

Математичка гимназија

Београд

Матурски рад из рачунарства и информатике

**АПЛИКАЦИЈА ЗА РЕЗЕРВАЦИЈУ КАРАТА И
УПРАВЉАЊЕ БИОСКОПСКИМ ПРОЈЕКЦИЈАМА**

Професор ментор:

Филип Хаџић

Ученик:

Лазар Савић 4б

Београд, мај 2021.

ПРЕДГОВОР

Сведоци смо дигитализације и технолошког развоја друштва. Рачунари су постали неизоставан сегмент свачијег живота, и баш та симбиоза оглашава марш ка новим променама и унапређењима. Управо се из такве интеракције родила идеја за настанак ове апликације.

Једна сасвим реална потреба човечанства јесте уштеда времена. У томе нам умногоме могу помоћи рачунари. Овај програм омогућава избегавање гужве, смањење дужине редова у биоскопским салама, као и време чекања за куповину карата и поруцбину хране.

Код за ову апликацију писан је у Јавином развојном окружењу NetBeans 8.2. Будући да је неопходна активна комуникација са базом података, коришћен је и DB Browser, алатка за прављење, одржавање и дизајнирање база компатибилних са SQLite стандардима.

Увод	4
1. УЧИТАВАЊЕ И СТАРТНИ ПРОЗОР АПЛИКАЦИЈЕ	5
2. ПРАВЉЕЊЕ НОВОГ НАЛОГА	6
3. АДМИНИСТРАТОРСКИ ИНТЕРФЕЈС	9
3.1. Преглед и манипулација подацима из базе података	9
3.2. Додавање нових филмова у базу података	11
3.3. Додавање нове хране у базу података	13
4. КОРИСНИЧКИ ИНТЕРФЕЈС	14
4.1. Заборављена лозинка	14
4.2. Регистрација и преглед филмова	15
4.3. Поручивање хране	18
4.4. Одабир седишта и завршетак резервације	19
4.5. Преузимање резервације	20
4.6. Уређивање профила	21
4.7. Слање повратних информација	21
5. АУТОМАТСКО АЖУРИРАЊЕ ПОДАТАКА	22
Закључак	24
Литература	25

Увод

Иако је простор за нове креативне идеје бесконачан, у овој апликацији биће обрађене главне и најважније компоненте једног биоскопског система за резервацију карата и хране. Овај програм је могуће користити на више начина, као администратор или као корисник.

****Администраторски интерфејс***

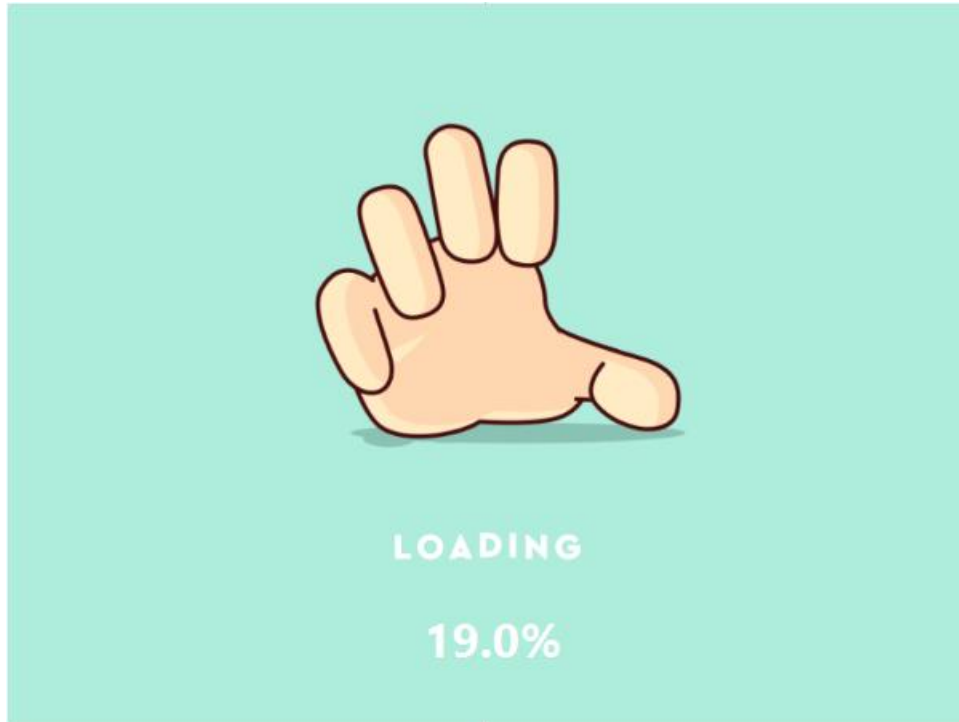
Након што се обаве неопходне провере, администратор може да се улогује и добије приступ различитим контролним компонентама. Омогућен је увид у списак резервација, филмова, хране, термина пројекција филмова и корисника. Администратор може да брише или ажурира постојеће податке, као и да у базу уноси нове филмове или храну која је у понуди. Као и администратори, и корисници могу да уређују свој профил.

****Кориснички интерфејс***

Након регистрације са својим корисничким именом и лозинком, кориснику се отварају различите могућности. Кликком на постер филма из одговарајућег жанра, отвара се нови прозор са подацима о изабраном филму и терминима пројекција тог филма. Могуће је и одгледати његов трејлер. Након што се одлучи за филм, од корисника се тражи да изабере одговарајућу пројекцију. Потребно је назначити број карата за децу и одрасле, а после тога је могуће и наручити храну. Уколико се одлучи за овај корак, потребно је изабрати храну и додати је у корпу. Током овог процеса, увек се може проверити тренутно стање на рачуну. Следећи корак је одабир слободних седишта. Последња етапа је преузимање PDF документа у којем су сумирани сви подаци за креирану резервацију. Од корисника се очекује да тај документ одштампа и понесе са собом приликом доласка на пројекцију филма, или да га покаже на свом уређају. Осим ствари уско везаних за прављење резервације, корисник може да уреди свој профил и да пошаље *feedback*, дакле утиске о раду биоскопа или функционалности апликације.

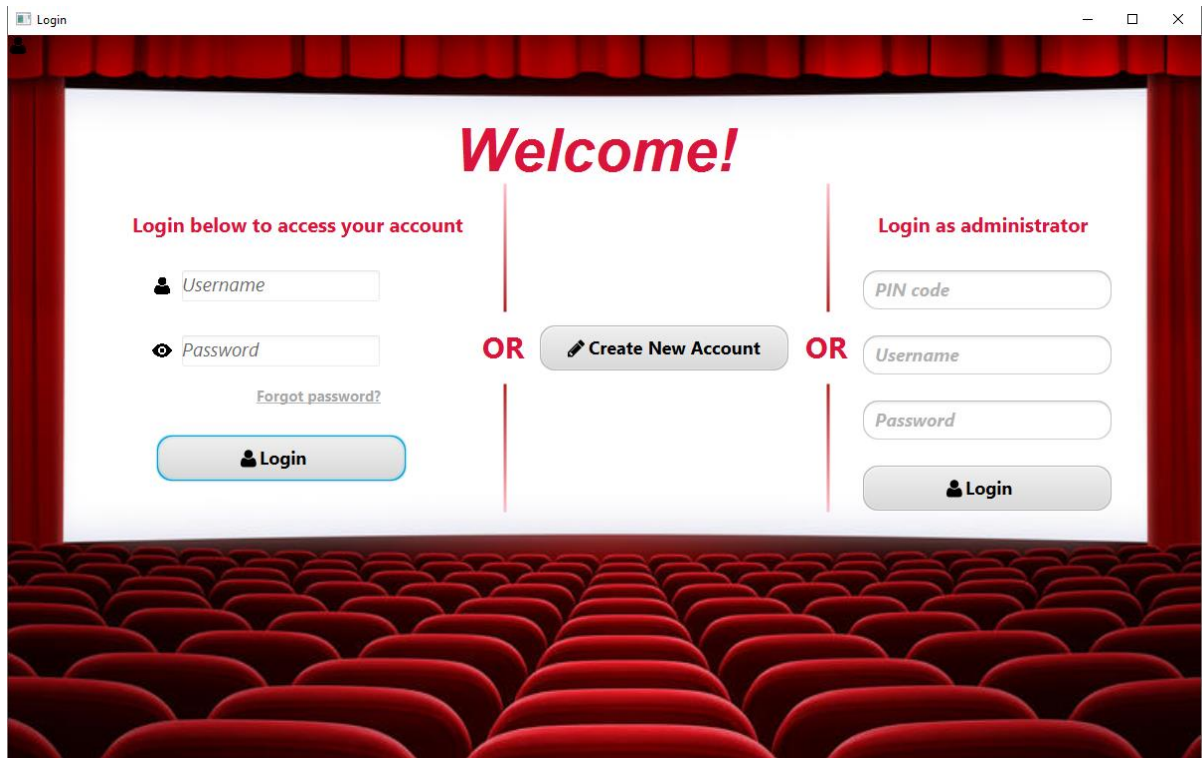
1. УЧИТАВАЊЕ И СТАРТНИ ПРОЗОР АПЛИКАЦИЈЕ

На самом почетку покретања овог програма, појављује се прозор који нас обавештава о проценту учитаности потребних података. Једног дана, када апликација буде захтевала више меморијског простора и имала више података, овај прозор ће доћи до већег изражаја.



