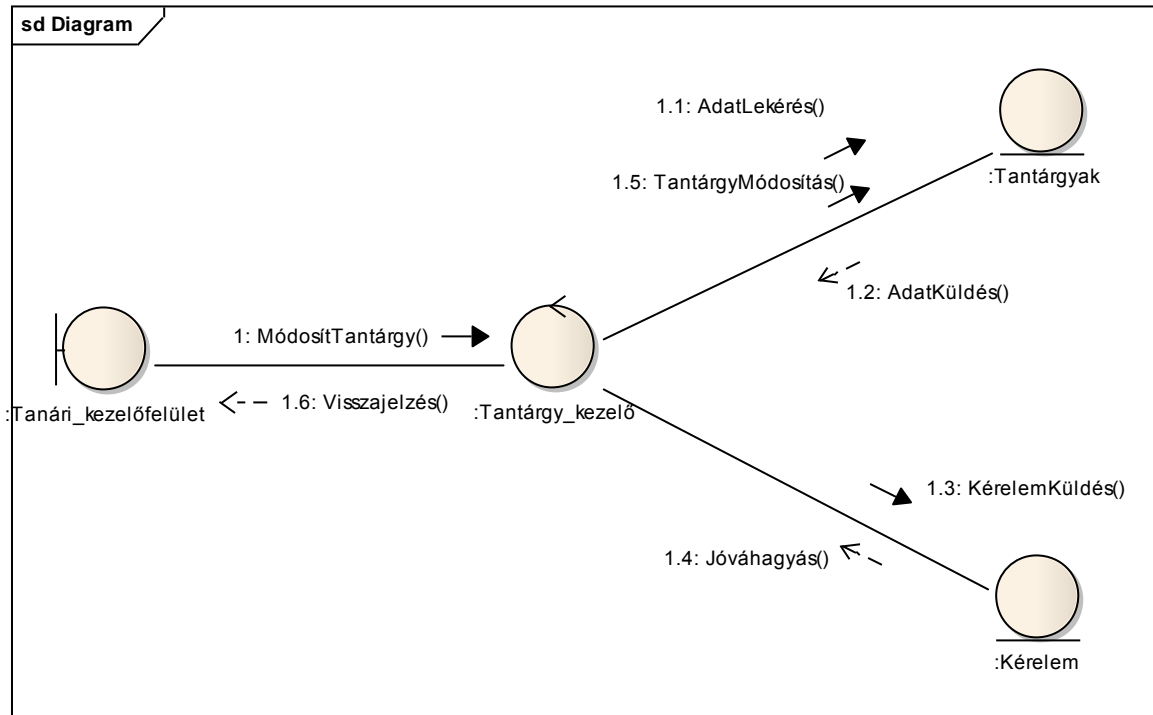
## Elutasított működés



Kimarad a rögzítés, helyette rögtön visszaküldi a kezelő a negatív eredményt.

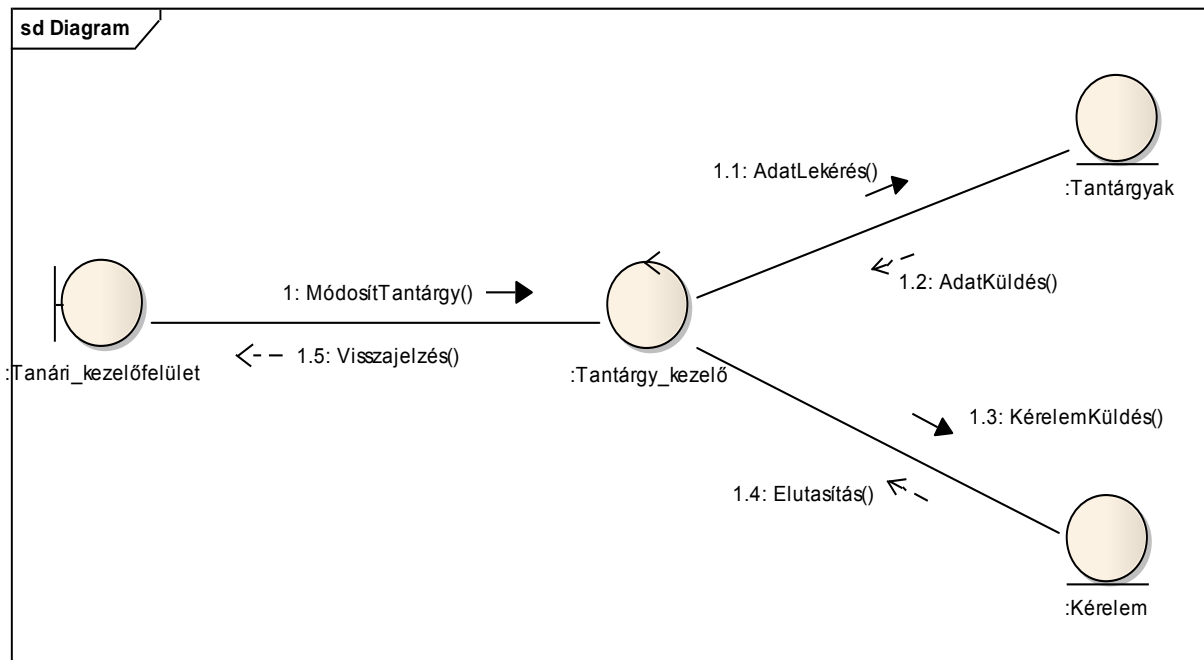
Tanár Tantárgy módosítás:

Normál működés



A kezelő ellenőrzi a tárgy adatait, majd elküldi jóváhagyásra a változtatásokat. Ha elfogadják átírja az adatbázisban a tárgy adatait és visszajelez a sikeres módosításról.

