

# Eseményvezérelt Alkalmazásfejlesztés

## 3. Beadandó

Bittner Barnabás

TC8TT8

2016. Május

# Contents

<b>1</b>	<b>Feladat</b>	<b>2</b>
<b>2</b>	<b>Elemzés</b>	<b>2</b>
2.1	Használati eset diagram . . . . .	2
<b>3</b>	<b>Adatbázis-terv</b>	<b>3</b>
<b>4</b>	<b>Rendszer-terv</b>	<b>4</b>
<b>5</b>	<b>További dokumentáció</b>	<b>4</b>

# 1 Feladat

Készítsünk programot, amellyel egy videotéka nyilvántartását valósíthatjuk meg az alábbi funkciókkal:

A programban listázhatóak a filmek adatai (cím, főszereplők, rendező, forgalmazó, megjelenés éve), és látható, hogy kölcsönözve vannak-e, vagy sem. Amennyiben egy film bent van, akkor zárolható (és a zárolás visszaállítható), ekkor nem kölcsönözhető. A lista szűkíthető a kikölcsönözhető, illetve a kikölcsönzött filmekre. Lehetőségünk van új filmet felvenni, módosítani, illetve törölni (amennyiben nincs kikölcsönözve).

Film kölcsönzéséhez válasszuk ki a tagot, illetve a filmet. Csak akkor lehet kölcsönözni, ha bent van a film, és nem zárolt. A kölcsönzés dátumát automatikusan rögzítsük.

A kikölcsönzött filmeket lehessen visszahozni. A visszahozatal dátumát automatikusan rögzítsük. Mivel a kölcsönzés ideje 24 óra, jelezzük, ha a tag késett (írjuk ki a késés idejét napban, illetve órában), és ezért pótdíjat kell fizetnie.

Az adatbázis az alábbi adatokat tárolja:

tagok (név, cím, telefonszám); filmek (cím, főszereplők, rendező, forgalmazó, megjelenés éve, zárolt-e) kölcsönzések (tag, film, kölcsönzés ideje, visszahozatal ideje).

## 2 Elemzés

1. Az alkalmazáshoz felépítünk egy MySQL adatbázist, amely tárolja a filmeket, a tagokat és a kikölcsönzéseket.
2. Az alkalmazásban két képernyőn dolgozhatunk, az egyiken a filmeket módosíthatjuk a másikon a rendeléseket.
3. A filmeket módosító panelen a filmek minden adatát tudjuk interaktívan módosítani. Tudunk szűrni a kikölcsönözhető és a már kikölcsönzött filmekre is. A kikölcsönözhető filmeket tudjuk törölni. Új filmet is tudunk az adatbázisba felvinni. A módosítások csakis a submit gomb megnyomása után kerülnek az adatbázisba. Minden módosítás és submit gombra kattintás között bármikor visszavonhatjuk az eddig módosítottakat a revert gombbal.
4. A kölcsönzéseket módosító fülön lehetőségünk van a kölcsönzéseket visszahozni, ekkor automatikusan a mai dátum kerül rögzítésre, ha ez egy napnál nagyobb idő akkor a rendszer kiírja az ügyfél számára fizetendő összeget.