Слика 1. Прозор за учитавање

Након учитавања, појављује се прозор за регистрацију. У том прозору се можете регистровати као корисник или као администратор, а ако сте нови, можете направити свој налог.



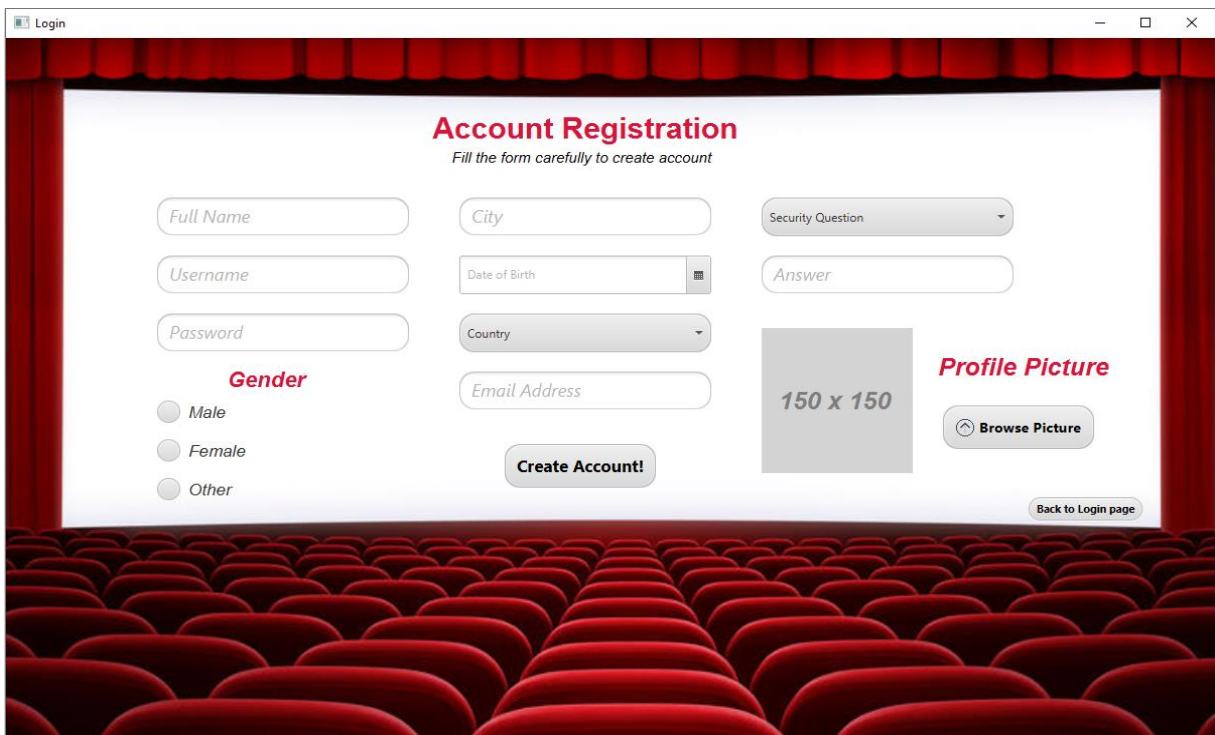
Слика 2. Регистрација

Сада се можемо кретати у три различита смера, што ће бити обрађено у поглављима 2, 3 и 4.

2. ПРАВЉЕЊЕ НОВОГ НАЛОГА

Да бисте направили нови налог, неопходно је да попуните формулар са траженим подацима (слика 3). Обавезно је попунити све податке, а након тога се кликом на дугме “Create account!” креира нови налог који се памти у табели User у бази података.

Важно је напоменути да се осетљиви приватни подаци, попут лозинке, мејл адресе и одговора на сигурносно питање, не чувају у бази као такви, већ се пре бележења провлаче кроз SHA-256 алгоритам, сет криптографских хеш функција. Тиме су ти подаци заштићени од потенцијалних хакерских напада, или било каквих других малверзацијских поступака, будући да је немогуће “дешифровати” податке обрађене овим алгоритмом.



Слика 3. Прављење новог налога

У наставку је код из класе MessageDigest којим се хеширају подаци пре уношења у базу.

```
1 package cinema;
2
3 import java.math.BigInteger;
4 import java.nio.charset.StandardCharsets;
5 import java.security.NoSuchAlgorithmException;
6
7 public class MessageDigest {
8
9     private String s1;
10
11     public static byte[] getSHA(String input) throws NoSuchAlgorithmException {
12         // Static getInstance metoda se poziva
13         // njome odredjujemo koji SHA algoritam zelimo da koristimoA
14         java.security.MessageDigest md = java.security.MessageDigest.getInstance("SHA-256");
15
16         //metoda digest() se poziva i ona vraca niz bajtova
17         //nastalih hesiranjem prosledjenog stringa
18         return md.digest(input.getBytes(StandardCharsets.UTF_8));
19     }
20 }
```

```

20
21 public static String toHexString(byte[] hash) {
22     BigInteger number = new BigInteger(1, hash);
23
24     // Pretvaranje "digestovanog" stringa u hex vrednost
25     StringBuilder hexString = new StringBuilder(number.toString(16));
26
27     // Dodavanje vodечih nula
28     while (hexString.length() < 32) {
29         hexString.insert(0, '0');
30     }
31
32     return hexString.toString();
33 }
34
35 public MessageDigest(String s1) {
36     try {
37         this.s1 = toHexString(getSHA(s1));
38     } // Za navodjenje pogresnih algoritama za hesiranje
39     catch (NoSuchAlgorithmException e) {
40         System.out.println("Exception thrown for incorrect algorithm: " + e);
41     }
42 }
43
44 public String getSHA() {
45     return this.s1;
46 }
47
48 }
49