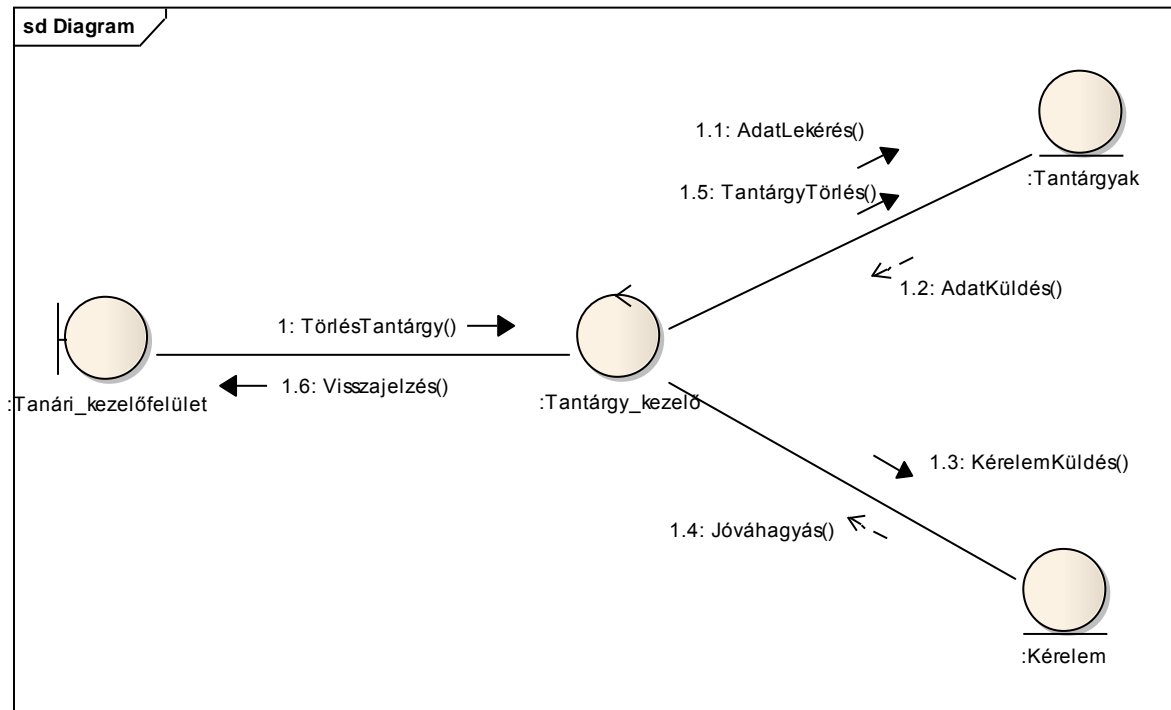
Elutasított működés



Az elutasítás után rögtön visszajelez a sikertelenségről és nem módosít az adatbázisban a kezelő.

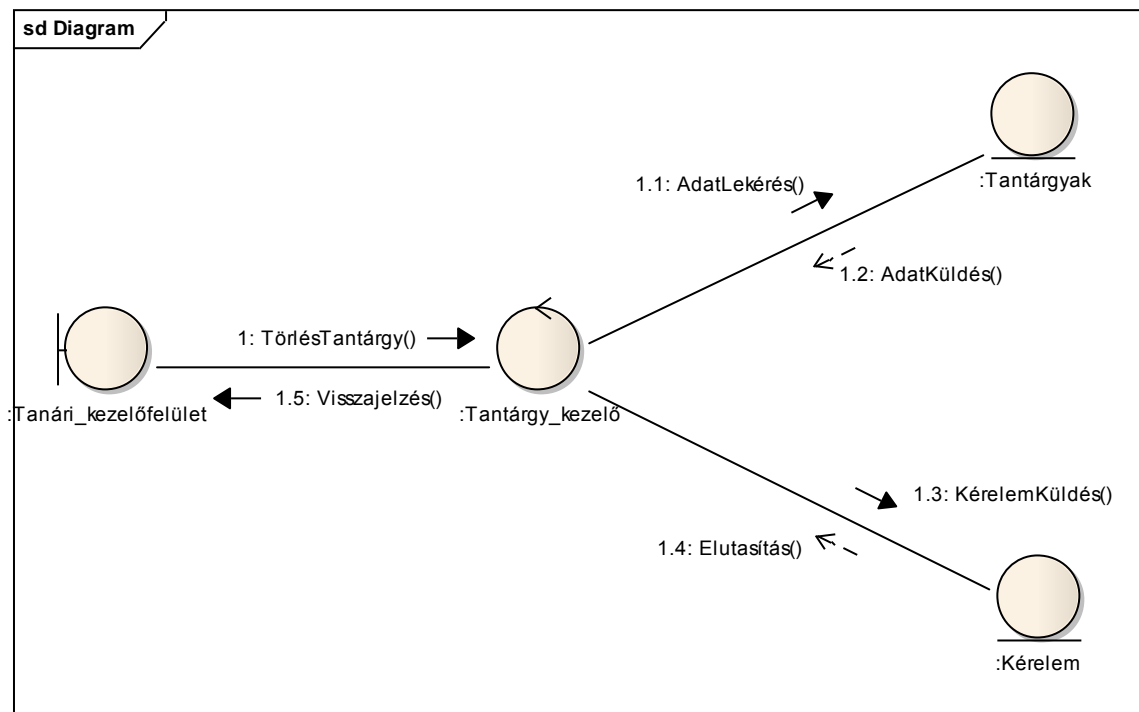
## Tanár Tantárgytörlés:

### Normál működés



A kezelő ellenőrzi a tárgy adatait, majd elküldi jóváhagyásra a vezetéshez. Ha elfogadják a kezelő törli a tárgyat az adatbázisból és visszajelez a tanárnak.