### 2.1 Használati eset diagram

Figure 1: A feladat használati eset diagramja

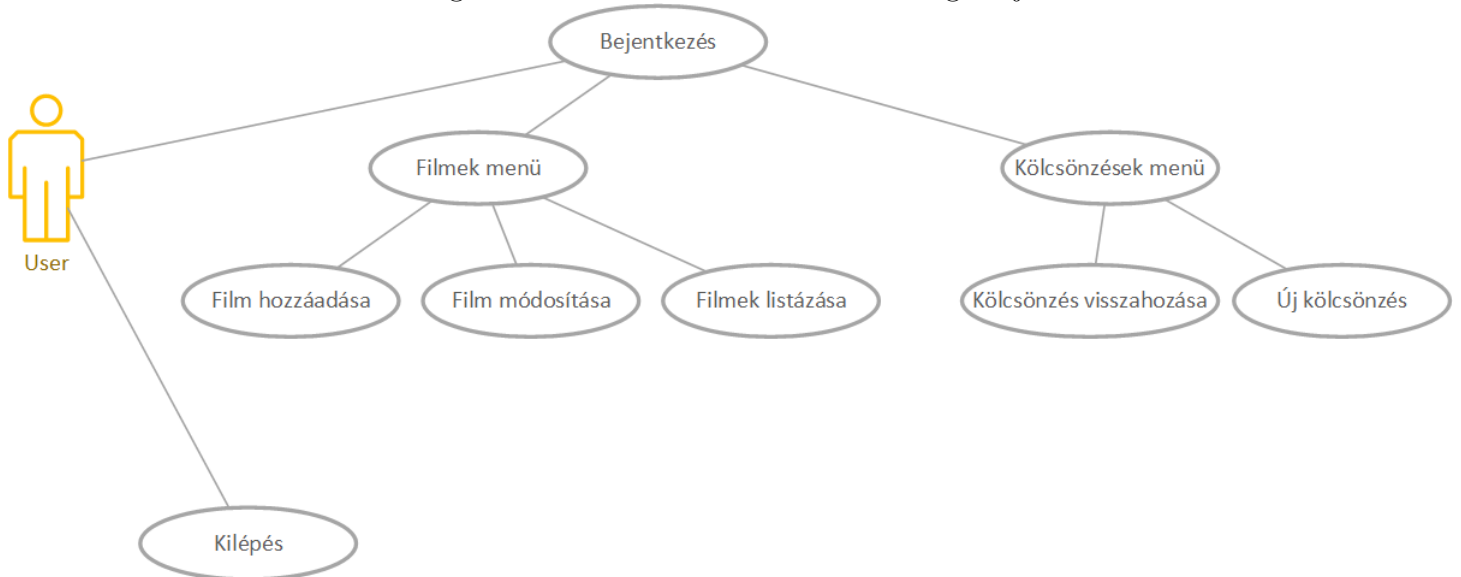
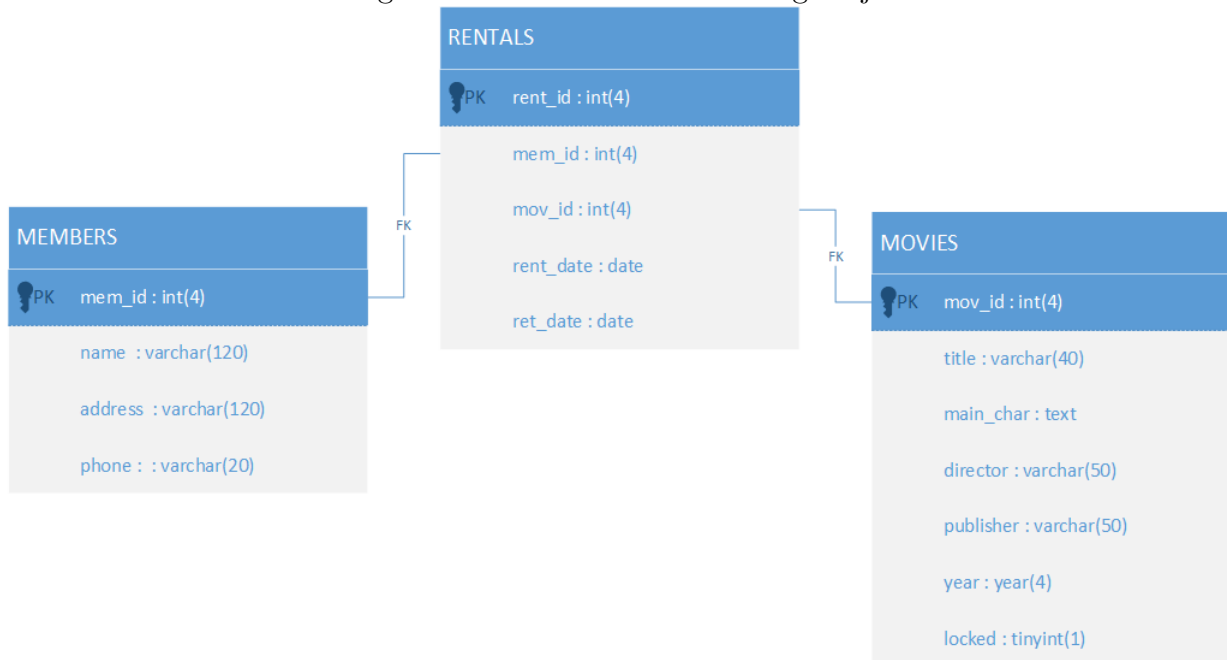