```

Слика 4. Процес хеширања података – обезбеђује сигурност складиштених података

Након провера да ли су сви подаци попуњени, нови корисник се бележи у базу следећим делом кода:

```

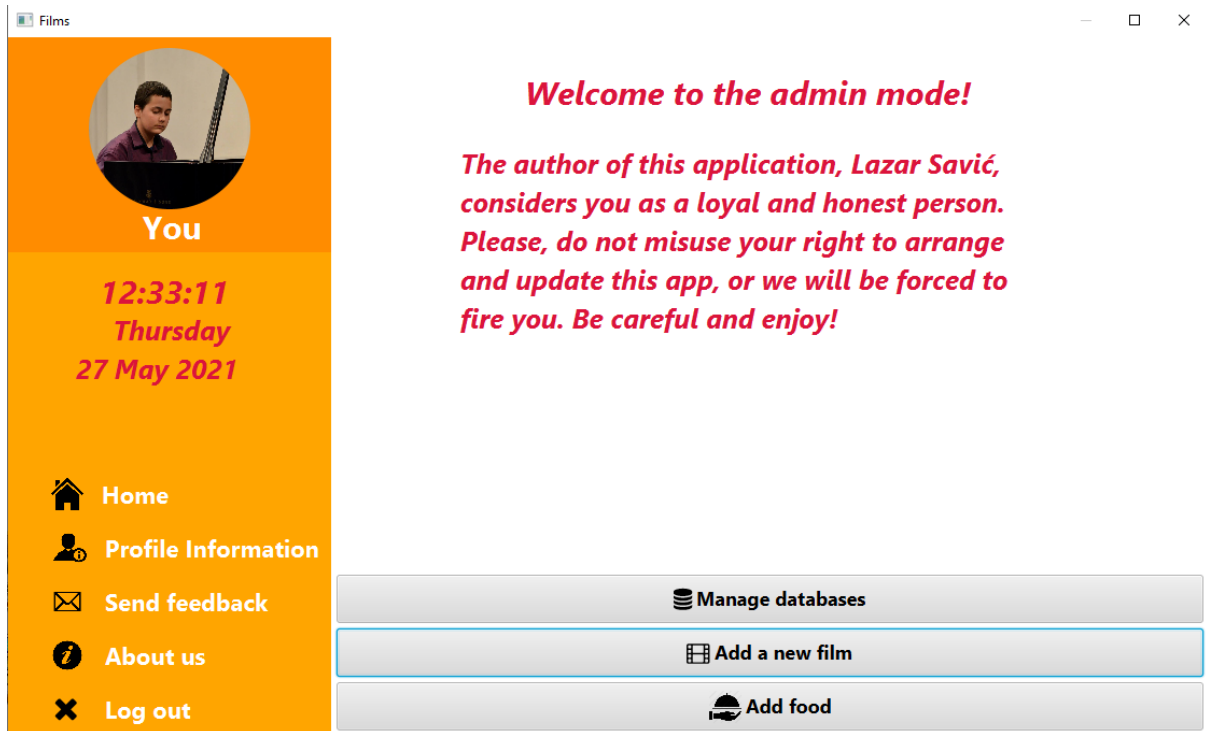
531 private static void insert(String FullName, String UserName, String Password, String Gender, String City, String DateOfBirth,
532     String Country, String Email, String SecurityQuestion, String Answer, byte[] image_user1) {
533
534     String sql = "INSERT INTO User(FullName,UserName,Password,Gender,City,DateOfBirth,Country,Email,SecurityQuestion,Answer,"
535         + "ProfilePicture,Admin) VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?)";
536     try (Connection conn = SqliteConnection.Connector(); PreparedStatement ps = conn.prepareStatement(sql)) {
537         MessageDigest md = new MessageDigest(Password);
538         MessageDigest mdEmail = new MessageDigest(Email);
539         MessageDigest mdAnswer = new MessageDigest(Answer);
540         ps.setString(1, FullName);
541         ps.setString(2, UserName);
542         ps.setString(3, md.getSHA());
543         ps.setString(4, Gender);
544         ps.setString(5, City);
545         ps.setString(6, DateOfBirth);
546         ps.setString(7, Country);
547         ps.setString(8, mdEmail.getSHA());
548         ps.setString(9, SecurityQuestion);
549         ps.setString(10, mdAnswer.getSHA());
550         ps.setBytes(11, image_user1);
551         ps.setInt(12, 0);
552         ps.execute();
553     } catch (SQLException ex) {
554         u = false;
555         Alert alert = new Alert(AlertType.ERROR);
556         alert.setTitle("Creating new account failed");
557         alert.setHeaderText("This username is already taken!");
558         alert.setContentText("Try with another username.");
559         alert.showAndWait();
560         //System.out.println(e.toString());
561     }
562
563 }

```

Слика 5. Убацивање новог корисника у базу

3. АДМИНИСТРАТОРСКИ ИНТЕРФЕЈС

Да бисте се улоговали као администратор потребно је да знате PIN код и да је ваш налог у бази означен као администраторски. Након обављених провера отвара се следећи прозор.



Слика 6. Почетни екран окружења за администраторе

Одавде се приступа свим осталим функционалностима које се пружају администраторима.

3.1. Преглед и манипулација подацима из базе података (дугме “Manage databases”)

Кликом на ово дугме отвара се нови прозор, који је кључан за контролу свих података (слика 7).

IdFilm	Name	Short descrip...	Trailer	Genre	Duration	Added by	Date added	IMDB Rating	Adult ticket ...	Child ticket c...
1	Matrix	When a beaut...	Videos/Matrix...	Thriller	02:16	karla	2020-12-25	8.7	4.0	3.2
2	Amélie	Amélie is an i... Director: Jean... Writers: Guilla... Stars: Audrey ...	Videos/Améli...	Comedy	02:09	karla	2020-12-25	8.3	4.0	3.15
3	Home Alone 2	One year afte... EN Director: Chris... Writers: John ... Stars: Macaul...	Videos/Home...	Comedy	02:01	karla	2021-01-07	6.8	4.0	3.15
4	Mrs. Doubtfire	After a bitter ... EN Director: Chris... Writers: Anne ... Stars: Robin ...	Videos/MrsD...	Comedy	02:05	killa	2021-01-07	7.0	4.0	3.15
5	Broadcast News	Take two rival... Director: Jam... Writer: James ... Stars: William ...	Videos/Broad...	Comedy	02:13	killa	2021-01-07	7.2	4.0	3.15

Слика 7. Учитавање табела за: резервацију, филмове, храну, термине и кориснике и измена њихових података

Дугмад за ажурирање и брисање података функционишу слично за сваку табелу, а ми ћемо њихову примену показати на примеру филмова. Да би се ажурирао податак, неопходно је унети ID реда у коме се он налази (слика 8а). Затим се исписују сви подаци из тог реда, и ми можемо да изменимо онај који желимо (слика 8б). Промена постера (или било ког другог податка у виду слике), може се одрадити притиском на дугме “Change picture”, након чега треба изабрати слику која се налази на вашем рачунару.

Слика 8а. Форма за ажурирање пре уношења ID-а

Слика 8б. Форма за ажурирање након уношења ID-а

Део кода који врши ажурирање ових података приказан је испод. После измене података, те нове податке треба да забележимо у нашу листу филмова, коју користимо у наставку програма. Такође, те податке треба ажурирати и у бази, што се реализује простим UPDATE упитом.

```
1003 up.setOnMouseClicked((javafx.scene.input.MouseEvent me) -> {
1004     for (int i = 0; i < filmList.size(); i++) {
1005         if (filmList.get(i).getIdFilm() == Integer.parseInt(id.getText())) {
1006             filmList.get(i).setIme(name.getText());
1007             filmList.get(i).setKratakOpis(sd.getText());
1008             filmList.get(i).setTrejler(trailer.getText());
1009             filmList.get(i).setZanr(genre.getText());
1010             filmList.get(i).setDuzina(dur.getText());
1011             filmList.get(i).setDodao(ab.getText());
1012             filmList.get(i).setDatumDodavanja(date.getText());
1013             filmList.get(i).setIMDBRating(IMDB.getText());
1014             filmList.get(i).setCenaKarteD(Double.parseDouble(ct.getText()));
1015             filmList.get(i).setCenaKarteO(Double.parseDouble(at.getText()));
1016
1017             textArea.setText(textArea.getText() + "Film with ID " + Integer.parseInt(id.getText()) + " updated.\n");
1018
1019             String query = "UPDATE Film SET Ime = ?, KratakOpis = ?, Trejler = ?, Zanr = ?, Duzina = ?, Dodao = ?, DatumDodavanja = ?, "
1020                 + "Rating = ?, CenaKarteD = ?, CenaKarteO = ? WHERE IdFil = ?";
1021             try (Connection con = SqliteConnection.Connector(); PreparedStatement ps = con.prepareStatement(query)) {
1022                 ps.setString(1, name.getText());
1023                 ps.setString(2, sd.getText());
1024                 ps.setString(3, trailer.getText());
1025                 ps.setString(4, genre.getText());
1026                 ps.setString(5, dur.getText());
1027                 ps.setString(6, ab.getText());
1028                 ps.setString(7, date.getText());
1029                 ps.setString(8, IMDB.getText());
1030                 ps.setString(9, ct.getText());
1031                 ps.setString(10, at.getText());
1032                 ps.setInt(11, Integer.parseInt(id.getText()));
1033                 ps.executeUpdate();
1034             } catch (SQLException ex) {
1035                 Logger.getLogger(Manage.class.getName()).log(Level.SEVERE, null, ex);
1036             }
1037         }
1038     }
1039 }
1040 });
```

Слика 9. Ажурирање података о филму

3.2. Додавање нових филмова у базу података (дугме “Add a new film”)

Кликом на ово дугме отвара се формулар који је неопходно попунити да бисмо додали нови филм на репертоар (слика 10). Поред општих података о филму, потребно је назначити и термине пројекција. Ово дугме је значајно и за задавање нових термина пројекција већ постојећих филмова на репертоару. Потребно је само назначити име филма и његову дужину.

The screenshot shows a web application window titled "Films". On the left is a sidebar with a user profile picture and the name "You". Below the profile, it shows the time "12:41:00", the day "Thursday", and the date "27 May 2021". The sidebar contains navigation links: Home, Profile Information, Send feedback, About us, and Log out. The main content area is a form for adding a new movie. It includes fields for Title, Poster, Trailer, Genre, and Length. There is also a section for IMDB rating and ticket costs. A "Specify projection times" button is visible. At the bottom, there are buttons for "Manage databases", "Add a new film", and "Add food".