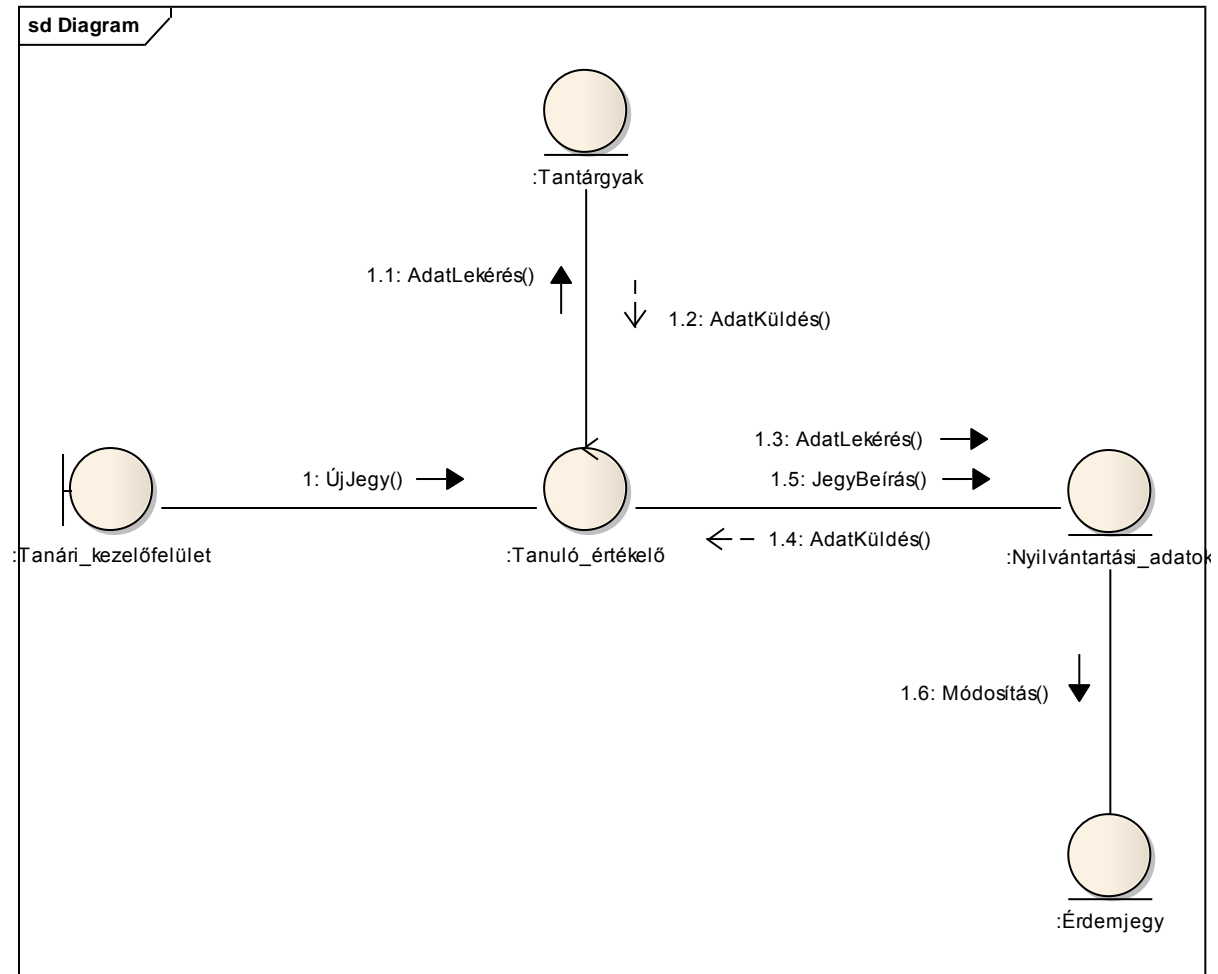
### Elutasított működés



Nem törlődik a tárgy, hanem rögtön visszajelez a kezelő elutasítás esetén.

