**Neumann János Egyetem**

**Dinh Tuan Hoang**

**-T8SKHC-**

**Git Repository address:**

https://github.com/Blacka61/Java\_beadando

**Contents:**

**1. DETAIL TASK DESCRIPTION**

**2. MACHINE AND LANGUAGE REQUIREMENTS**

2.1. RECOMMENDED SPECIFICATIONS

**3. FOLDER AND FILE STRUCTURE BY NAME AND FUNCTION**

3.1. GITHUB REPOSITORY CONTENTS:

3.2. JAVA CLASSES

**4. SCREENSHOTS**

4.1. MAINBOARD

4.2. Reader1

4.3. Reader2

4.4. WRITTING

4.5. MODIFICATION

4.6. DELETION

1. Detail Task Description

Java alkalmazások Alapképzés Előadás Beadandó feladat

A feladathoz segítséget a feladat végénél talál!

Alakítsanak ki két-fős csoportokat és a feladatot közösen, projektmunkában oldják meg. A dokumentációban majd írják le, hogy a két főből ki melyik feladatrészt csinálta meg.

A csoportoknál a kapcsolatfelvételt és a közös munkát a Teams-en meg lehet oldani.

Ha valaki nem talál társat (pl. páratlanul vannak a csoportban), a GitHub-os feladatot akkor is projektmunkában csinálja meg két GitHub fiókot felhasználva.

Az Adatbázisok mappában találnak adatbázis forrásokat, amik közül kell egyet a csoportnak választani és felhasználni a feladatában. Egy adatbázist maximum 1 csoport választhat.

Az adatbázist lehet újabb táblákkal bővíteni. Importálja be a választott adatbázis adatait.

A feladatban egy menüvel ellátott grafikus asztali alkalmazást kell készíteni Java nyelven JavaFX technológiával.

Összesen 30 pont

1. Adatbázis menü: egy CRUD alkalmazás a választott adatbázissal​​​​(5 pont)

Olvas almenü: Megjeleníi az adatbázis adatait egy táblázatban. Az adatbázis legalább 3 táblájának adatait használja fel. Ha túl sok adatot tartalmaz az adatbázis, akkor elég csak egy részét megjeleníteni.

Olvas2 almenü: Hasonló mint az előző, de az oldalon először jelenjen meg egy Űrlap, aminek segítségével tud szűrni az adatbázis adataira. Az űrlapon használja a következő elemeket: szöveges beviteli mező, lenyíló lista, radio gomb, jelölőnégyzet.

Ír almenü: Az oldalon jelenjen meg egy űrlap, aminek segítségével új rekordot tud hozzáfűzni az adatbázis egyik táblájához.

Módosít almenü: Az oldalon jelenjen meg egy űrlap, aminek segítségével egy már meglévő rekordot tud módosítani az adatbázis egyik táblájánál. A rekord azonosítója lenyíló listából legyen választható.

Töröl almenü: Az egyik tábla egy rekordjának törléséhez. A rekord azonosítója lenyíló listából legyen választható.

1. Rest1 menü: Készítsen egy Restful klienst az eladáson bemutatott ​​​(5 pont)

https://gorest.co.in Rest szerverhez. A Create, Read, Update, Delete almenükben valósítsa meg a hozzájuk tartozó műveletet.

1. Készítsen a Microsoft Azure felhőben egy ReatFul szerver szolgáltatást a választott ​(5 pont)

adatbázis egyik táblája alapján.

Rest2 menü: Készítsen egy Restful klienst a Microsoft Azure felhőben létrehozott Rest szerverhez. A Create, Read, Update, Delete almenükben valósítsa meg a hozzájuk tartozó műveletet.

1. SoapKliens menü: Készítsen egy SOAP klienst a Magyar Nemzeti Bank ​​(5 pont)

webszolgáltatásához.

Letöltés almenü: Letölti az összes adatot helyi adatbázisba.

Letöltés2 almenü: Az oldalon megjelenik egy űrlap (beviteli mezők, lenyíló lista, radio gomb, jelölő négyzet). Letölti a kiválasztott adatokat helyi adatbázisba és a Microsoft Azure felhő adatbázisába is.