Слика 10. Форма за уношење новог филма у базу података

За задавање пројекције, неопходно је навести датум приказивања, као и број пројекција у току дана. Затим је за сваку пројекцију потребно назначити салу у којој се емитује филм, као и време почетка, а време завршетка се аутоматски израчунава. Унети термини не смеју да се преклапају временски (ако су у истој сали), како међусобно, тако и са раније заказаним пројекцијама. Администратор може лако да провери који термини су слободни, и затим може да изабере најпогоднији термин за пројекцију. Дугме “Record” снима унете податке у базу.

The screenshot shows a web application window titled "Projection times". It features a date picker for "Choose the date of projection" set to 5/23/2021, and a dropdown menu for the number of projections set to 6. There is a "Record" button and a table for specifying start and end times for multiple screens. The table has columns for "Starts:", "Ends:", and "Screen:". The "Screen:" column has radio buttons for 1, 2, and 3. The "Starts:" and "Ends:" columns have input fields for hh:mm. A link "See available terms for 2021-05-23" is visible at the bottom left.

Слика 11. Задавање термина пројекција филма

Део кода које оно извршава дат је испод (слика 12).

```

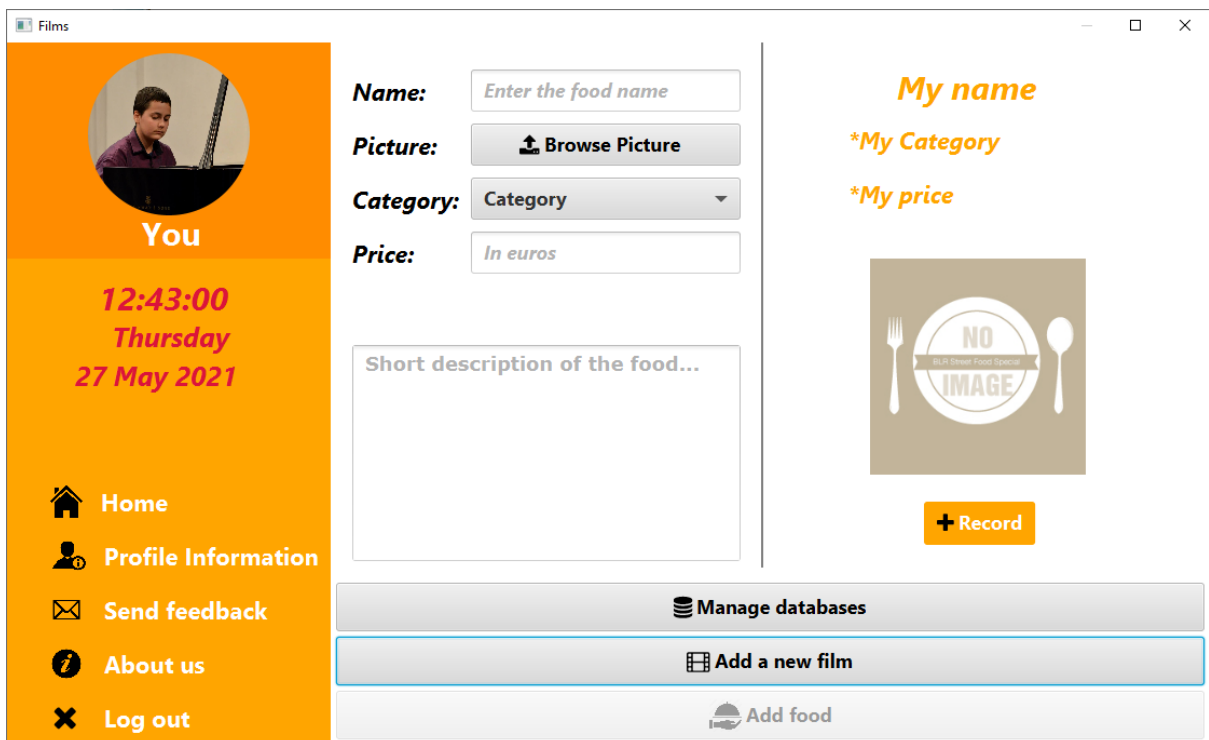
391 String sql = "INSERT INTO Termin(ImeFilma,Datum,Pocetak,Kraj,Dodao,Sala,Aktivan) VALUES(?,?,?,?,?,?,?)";
392 try (Connection con = SqliteConnection.Connector(); PreparedStatement ps = con.prepareStatement(sql)) {
393     ps.setString(1, imeFilma);
394     ps.setString(2, dpl.getValue().toString());
395     ps.setString(3, t[i].getText());
396     ps.setString(4, to[i].getText());
397     ps.setString(5, user);
398     ps.setInt(6, Integer.parseInt(((RadioButton) group[i]).getSelectedToggle().getText()));
399     ps.setInt(7, 1);
400     ps.execute();
401 } catch (SQLException e) {
402     uspeo = false;
403     System.out.println(e);
404 }

```

Слика 12. Додавање нових термина пројекција у базу

3.3. Додавање нове хране у базу података (дугме “Add food”)

Прозор за додавање нове хране приказан је на слици 13. Слично као код додавања новог филма, и за храну је неопходно уписати све тражене информације. Снимање података се врши по истом принципу као и код филмова.



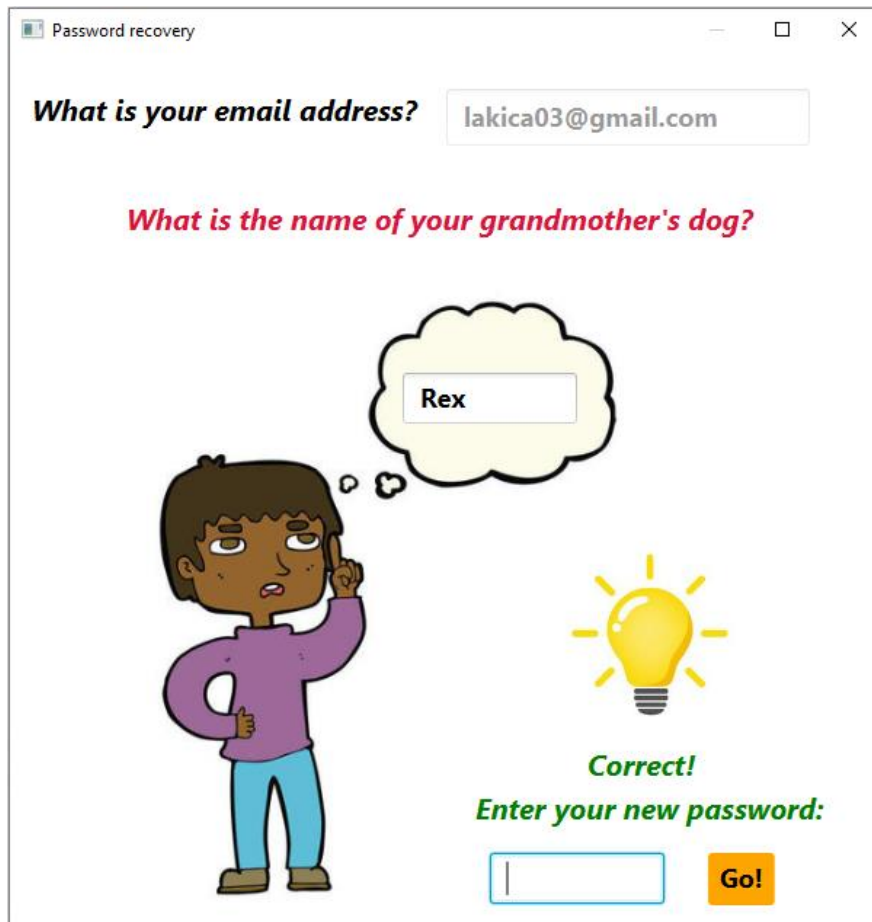
Слика 13. Додавање хране у базу података

4. КОРИСНИЧКИ ИНТЕРФЕЈС

Да бисте се пријавили на свој налог, неопходно је да унесете своје корисничко име и лозинку. Уколико сте заборавили своју лозинку, можете задати нову.

4.1. Заборављена лозинка

Често се дешава да заборавимо нашу лозинку. У овом програму није могуће повратити стару лозинку, али се зато може задати нова лозинка. Потребно је да упишете своју мејл адресу и одговор на сигурносно питање који сте назначили приликом регистрације. Након тога можете задати своју нову лозинку.