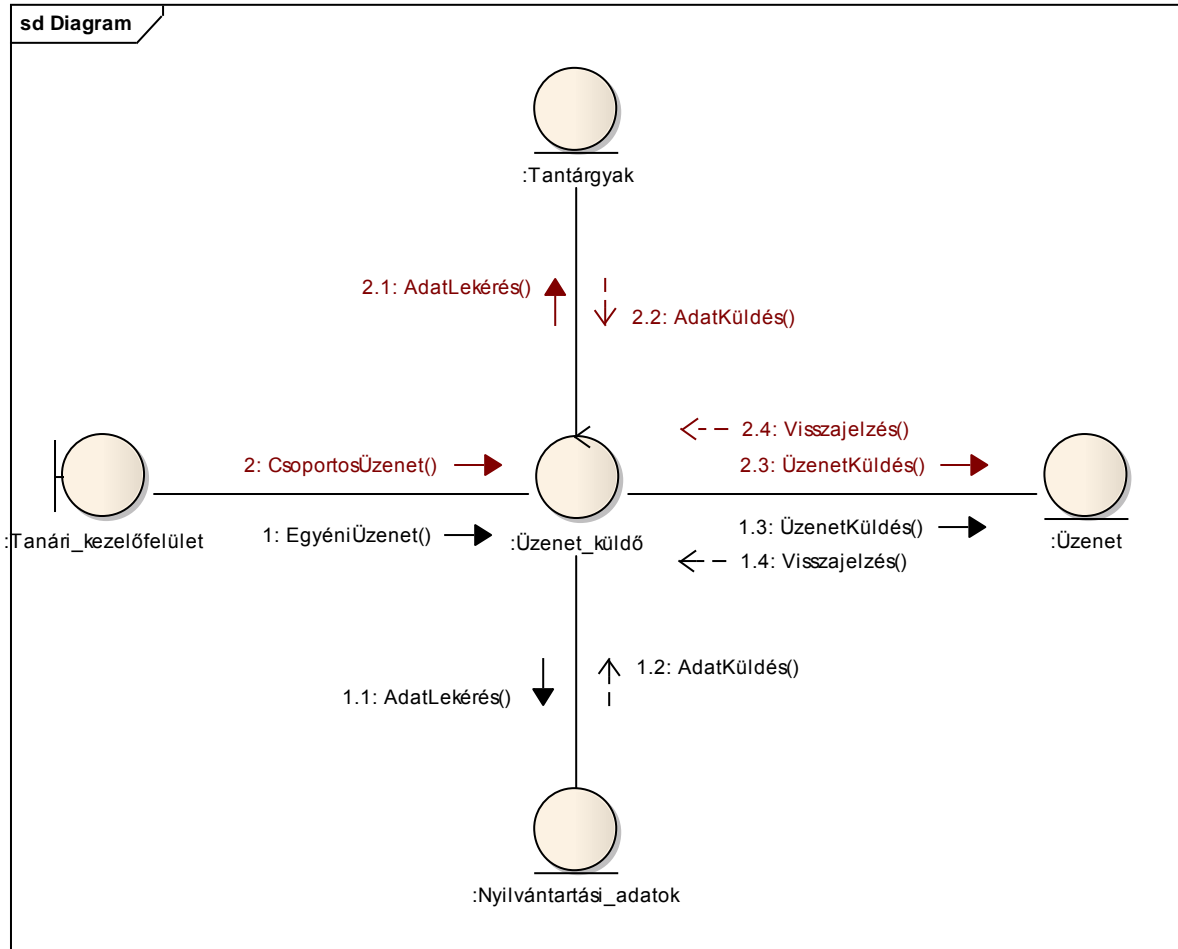


Tanár Tanuló értékelés:



A tanár elindítja az értékelési folyamatot, az értékelő listázza a tárgyakat és begyűjti a tanulók adatait. Ha be lett írva az érdemjegy az értékelő módosítja a tanuló tárgyhoz tartozó eredményét.

Tanár és Vezető Üzenet küldés:

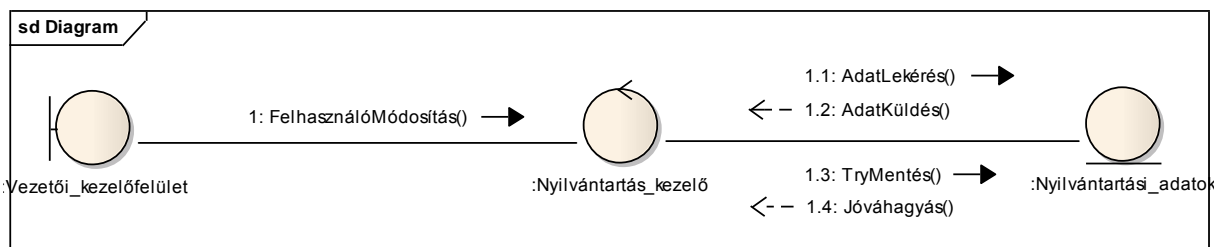


Egyéni üzenetküldésnél az Üzenet küldő lekérdezi a lehetséges címzettek listáját, az üzenet adatbázisban rögzíti hozzájuk a kívánt tartalmat, majd visszajelez.

Csoportos küldésnél a tantárgyakra járó hallgatók szerint szűr a kezelő, hasonló mód rögzíti a kívánt szöveget az üzenet adatbázisban és visszajelez.

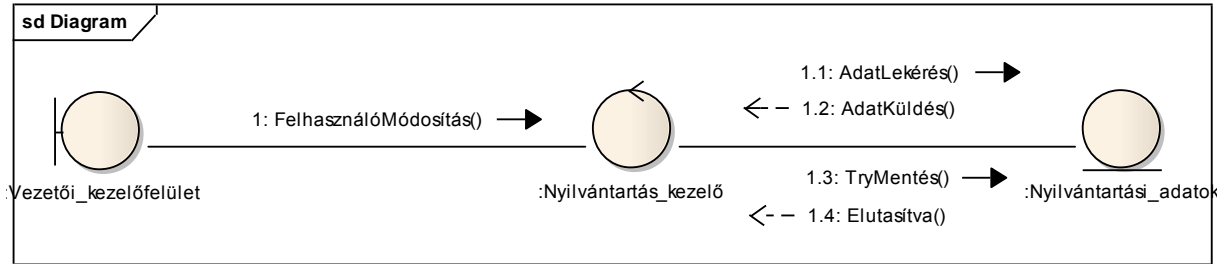
Vezető felhasználó módosítása:

Normál működés



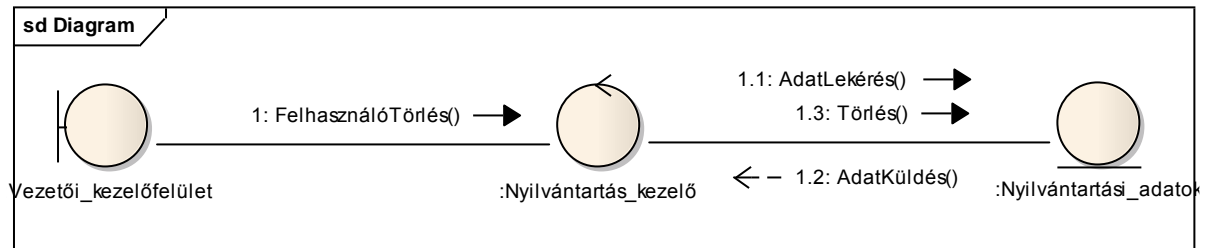
A nyilvántartás kezelő ellenőrzi a nyilvántartást, visszaküldi listázásra, ha a célszemély jóváhagyja a módosítást a kezelő módosítja az adatbázist.