Grafikon almenü: Letöltés2 almenühöz hasonlóan választ egy adatsort és megjeleníti az adatokat grafikonon is.

Minta: https://www.mnb.hu/arfolyam-lekerdezes

1. Adatbányászat menü: ​​​​​​​​(5 pont)

Válasszanak ki egy fájlt az Adatbányászat-források mappából (az órán vizsgát credit-g.arff fájlt nem lehet választani). Válasszanak ki ebben egy olyan attribútumot, ami nem sok fajta adatot tartalmaz. Általában az utolsó Class nevű attribútum ilyen.

Döntési fa almenü: Készítsenek döntési fával gépi tanuló algoritmus a kiválasztott attribútum előrejelzésére. Eredményül írassák ki a Döntési fa.txt fájlba a következő adatokat:

Tanító és kiértékelő halmaz mérete,

TP, TN, FP, FN, Correctly Classified Instances, Incorrectly Classified Instances

A döntési fa kiíratása

A kiértékelő halmaz minden egyedére a prediktált és a valós érték

Több algoritmus almenü: A választott adathalmazra futtassa le a tanult algoritmusokat. A Gépi tanulás.txt fájlba írassa ki mindegyikre:

TP, TN, FP, FN, Correctly Classified Instances, Incorrectly Classified Instances

Válassza ki az algoritmusok közül azt, amelyiknek a legjobb lett a Correctly Classified Instances eredménye és írassa ki az algoritmus nevét az alkalmazás ablakába.

Több algoritmus almenü-2: Lenyíló listából lehessen választani a tanult algoritmusok közül. Az algoritmust lefuttatva írassa ki az alkalmazás ablakába a következő adatokat:

TP, TN, FP, FN, Correctly Classified Instances, Incorrectly Classified Instances

1. Egyéb menü:​​​​​​​​​​​(5 pont)

Páthuzamos almenü: Mutassa be a párhuzamos programvégrehajtást egy oldalon. pl. egy gombra való kattintás után egy Label-ben 1 másodpercenként, a másik Labelbe 2 másodpercenként jelenjen meg egy változó szöveg.

Stream almenü: Olvassa ki a válsztott adatbázis minden adatát egy Stream-be. Késztsen egy űrlapot (beviteli mezők, lenyíló lista, radio gomb, jelölő négyzet), ahol több szűrőfeltételt is beállíthat az adatokra. Jelenítse meg egy táblázatban a szűrt adatokat.