Слика 14. Заборављена лозинка

Део кода који врши ажурирање нове лозинке је у наставку.

```
179 MessageDigest md = new MessageDigest(newPass.getText());
180 MessageDigest md1 = new MessageDigest(tr.getText());
181 String query = "UPDATE User SET Password = ? WHERE Email = ?";
182 try (Connection con1 = SqliteConnection.Connector(); PreparedStatement ps = con1.prepareStatement(query)) {
183     ps.setString(1, md.getSHA());
184     ps.setString(2, md1.getSHA());
185     ps.executeUpdate();
186 }
```

Слика 15. Ажурирање лозинке

4.2. Регистрација и преглед филмова

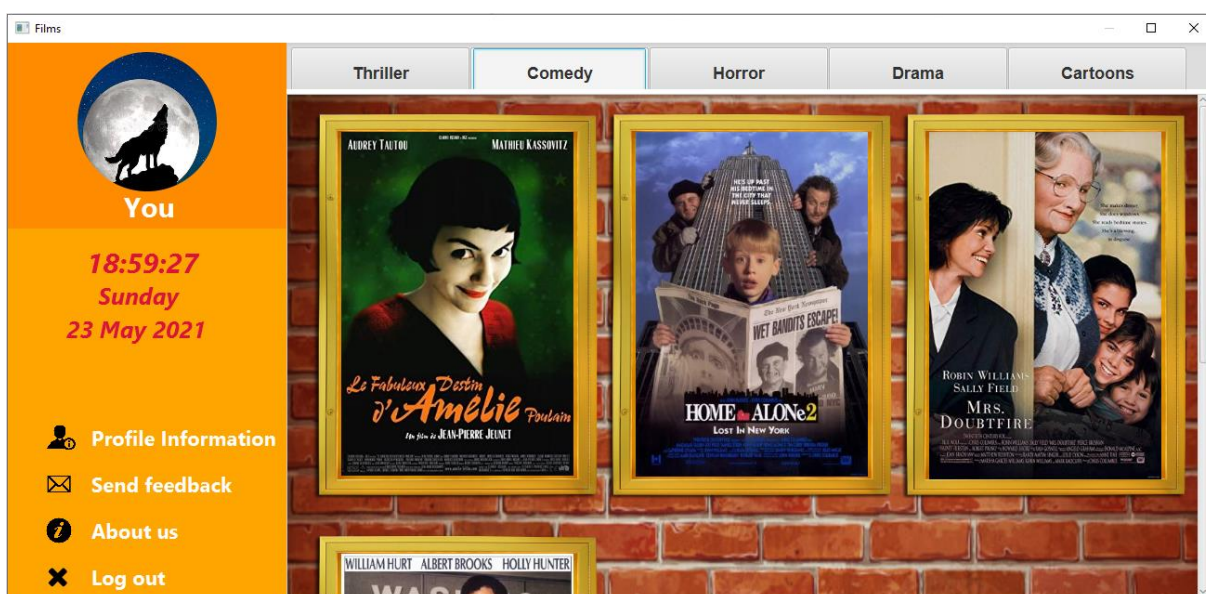
Ово је део кода који показује како се проверава да ли је корисник унео валидне податке и да ли може да се пријави на свој профил.

```
93 try {
94     String sql = "SELECT * FROM User WHERE UserName = ?";
95     ps = con.prepareStatement(sql);
96     ps.setString(1, username);
97     rs = ps.executeQuery();
98     MessageDigest md = new MessageDigest(password);
99     String password1 = md.getSHA();
100    String realPassword = rs.getString("Password");
101    admin = rs.getInt("Admin");
102    if (!realPassword.equals(password1)) {
103        Alert alert = new Alert(Alert.AlertType.ERROR);
104        alert.setTitle("Error");
105        alert.setContentText("Wrong password!");
106        alert.showAndWait();
107    } else if (admin != adminInd) {
108        Alert alert = new Alert(Alert.AlertType.ERROR);
109        alert.setTitle("Error");
110        if (admin == 0) {
111            alert.setContentText("You can not login as an administrator.");
112        }
113        if (admin == 1) {
114            alert.setContentText("You can not login as a user.");
115        }
116        alert.showAndWait();
117    } else {
119        this.fullname = rs.getString("FullName");
120        this.gender = rs.getString("Gender");
121        this.city = rs.getString("City");
122        this.dateofbirth = rs.getString("DateOfBirth");
123        this.country = rs.getString("Country");
124        this.email = rs.getString("Email");
125        this.securityquestion = rs.getString("SecurityQuestion");
126        this.answer = rs.getString("Answer");
127
128        byte[] slika = rs.getBytes("ProfilePicture");
129        rs.close();
130        ps.close();
131        con.close();
132
133        ByteArrayInputStream bis = new ByteArrayInputStream(slika);
134        BufferedImage bImage2 = ImageIO.read(bis);
135        Image image = SwingFXUtils.toFXImage(bImage2, null);
136        this.iv = new ImageView(image);
137        con.close();
138        primaryStage.close();
139        if (admin == 0) {
140            User u = new User(fullname, username, password1, gender, city, dateofbirth, country,
141                email, securityquestion, answer, iv, filmList, foodList);
142        } else {
143            Admin a = new Admin(fullname, username, password1, gender, city, dateofbirth, country,
144                email, securityquestion, answer, iv, filmList, foodList, users);
145        }

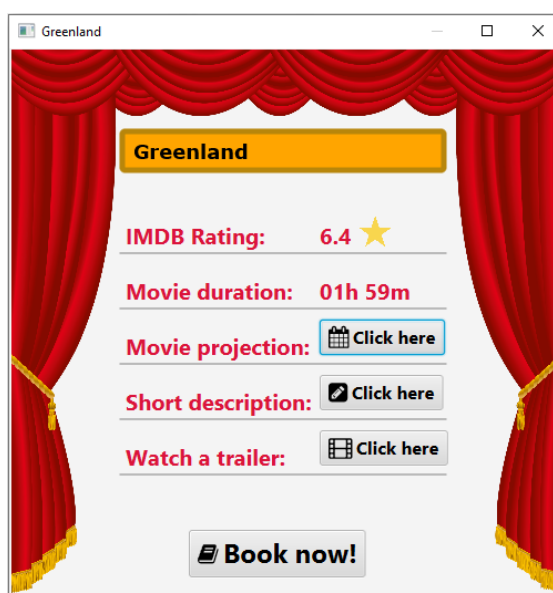
```

Слика 16. Једноставним SELECT упитом узимамо хеширану лозинку за унето корисничко име, провлачимо унету лозинку кроз SHA-256 алгоритам и проверавамо да ли су та два стринга једнака

Филмови су подељени у 5 жанрова. Корисник може да бира неки од доступних филмова из одговарајућег жанра који жели да гледа, и кликом на њега отвара се прозор са основним подацима о филму (слика 17). Поред кратког описа филма, корисник може и да погледа његов трејлер (слика 18). Уколико му се филм допадне, може да погледа распоред термина, а потом и да резервише карте за избрани термин.