## Elutasított működés



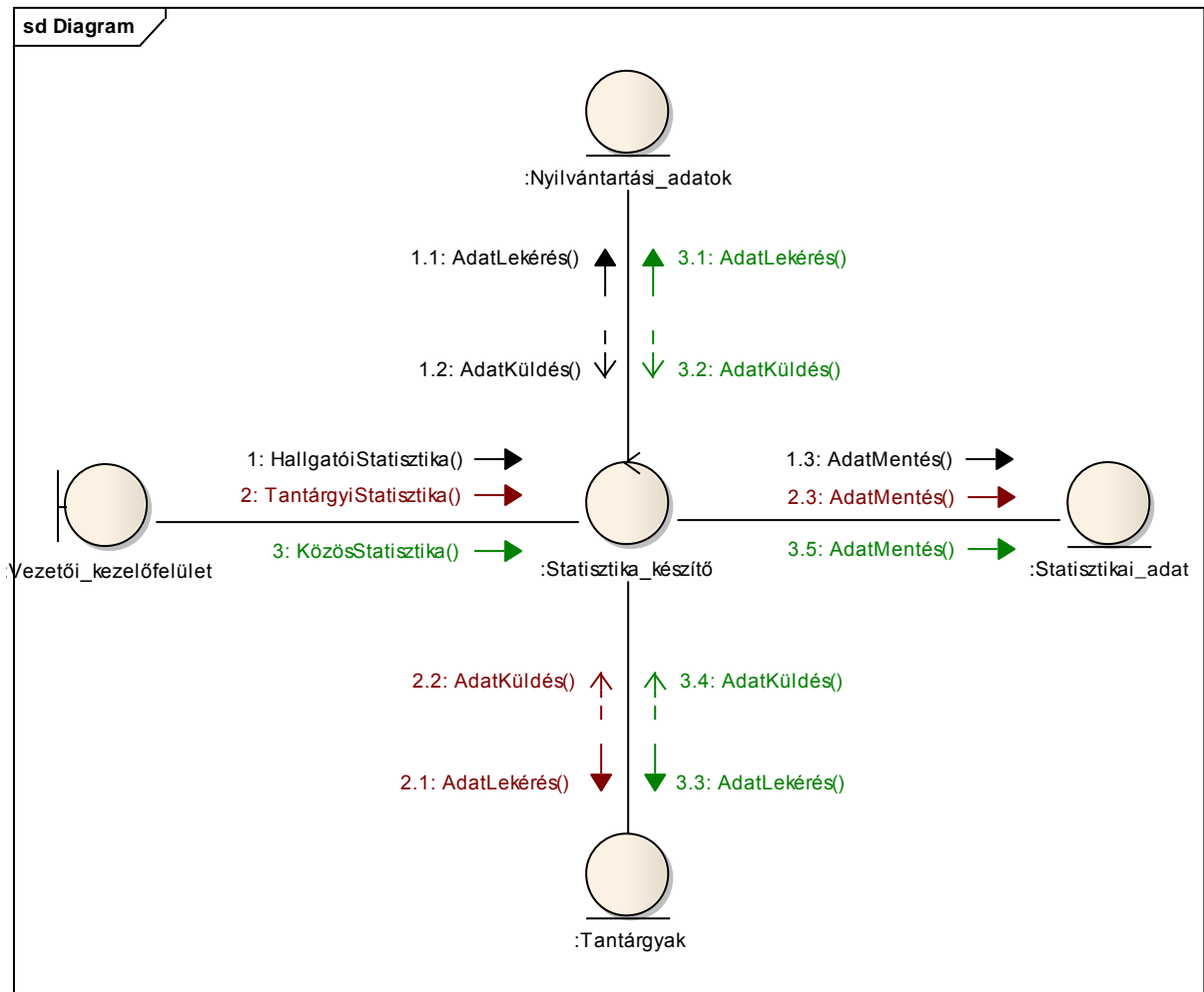
Ha a célszemély elutasítja a változtatást nem íródik át az adatbázis.

## Vezető felhasználó törlés:



A kezelő listázza az felhasználókat, majd törli kívánt felhasználót.