1. Készítsen egy JAR futtatható fájlt, ami a GitHub-on is megjelenik a főmappában.​

8. Használják a GitHub (github.com) verziókövető rendszert. ​​​

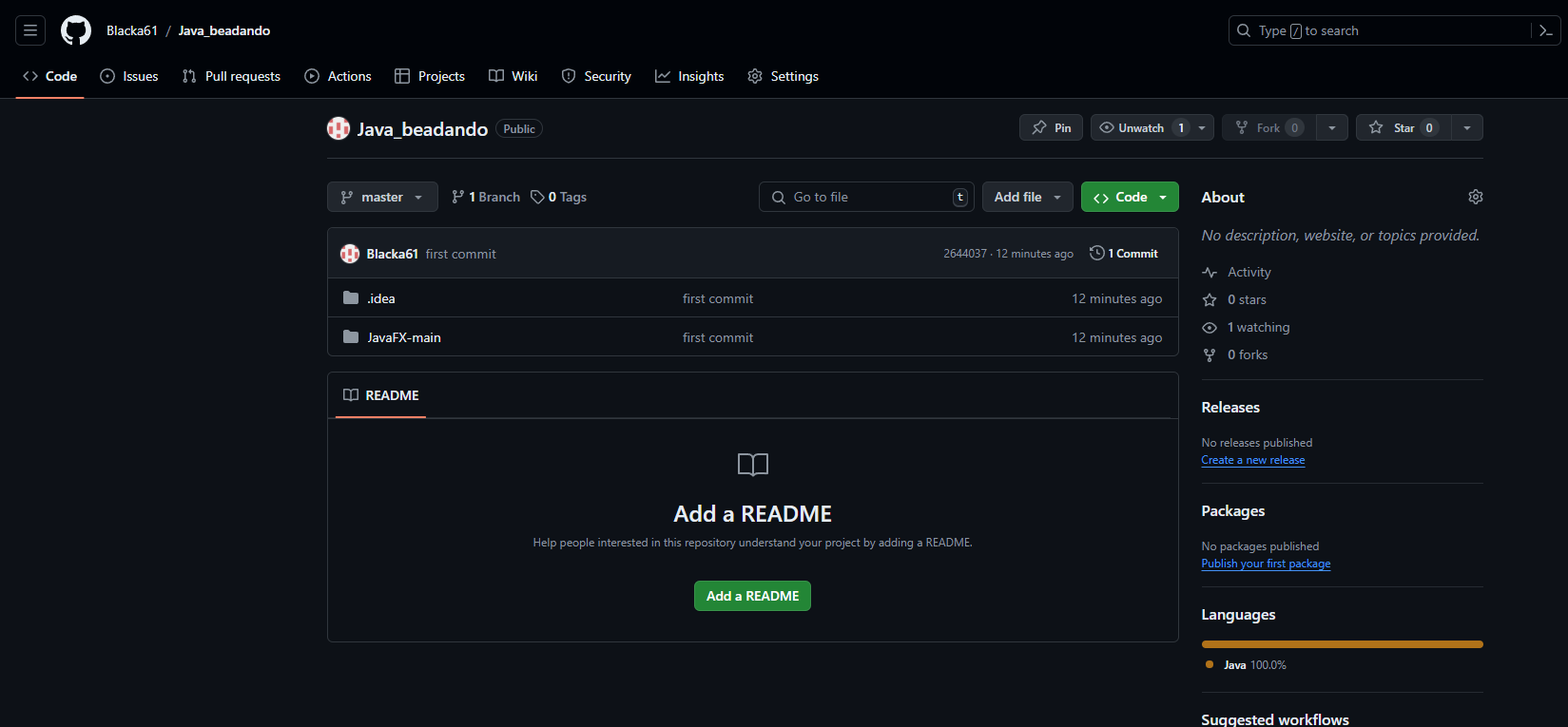
(Kötelező elem! A forrás ez alapján lesz javítva)

1. A GitHub-on a projektmunka módszert alkalmazzák: látszódjék, hogy a csoport tagjai melyik részt készítették el és kb. fele-fele arányban járuljanak hozzá a projekthez. Ne csak a kész alkalmazást töltsék fel egy lépésben, hanem a részállapotokat is még legalább 5 lépésben személyenként.
2. Machine and language requirements

The application was developed in the Java programming language. Thus, a java runtime environment must be installed to use the application.

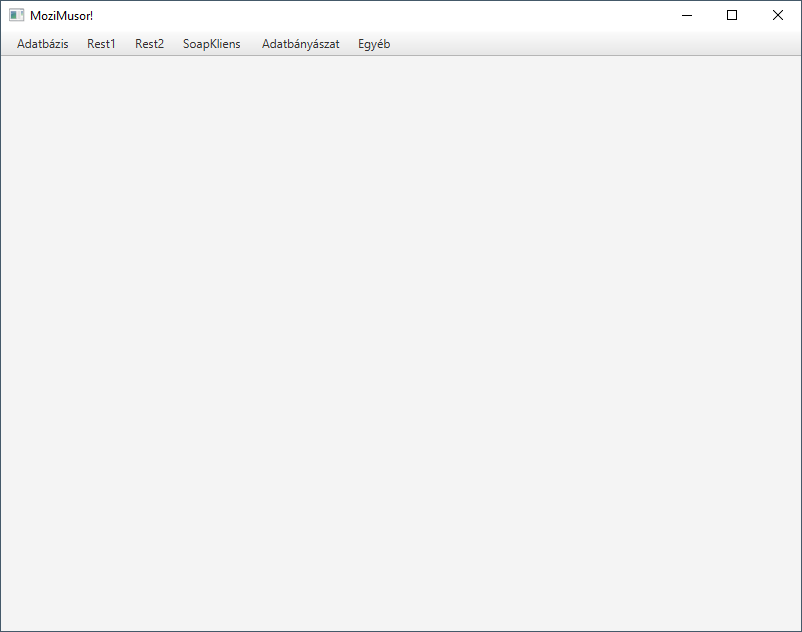
* 1. Recommended specifications:
* System with ARM or amd-64 architecture
* Operating system: Any OS. on which the java runtime environment can be installed.
* Network: Broadband Internet connection
* Storage: 2 GB free space.