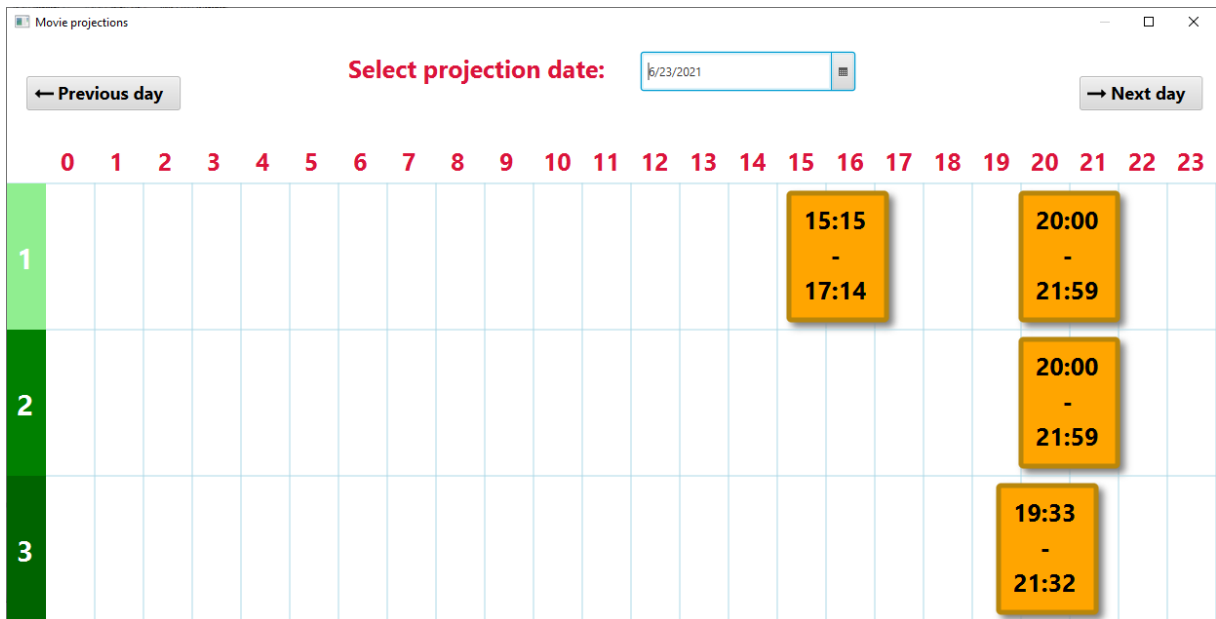


Слика 17. Почетни кориснички екран: доступни филмови и лични подаци



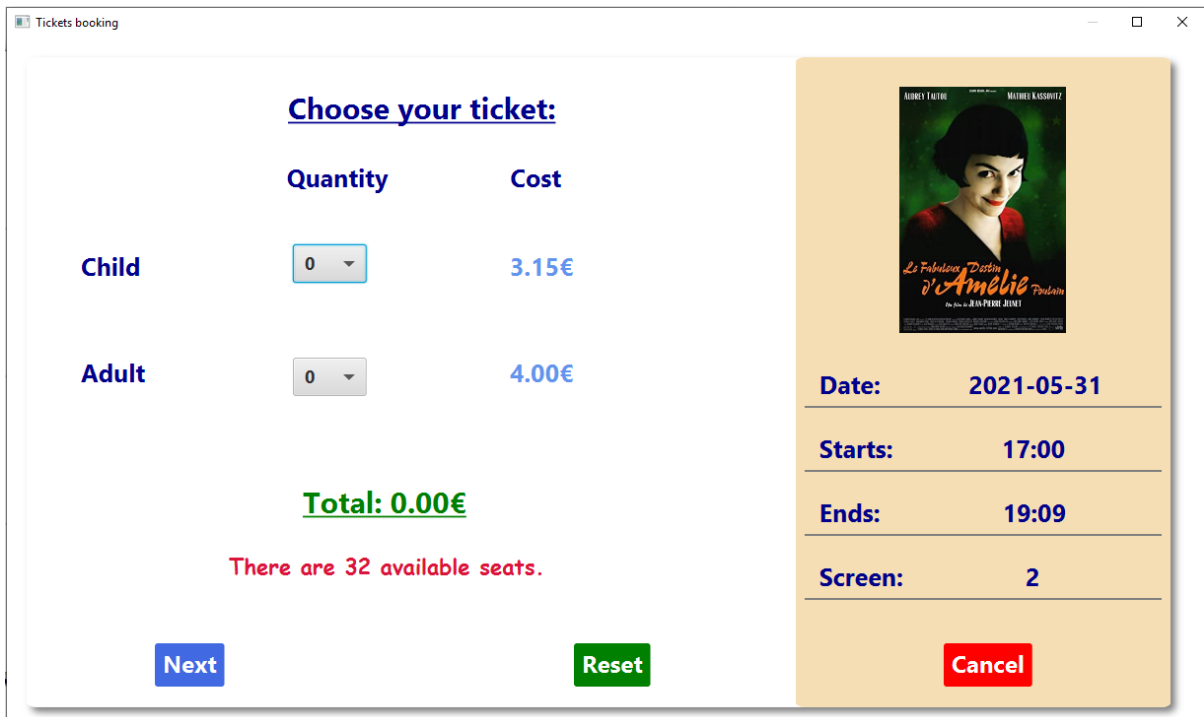
Слика 18. Информације о филму

Кликом на дугме “Book now!” отвара се нови прозор у којем креће креирање резервације. Најпре је потребно изабрати неку од предстојећих пројекција (слика 19).



Слика 19. Одабир жељене пројекције

Након одабира пројекције, отвара се нови прозор, на којем су сумирани подаци о изабраној пројекцији, као и са ценама карте за децу и одрасле. Неопходно је назначити број карата за обе категорије (слика 20). Кликом на дугме "Next" прелази се на следећи корак резервације.



Слика 20. Информације о пројекцији

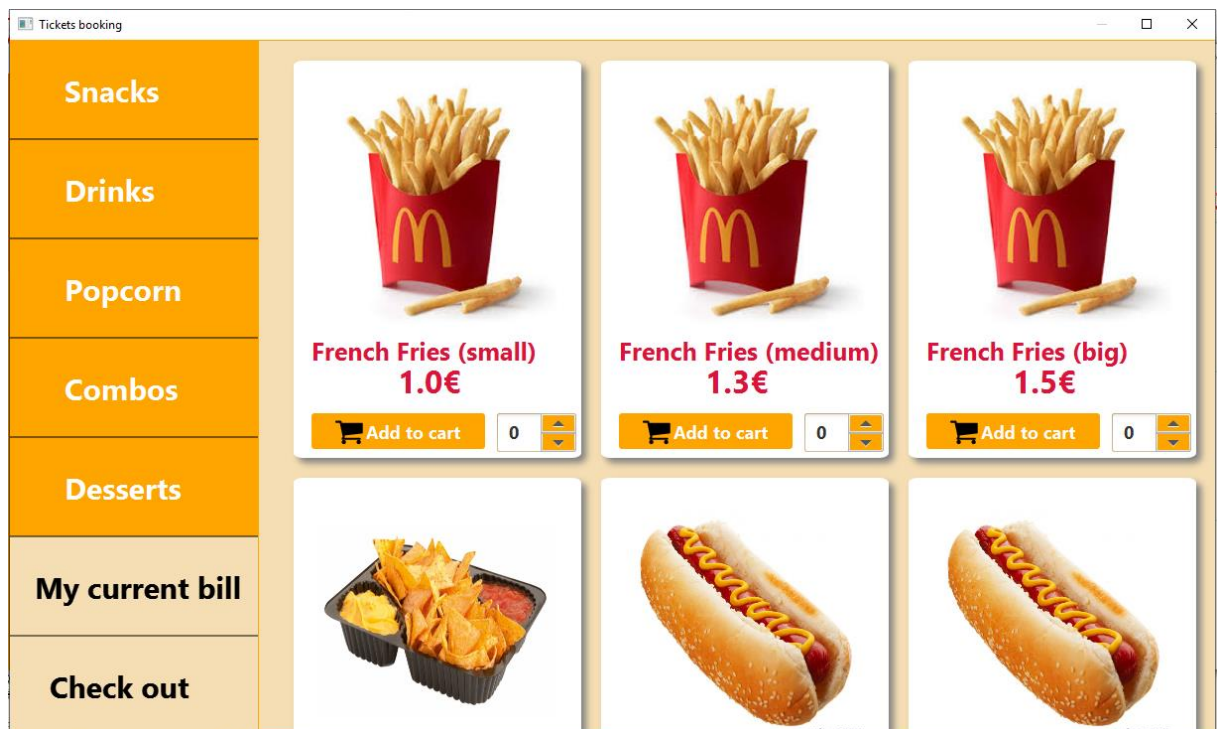
У наредном кораку, корисник одлучује да ли жели да наручи храну. Уколико је одговор потврдан, отвара се мени са храном у понуди, а у супротном се одмах прелази на следећи корак резервације.