## Vezető kimutatás készítés:

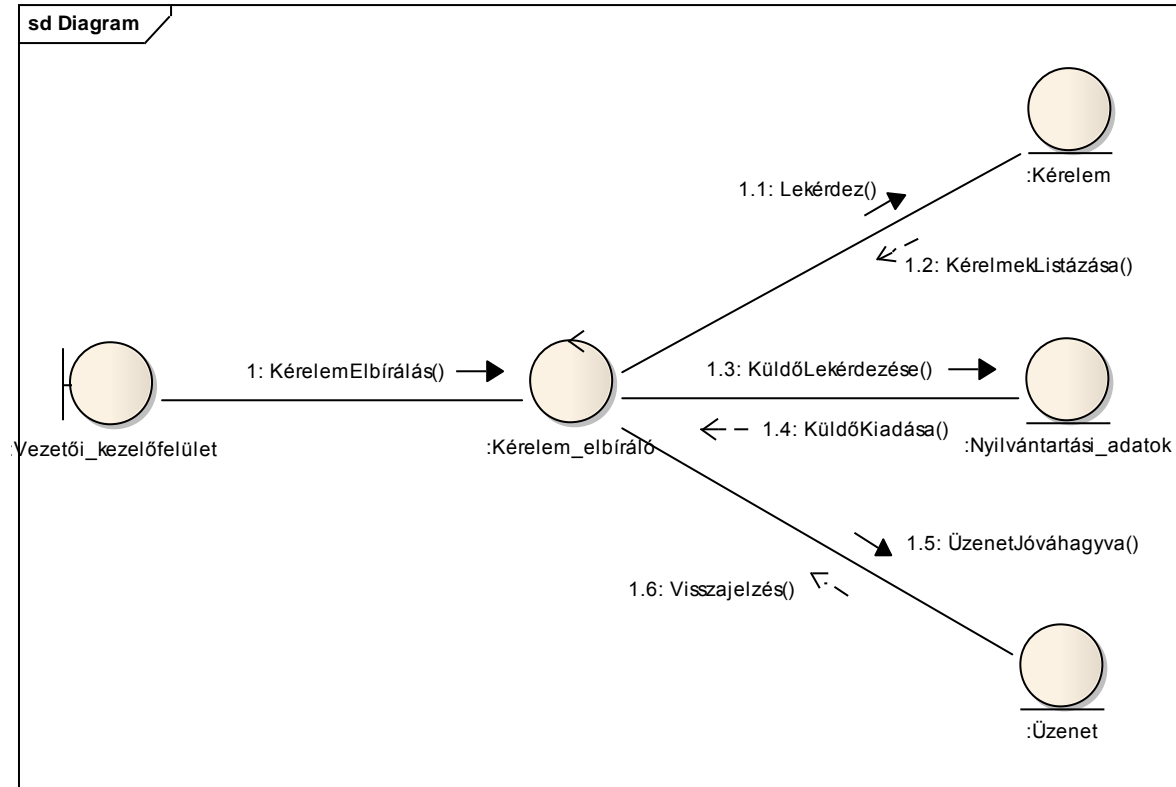


Statiztika készítésekor a hallgatói statiztikákhoz a kezelő lekérdezi a nyilvántartást, elküldi az adatokat egy közös statiztika tárolóba, a tantárgyak adatait szintén lekérdezi és menti a

közös tárolóba. A folyamatok külön-külön és együttesen is működnek, attól függően, hogy miket akarnak kimutatni.

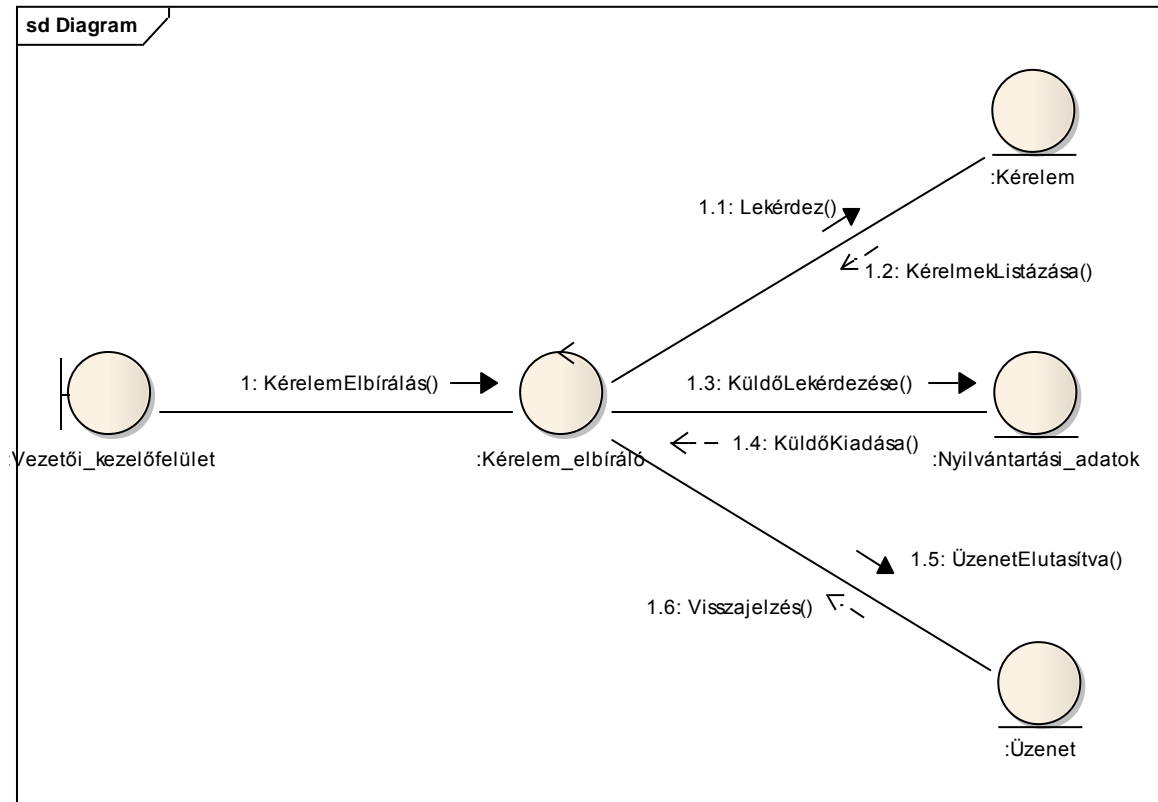
## Vezetői kérelem elbírálás:

Normál eset



Az elbíráló lekérdezi a Kérelmek adatbázis, hogy van-e döntésre váró kérelem, kilistázza, kiolvassa a nyilvántartásból a kérelmező adatait, majd az elbíráló a jóváhagyást elküldi a kérelmezőnek, végül a vezető is kap egy visszaigazolást a sikerességről.

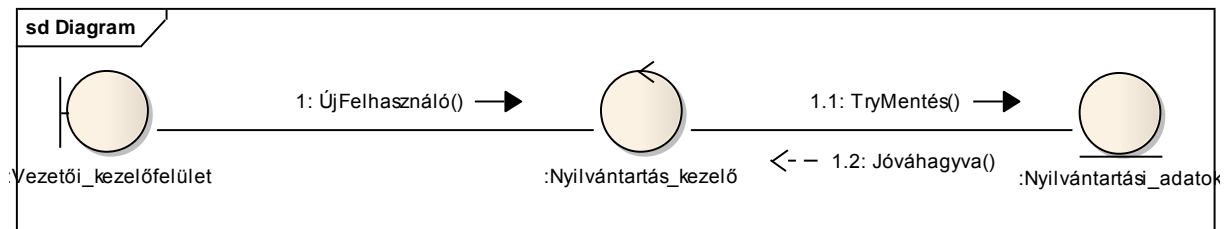
Elutasított eset



Elutasítás esetén az elbíráló az elutasítást küldi vissza a kérelmező, végül ő is kap róla visszaigazolást.