1. Folder and file structure by name and function
   1. Github repository tartalma:

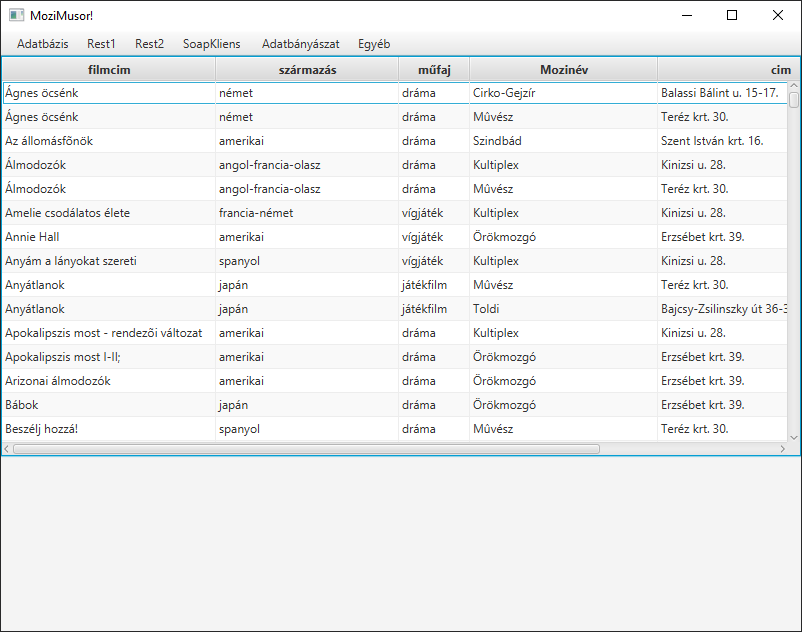


* JavaFx\_Beadando folder: The project directory
* Java beadando.docx: Documents
* mozimusor.sql: Data and tables belonging to the database of the application
  1. Java classes:
* Database: The model required for linked table queries.
* DatabaseDAO: Functions and data members for linked table query.
* HelloApplication: This contains the main function.
* HelloController: Contains the server-side logic, controls the application and selects which view to show the user.
* Mozi: The model required to query the cinema board.
* MoziDAO: Implements database operations related to the cinema board.
* RestKliens: Functions of communication with the rest server in task 2.
* RestUser: Model required for the rest server in task 2.
* Launcher: Helps translate the project into a jar file.