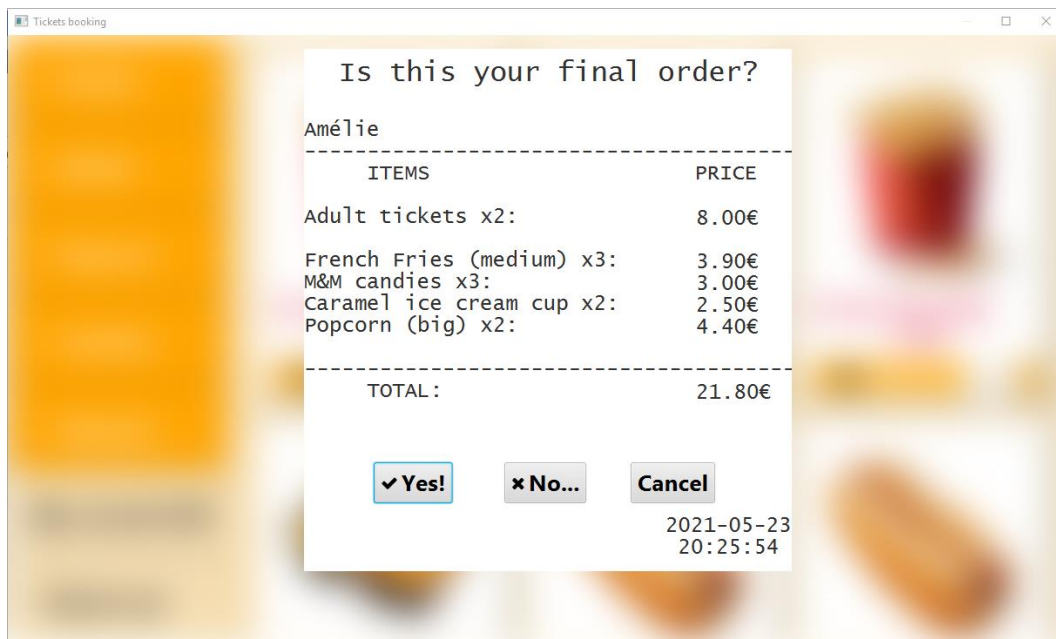
Слика 21. Могућност избора хране

4.3. Поручивање хране

Уколико је корисник одлучио да поручи храну, појавиће се прозор као на слици 22. Постоје различите категорије хране које могу да се изаберу. Када се корисник определи коју храну жели да узме, потребно је да назначи број колико пута жели да узме то јело и да га затим дода у корпу. У сваком тренутку корисник може да има увид у свој рачун (слика 23).



Слика 22. Изглед менија

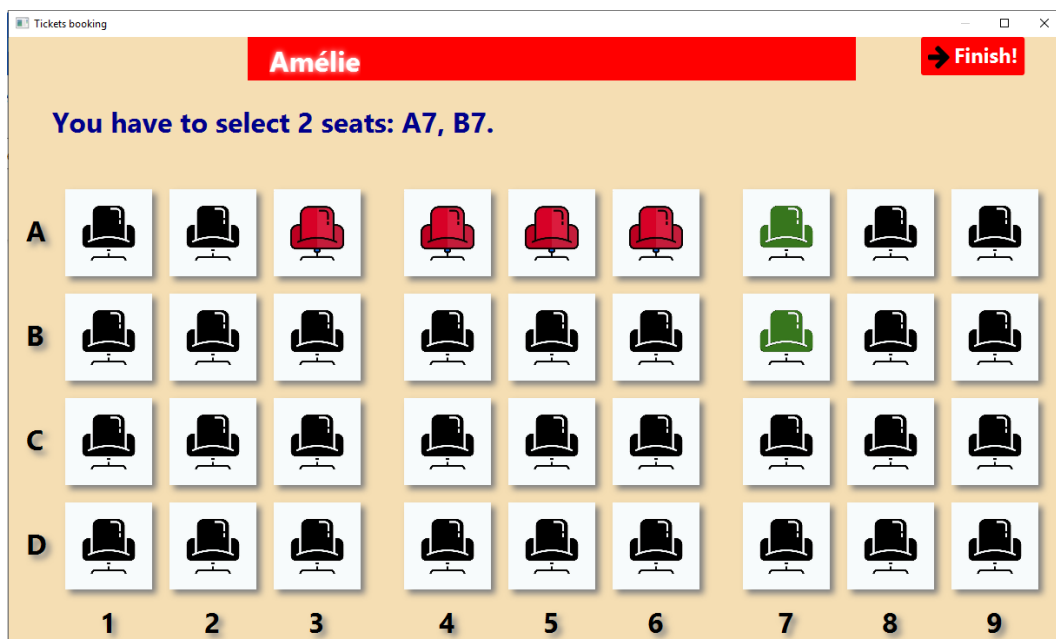


Слика 23. Пример једног рачуна

Када заврши са одабиром хране, потребно је да кликне на “Check out” и провери да ли је то његов рачун. Уколико је одговор потврдан, прелази се на следећи корак прављења резервације, а уколико је дошло до неке грешке, корпа се празни и поново се бира жељена храна.

4.4. Одабир седишта и завршетак резервације

Након што се обаве све неопходне припреме, последњи корак прављења резервације јесте одабир седишта у биоскопској сали. Црвеном бојом су означена заузета седишта, а црном слободна. Потребно је изабрати онолико црних седишта колико је назначено карата на почетку резервације. Након што се довољан број седишта означи, у горњем десном ћошку се појављује дугме за завршетак резервације.

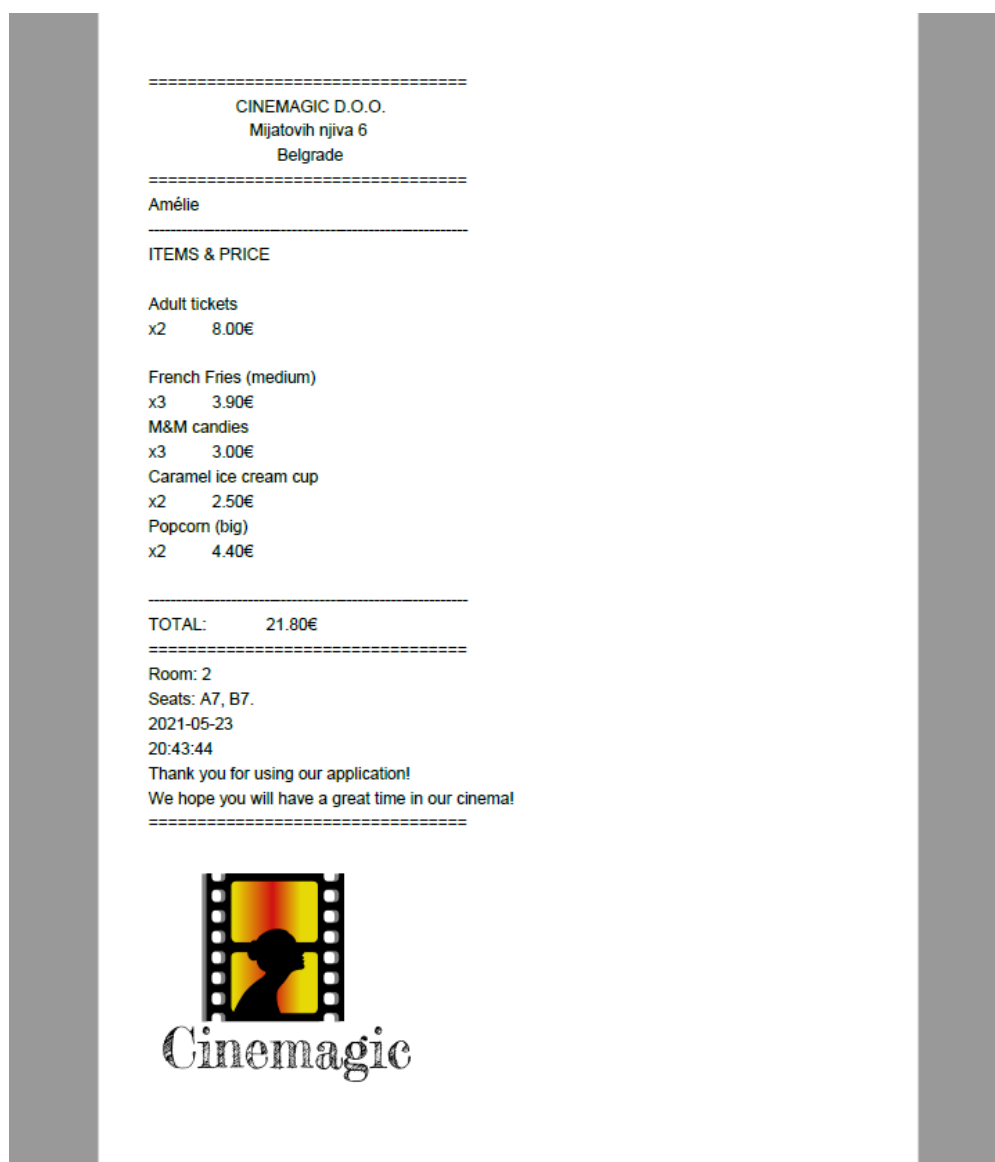


Слика 24. Изглед седишта у биоскопској сали. Изабрана седишта су означена зеленом бојом.

4.5. Преузимање резервације

Након што су сви неопходни подаци за резервацију попуњени, преостаје још само преузимање карте за резервацију. Кликом на дугме “Finish!”, отвара се помоћни прозор који обавештава корисника да је карта спремна за преузимање.