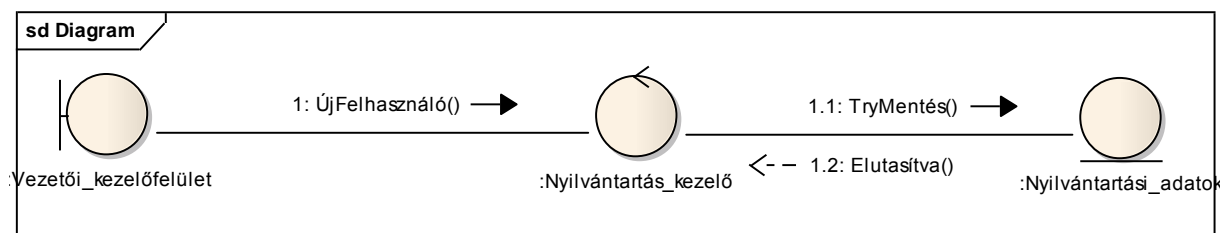
Vezető Új felhasználó létrehozása:

Normál eset



A vezető megpróbál létrehozni egy új felhasználót, a nyilvántartás kezelő, pedig megpróbálja beírni az adatbázisba. Sikeres művelet esetén visszajelzi ezt a vezetőnek.

Elutasított eset



Íráskor valami hiba lép fel (pl van már egy ilyen nevű ember), nem íródik ki az adatbázisba az új felhasználó, a hiba továbbítódik a vezetőnek.

### **3. Melléklet Tesztesetek**

#### **Teszteset 1**

XML létrehozás

```
public void xmltest()
{
    if (!File.Exists("Hallgató.xml"))
    {
        XElement hallgatók = new XElement("Hallgatók");
        hallgatók.Save("Hallgató.xml");
    }
}
```

Elvárt működés:

Ha nem létezik a hallgató.xml, létrehozza és beleteszi <Hallgatók> keretet

Eredmény:

Teljesíti az elvárt működést.

#### **Teszteset 2**

Osztály objektum mentése XML-be

```
XDocument doc = XDocument.Load("Hallgató.xml");
XElement hallgató = new XElement(Hallgatók,
    new XElement("Hallgató",
        new XAttribute("Zeuszkód", újHallgató.Zeuszkód),
        new XElement("Név", újHallgató.Név),
        new XElement("Lakhely", újHallgató.Lakhely),
        new XElement("SzemélyIgazolványSzám", újHallgató.SzemélyIgsz),
        new XElement("SzületésiDátum", újHallgató.SzületésiDátum),
        new XElement("SzületésiHely", újHallgató.SzületésiHely),
        new XElement("FelvettTárgyak", újHallgató.FelvettTárgyak),
        new XElement("Aktív", újHallgató.Aktív),
        new XElement("Végzett", újHallgató.Végzett),
        new XElement("Jelszó", újHallgató.Jelszó)));
doc.Save("Hallgató.xml");
```

Elvárt működés:

Létrehozza a Hallgatók kereten belül a Hallgató node-ot és a tulajdonságokat.

Eredmény:

Az elvárt működést hozza, ha csak egy darab objektumot akarunk kimenteni. ha újra meghívjuk felülíródik az xml tartalma.

Megoldás:

A „new XElement(„Hallgatók,„) keretet ki kell venni és a végén  
doc.Element(„Hallgatók“).Add(hallgató) formában hozzáfűzni az xml tartalomhoz, majd  
menteni.

## Teszteset 3

Azonosító generálás

```
string myZeusz = "";  
Random rnd = new Random(); //véletlen az ascii generáláshoz  
  
for (int i = 0; i < 6; i++)  
{  
    int j=rnd.Next(0, 2);  
    if (j == 0)  
    {  
        myZeusz += rnd.Next(0, 9);  
    }  
    else  
    {  
        myZeusz += Convert.ToChar(rnd.Next(65, 91));  
    }  
}
```

Elvárt működés:

Véletlenszerűen generálunk egy 6 jegyű kódot amiben 0-9 és A-Z közötti értékek szerepelnek.

Eredmény:

A kívánt formátumú kód létrehozása sikeres.

## Teszteset 4

Vezetői jelszómódosítás

```
foreach (Vezető v in beolvasottVezetők)  
{  
    if (v.Zeuszkód == személy.Zeuszkód)  
    {  
        v.Jelszó = újjelszó;  
  
        XDocument doc = XDocument.Load("Vezető.xml");  
        var vezetők = from x in doc.Descendants("Vezető")  
                       where x.Element("Zeuszkód").Value ==  
személy.Zeuszkód  
                       select x;  
        foreach (XElement vezetőadat in vezetők)  
        {  
            vezetőadat.SetElementValue("Jelszó", v.Jelszó);  
        }  
    }  
}
```

```
        doc.Save("Vezető.xml");  
        break;  
    }  
}
```

Elvárt működés:

A beolvasott listából kikeressük kívánt vezető azonosítóját, módosítjuk a jelszavát, majd mindezt ki is iratjuk xmlbe.

Eredmény:

A jelszó módosul, de a kiírás hibára fut.

Megoldás:

Az alábbi kódot `x.Element("Zeuszkód").Value` át kellett írni az attribútumra `x.Attribute("Zeuszkód").Value`, így működött az xmlben történő átírás is.

### Teszteteset5

Formon kiválasztott tárgy adatainak betöltése

```
tárgy = tantárgyak.Find((Predicate<Tantárgy>)lsb_felveheto.SelectedItem);  
    txb_targynev.Text = tárgy.Tárgynév;  
    lbl_kod2.Text = tárgy.Tárgykód;  
    lbl_parose.Text = tárgy.Hét;  
    txb_terem.Text = tárgy.Helyszín;  
    txb_tanar.Text = tárgy.Oktatók[0];  
    txb_kezdes.Text = tárgy.KezdésÓra.ToString() + ":" +  
tárgy.KezdésPerc.ToString();  
    txb_befejezes.Text = tárgy.VégeÓra.ToString() + ":" +  
tárgy.VégePerc.ToString();  
    lbl_követelmeny.Text = tárgy.Követelmények;
```

Elvárt működés:

A listboxban kiválasztott tárgykód és tárgynévből kikeresi a létező tárgyak közül annak az objektumát, amelyikre kattintottunk, majd betölti az adatait a form felületére.