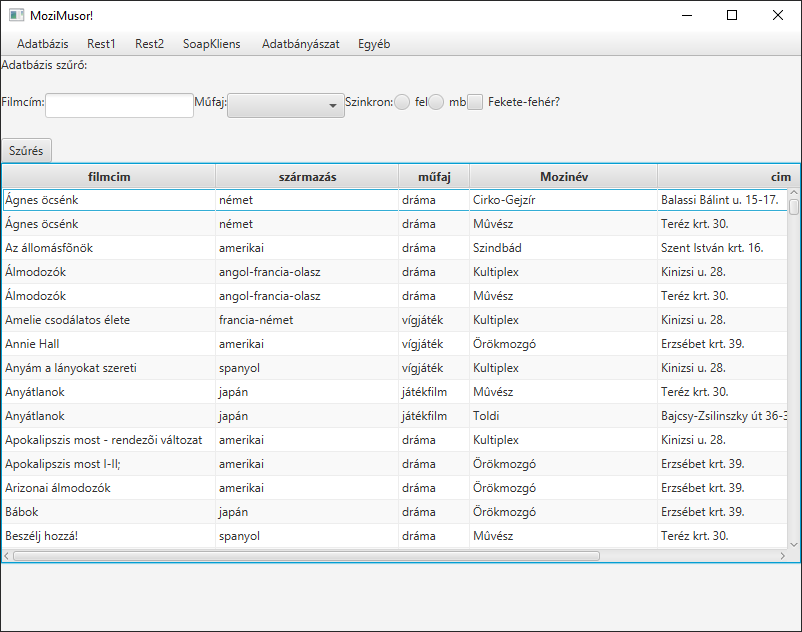
1. Screenshots
   1. Mainboard



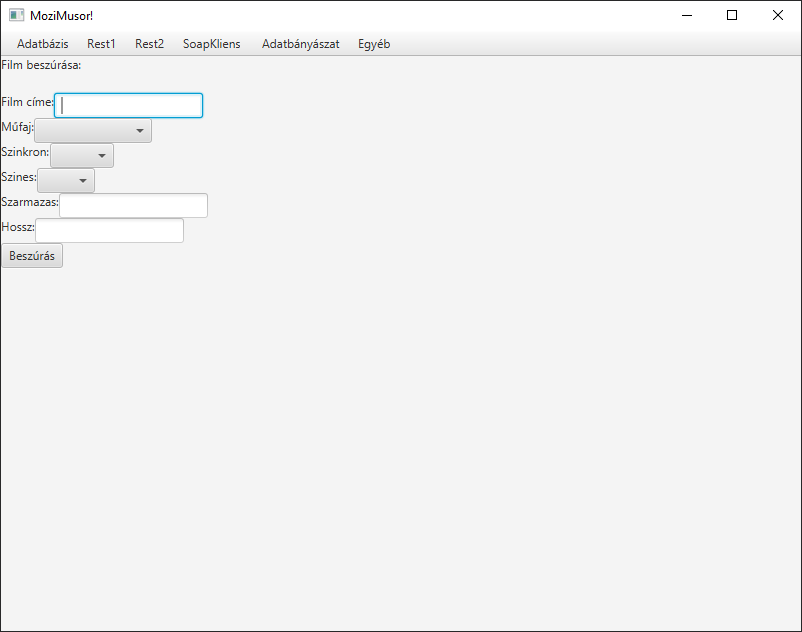
* 1. Reader1



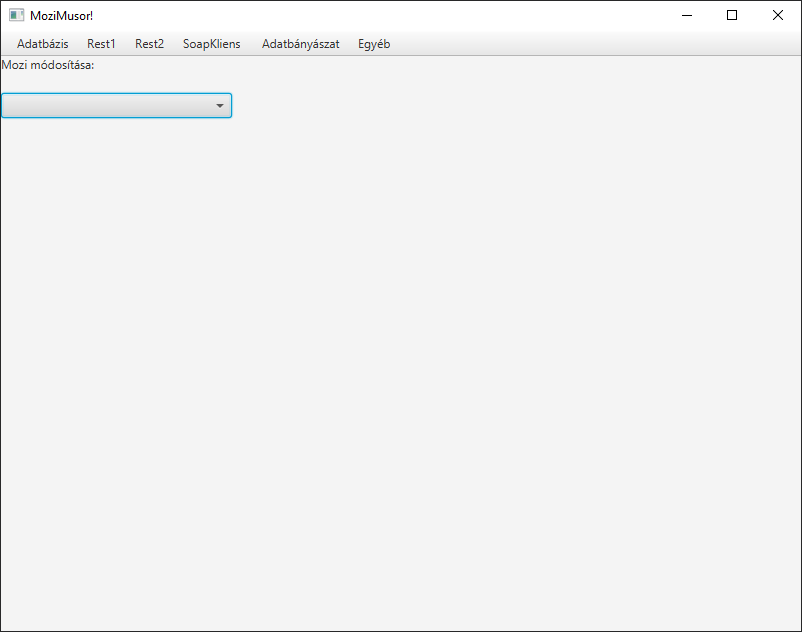
* 1. Reader2



* 1. Fill in information



* 1. Modification



* 1. Deletion