Figure 2: A feladat adatbázisdiagramja

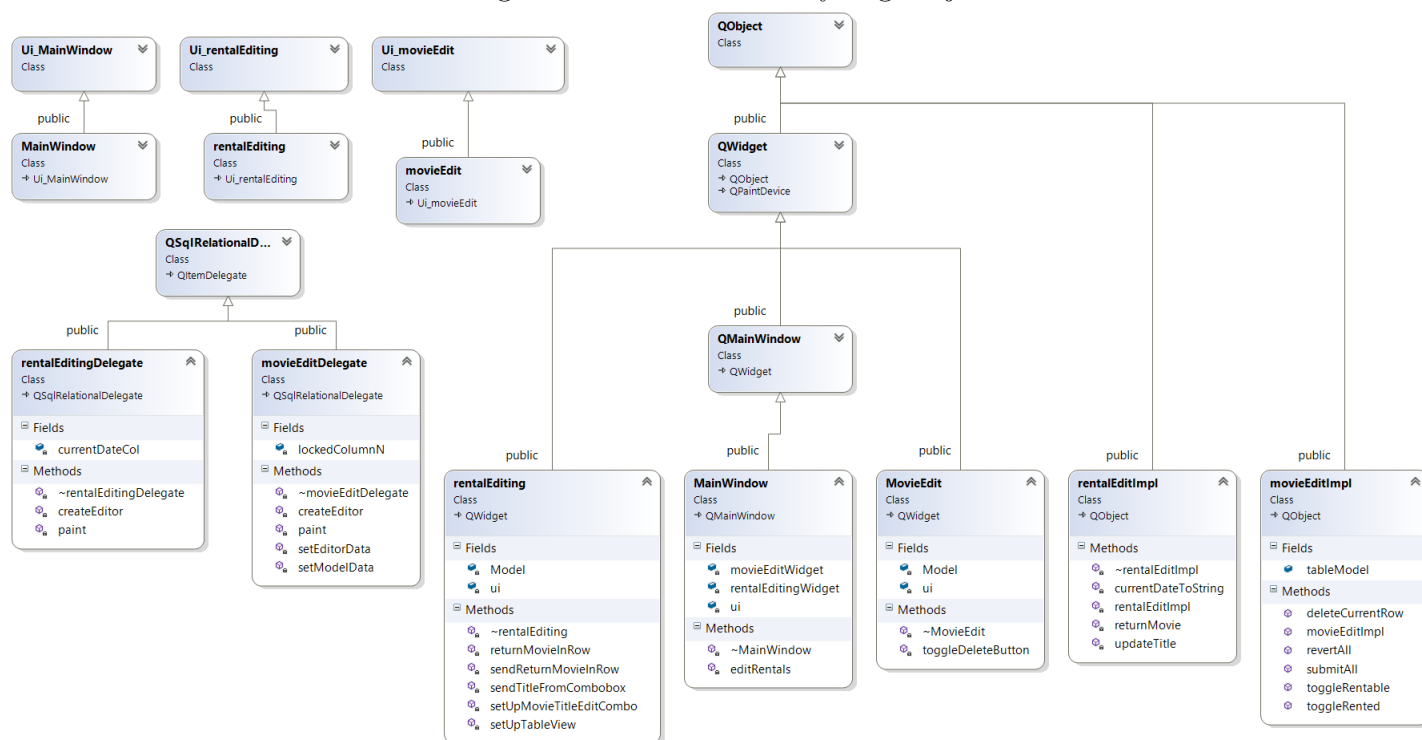


### 3 Adatbázisinterv

Az adatbázis felépítése (Videobase) az alábbi:

1. **Tagok:** tartalmazza a tagok adatait, **azonosítóját**, a **címüket**, **nevüket**, **telefonszámukat**
2. **Filmek:** tartalmazza a filmek adatait, **azonosítóját**, **címet**, **főszereplők listáját**, **rendezőjét**, **kiadóját** és **megjelenés évét**
3. **Kölcsönzések:** Összeköti a filmek és tagok listáját, tartalmazza a kölcsönzött **film azonosítóját**, kölcsönző **tag azonosítóját**, **kölcsönzés napját** és **visszahozatal napját**, ha vissza lett hozva.

Figure 3: A feladat osztálydiagramja



## 4 Rendszerterv

A program szerkezetét a Qt modell/nézet architektúrának megfelelően építjük fel, azaz az adatbázis kezelésére táblamodell osztályokat alkalmazunk, amelyek tartalmát megfelelő táblamegjelentőkkel és egyéb vezérlőkkel jelenítjük meg. A beépített architektúrát több ponton kiegészítjük, ahol egyedi viselkedésre/megjelenésre van szükségünk, továbbá megvalósítjuk a megjelenítő ablakokat

1. A főprogram betöltésénél egyelőre nincs szükség bejelentkezni, az automatikus. Ha nem jött létre adatbázis kapcsolat **QMessageBox** figyelmeztet.
2. A sikeres adatbázis kapcsolatot követően a programban a menüt használva két opció közül választhatunk. Ezt akciókkal ellátott menüsor biztosítja. Amikor rákattintunk az egyik menüre a **QMainWindow** példányosít egy a menühöz tartozó **QWdiget**-et.
3. A **MovieEditing** widget biztosítja a filmek megjelenítését egy **QTableView** végzi, mely egy **QSqlRelationalDatabaseModel** osztálytól kapja az megjelenítendő adatokat. Az adatok szerkesztéséhez és megjelenítéséhez egy **QSqlRelationalItemDelegate** osztályból származó saját definiálású osztály gondoskodik.

A filmeket **QPushButton**-okkal és **QCheckBox**-okkal lehet a filmeket rendezni, listázni. Az éppen nem kölcsönzött könyveket lehet törölni. Minden változtatás csak a Submit gomb megnyomása után válik permanenssé.

4. A **RentalEditing** widget gondoskodik a kölcsönzött filmek megjelenítéséről és az új kölcsönzések felvételéről. Az adatokat egy **QTableView** objektum jeleníti meg mely egy **QSqlRelationalDatabaseModel** osztálytól kapja az megjelenítendő adatokat. Lehet visszahozni egy filmet. Ha a kölcsönzés több mint egy napig tartott akkor egy **QLabel**-ben jelzi, hogy mennyit kell fizetni a tagnak.

## 5 További dokumentáció

A programot mélységében jobban leíró, automatikusan generált dokumentáció megtalálható itt: [Doxygen dokumentáció](#)