Eredmény:

`InvalidCastException` – nem tud stringet konvertálni tantárgyá

Megoldás:

Foreachre cserlétük a keresést és substringgel azonosítottuk a tárgykódot:

```
foreach (Tantárgy t in tantárgyak)  
{  
    if (t.Tárgykód ==  
lsb_felveheto.SelectedItem.ToString().Substring(0,6))  
    {  
        tárgy = t; break; }  
}
```



## **4. Mellékletek Jegyzőkönyvek**

### **JEGYZŐKÖNYV**

**Időpont:** 2013. március 8.

**Jelen vannak:**

- Dori Bence
- Schönberger Zsófia
- Balogh Tamás
- Esse Zsolt

**Események:**

A Szoftvertechnológia gyakorlat órán a használati eset és analízis modellekkel kapcsolatos ismereteinket bővítettük. Az óra végén a következő munkamegosztás került elfogadásra:

Dori Bence az eddig elkészült használati eset diagramokat bővíti, a többiek a kommunikációs diagramokat készítik el, mégpedig a következő felosztásban:

Schönberger Zsófia a Hallgatóhoz tartozót, Balogh Tamás a Tanárhoz tartozót, Esse Zsolt a Vezetőt.

---

Esse Zsolt  
jegyzőkönyvvezető

---

Dori Bence  
projektvezető

## **JEGYZŐKÖNYV**

**Időpont:** 2013. március 22.

**Jelen vannak:**

- Dori Bence
- Schönberger Zsófia
- Balogh Tamás
- Esse Zsolt

**Események:**

A Szoftvertechnológia gyakorlat órán a telepítési és tervezési modellekkel ismerkedtünk meg, majd ezek elkészítését gyakoroltuk az Enterprise Architect programban. Ezek után a csapaton belüli kommunikációs eszközök bővítésében egyeztünk meg és a projektfeladatokat osztottuk fel.

Dori Bence vállalta a telepítési modell elkészítését. A többiek az eddig elkészült modelleket vizsgálják át és megkezdik az osztálydiagramok tervezését.

---

Esse Zsolt  
jegyzőkönyvvezető

---

Dori Bence  
projektvezető

## **JEGYZŐKÖNYV**

**Időpont:** 2013. április 5.

**Jelen vannak:**

- Dori Bence
- Schönberger Zsófia
- Balogh Tamás
- Esse Zsolt

**Események:**

A Szoftvertchnológia gyakorlat órán átismételtük az szekvenciadiagramokkal kapcsolatos tudnivalókat, gyakoroltuk elkészítését CASE programban. Ezek után tisztáztuk a részletes osztálydiagramokkal kapcsolatos fontosabb kérdéseket és a projektben résztvevők között felosztottuk a feladatokat.

A következő munkamegosztásban egyeztünk meg: Schönberger Zsófia és Esse Zsolt elkészíti az alkalmazási réteg részletes osztálydiagramjait, Dori Bence pedig a köztes és a megjelenítési réteg diagramjait továbbá a telepítési diagramot. Balogh Tamásnak a szekvenciadiagramokat kell elkészíteni. A használt CASE szoftver továbbra is az Enterprise Architect.

---

Esse Zsolt  
jegyzőkönyvvezető

---

Dori Bence  
projektvezető

## **JEGYZŐKÖNYV**

**Időpont:** 2013. április 12.

**Jelen vannak:**

- Dori Bence
- Schönberger Zsófia
- Balogh Tamás
- Esse Zsolt

**Események:**

A gyakorlat során az állapotdiagramokról tanultunk, ezek létrehozását CASE programban is gyakoroltuk. A következőkben egyeztünk meg: Schönberger Zsófia és Esse Zsolt átdolgozza a részletes osztálydiagramokat a megbeszéltek alapján, erre építve Balogh Tamás a szekvencia diagramokat dolgozza ki. Dori Bence az állapotdiagramokat készíti el.

---

Esse Zsolt  
jegyzőkönyvvezető

---

Dori Bence  
projektvezető

## **JEGYZŐKÖNYV**

**Időpont:** 2013. április 19.

**Jelen vannak:**

- Dori Bence
- Schönberger Zsófia
- Balogh Tamás
- Esse Zsolt

**Események:**

A tervek leadása után elkezdődött magának a programnak a kódolása. Mivel négy nagyobb kezelőnk akad, valamint a hozzájuk tartozó interfészek, így négy felé lettek osztva. Dori Bence elkészíti az adat kezelőt és a hozzá tartozó interfészt, Esse Zsolt a tárgykezelőt és az ide tartozó interfészt, Schönberger Zsófia elkészíti az üzenet kezelést valamint annak interfészét, Balogh Tamás pedig a kérelemkezelőt és interfészét, valamint a grafikus felületet.

---

Esse Zsolt  
jegyzőkönyvvezető

---

Dori Bence  
projektvezető

## **JEGYZŐKÖNYV**

**Időpont:** 2013. április 26.

**Jelen vannak:**

- Dori Bence
- Schönberger Zsófia
- Balogh Tamás
- Esse Zsolt

**Események:**

A múlt héten megbeszélt kezelők még nem készültek el, a grafikus rész is lassan halad. Beláttuk, hogy ekkora program lekódolására a két hét kevés 4 emberes teammel. Kivettük az exrának tervezett kérelem kezelést és üzenet küldést a rendszerből, megbeszéltük, hogy a többire koncentrálnunk.

---

Esse Zsolt  
jegyzőkönyvvezető

---

Dori Bence  
projektvezető