Кликом на “Download” од корисника се тражи да назначи фолдер у којем жели да сачува своју карту у PDF формату и потом се тај документ смешта у назначени фолдер. Он у себи садржи све информације о креираној резервацији. Пример таквог документа дат је на слици 25.



Слика 25. Пример резервације

4.6. Уређивање профила

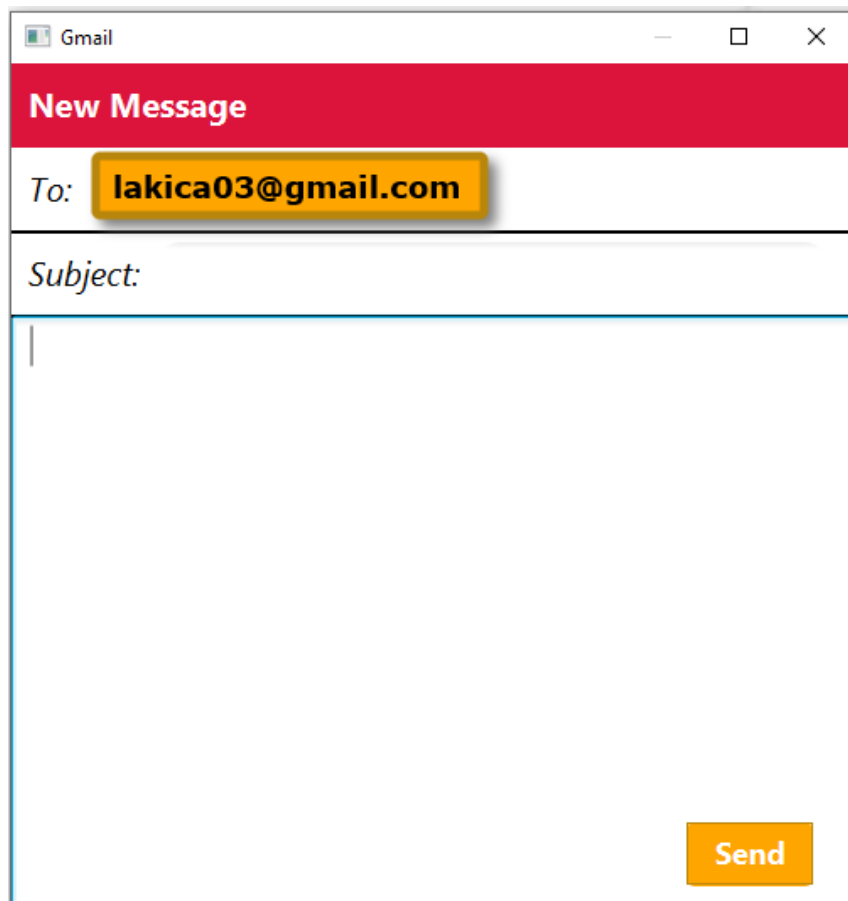
Опцију за преуређивање профила имају и корисници и администратори (слика 26). Промењене податке је потребно сачувати.



Слика 26. Информације о профилу

4.7. Слање повратних информација

Будући да је мени као оснивачу ове апликације стало до мишљења корисника, омогућено је и слање мејла (слика 27). У њему се може изнети мишљење о раду биоскопа и функционалности апликације, као и сугестија за побољшање истих.



Слика 27. Изглед мејла

5. АУТОМАТСКО АЖУРИРАЊЕ ПОДАТАКА

Уколико не бисмо имали овај помоћни програм, базе података би се у једном тренутку препуниле подацима који нису актуелни, и програм не би функционисао како треба. Зато је сврха овог програма да се системски, сваких 10 минута (ово се лако може мењати у зависности од потреба), покреће и извршава означавање оних података који више нису актуелни. Под тиме се подразумевају термини за пројекцију филма који су прошли, као и резервације које су биле на тим пројекцијама.

Да би се ово постигло коришћен је Task Scheduler, који извршава задатке који му се задају. Конкретно, потребно је направити нови задатак, и задати колико пута и у ком временском интервалу желимо да се овај задатак понавља. Неопходно је назначити путању до фолдера у ком је сачуван *executable jar file* пројекта. Ту треба да се налази Windows Batch File (.bat), који ће да се покреће сваких 10 минута. Конкретно, у мом .bat фајлу се налази следећа наредба:

```
1 //Putanja do jave
2 cd "C:\Program Files\Java\jdk1.8.0_221\bin"
3
4 //Putanja do foldera u kom je sacuvan executable jar file projekta
5 java -jar "C:\Users\Lazar\Documents\NetBeansProjects\FilterTermins\dist\FilterTermins.jar"
```

Слика 28. Код из Windows Batch фајла, који се покреће сваких 10 минута

Принцип функционисања програма је једноставан: у тренутку када се покрене програм, пролази се кроз податке из базе и упоређују се тренуци завршетка пројекције и тренутног времена (Јавино системско време). Уколико се одреди да је пројекција завршена, статус пројекције или резервације се поставља на 0 (неактивно). Ако филм нема ниједну активну пројекцију, његов статус се такође поставља на нулу. Додавањем нове пројекције, његов статус се враћа на један. Део кода који то извршава дат је у наставку.

```

29 String sql = "SELECT IdTerm,ImeFilma,Datum,Kraj FROM Termin";
30 String query = "UPDATE Termin SET Aktivan = 0 WHERE IdTerm = ?";
31 try (Connection con = SqliteConnection.Connector();
32     PreparedStatement ps = con.prepareStatement(sql);
33     ResultSet rs = ps.executeQuery();
34     PreparedStatement psl = con.prepareStatement(query)) {
35     while (rs.next()) {
36         int IdTerm = rs.getInt("IdTerm");
37         String imeFilma = rs.getString("ImeFilma");
38         String datum = rs.getString("Datum");
39         String kraj = rs.getString("Kraj");
40         String vreme = datum + " " + kraj;
41         int year = Integer.parseInt(vreme.substring(0, 4));
42         int month = Integer.parseInt(vreme.substring(5, 7));
43         int day = Integer.parseInt(vreme.substring(8, 10));
44         int hour = Integer.parseInt(vreme.substring(11, 13));
45         int minute = Integer.parseInt(vreme.substring(14, 16));
46         if (year < tmpyear || (year == tmpyear && month < tmpmonth) ||
47             (year == tmpyear && month == tmpmonth && day < tmpday) ||
48             (year == tmpyear && month == tmpmonth && day == tmpday && hour < tmphour) ||
49             (year == tmpyear && month == tmpmonth && day == tmpday && hour == tmphour && minute < tmpminute)) {
50             psl.setInt(1, IdTerm);
51             psl.executeUpdate();
52         }
53     } else {
54         if (!availableFilms.contains(imeFilma)) {
55             availableFilms.add(imeFilma);
56         }
57     }
58 }
59 }
60 } catch (SQLException e) {
61     System.out.println(e);
62 }

```

Слика 29. Избацивање пројекција које су се завршиле

Закључак

Развојем човечанства мењају се и потребе. Глобални циљ тог развоја јесте олакшавање живота, убрзавање процеса и прецизније извршавање задатака. Ова апликација би могла да представља један делић тог пута ка променама. Одлазак у биоскоп треба да преставља задовољство, па и сам процес припреме за то не треба да има примесе бриге, нервозе и стреса.

Принцип функционисања апликације је прилично једноставан. Као корисник, можете да направите своју резервацију и без чекања у реду уђете да погледате свој филм, а као администратор, имате одређене повластице и приступ оним подацима које немају корисници. Даљим усавршавањем ове апликације, могле би да се покрију и многе друге ствари. Неки од тих предлога су: подсетник за филм, обавештење за нови репертоар, нови жанрови филмова, садржајнији мени, повластице за сталне кориснике...

Литература

- [1] Herbert Schildt. *Java JDK 9: kompletan priručnik, prevod 10. Izdanja*. Oracle, 2018.
- [2] Vladimir Blagojević. *Relacione baze podataka I*. ICNT Beograd, 2006.
- [3] Doug Lowe. *JavaFX For Dummies*. John Wiley & Sons, Inc, 2015.
- [4] Allen G. Taylor. *SQL All-In-One For Dummies, 3rd Edition*. John Wiley & Sons, Inc, 2019.
- [5] Alan Beaulieu. *Learning SQL: Generate, Manipulate and Retrieve Data*. O'Reilly Publishing, 2020.
- [6] Upom Malik, Matt Goldwasser and Benjamin Johnston. *SQL for Data Analytics*. Packt publishing, 2019.
- [7] Martin Kleppmann. *Designing Data-Intensive Applications. The big ideas behind reliable, scalable and maintainable systems*. O'Reilly Publishing, 2017.