**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА**

**ФАКУЛТЕТ ПО ИЗЧИСЛИТЕЛНА ТЕХНИКА И АВТОМАТИЗАЦИЯ**

Катедра:

**Софтуерни и интернет технологии**

Специалност:

**Софтуерни и интернет технологии**

Дисциплина:

**Обектно-ориентирано програмиране част 2**

**КУРСОВА РАБОТА**

Билетен Център – Група 25

**Име на студента:** Георги Вълков

**Факултетен номер:** 19621749

**Курс: III**

**Група: 5б**

**Име на студента:** Стоян Чомпов

**Факултетен номер:** 19621746

**Курс: III**

**Група: 5б**

**Условие на проект „Билетен Център“**

**Билетен център**

**Да се разработи информационна система, предоставяща услуга билетен център.**

**Програмата съхранява и обработва данни за разпространение на билети. Системата**

**позволява множествен достъп.**

**Системата поддържа два вида потребители администратор и клиенти (организатор,**

**разпространител) с различни роли за достъп до функционалностите в системата.**

**Операции за работа с потребители:**

* **Създаване на организатори от администратор;**
* **Създаване на разпространители от администратор;**
* **Създаване, редактиране на събития с избор на един или списък разпространители от собственик за продаване на билети**
* **Поддържане на профили с характеристики на организатори и разпространители (хонорар и др…)**
* **Рейтинговане на разпространителите**

**Системата поддържа операции за работа със събития:**

* **Добавяне на ново събитие от организатор (вид на събитието, брой места, видове места, цена на билетите по видове, ограничение в закупуването на билет от едно лице и др…);**
* **Продаване на билети за събитие от разпространител, създаване на формуляр за закупуване (информация за купувача, избор на място и др…);**

**Системата поддържа справки по произволен период за:**

* **Разпространител (разпродадени билети от различни видове събития, данните на разпространителя, рейтинг)**
* **Събития (дата, статус, местоположение, и др…); Организатор на събитие достъпва справки само за събитията, на който е организатор.**
* **Организатора достъпва справки за всички организирани от него събития.**
* **Разпространител има право на справки за събития които са му.**

**Системата поддържа Известия за събития:**

* **Новопостъпила заявка за събитие (в профила на разпространител);**
* **Периодично уведомление за продадени билети от събитие (в профила на собственика).**
* **Наближаващо събитие с не продадени билети (собственик, разпространител на билети)**

**Функционалани изисквания**

* Създаване на организатори от администратор;
* Създаване на разпространители от администратор;
* Създаване, редактиране на събития с избор на един или списък
* разпространители от собственик за продаване на билети
* Поддържане на профили с характеристики на организатори и разпространители
* (хонорар и др…)
* Рейтинговане на разпространителите
* Добавяне на ново събитие от организатор (вид на събитието, брой места,
* видове места, цена на билетите по видове, ограничение в закупуването на
* билет от едно лице и др…);
* Продаване на билети за събитие от разпространител, създаване на формуляр
* за закупуване (информация за купувача, избор на място и др…);
* Разпространител (разпродадени билети от различни видове събития, данните
* на разпространителя, рейтинг)
* Събития (дата, статус, местоположение, и др…);
* Новопостъпила заявка за събитие (в профила на разпространител);
* Периодично уведомление за продадени билети от събитие (в профила на
* собственика).
* Наближаващо събитие с не продадени билети (собственик, разпространител на
* билети)

**Структура на проекта**

**/Ticket\_Center.idea/src/main/java/bg/tu\_varna/sit/group25/ticketcenter/Application**

**>HelloApplication**

**Използваме HelloApplication за показване на прототипната страница, както и заявки от баата данни.**

**>TestApp**

**Използваме HelloApplication за реалното стартиране на приложението, като първото което излиза е, опция за вписване, като „Разпространител“ или „Организатор“.**

**/Ticket\_Center.idea/src/main/java/bg/tu\_varna/sit/group25/ticketcenter/business.services**

>CustomerService

package bg.tu\_varna.sit.group25.ticketcenter.Application.business.services;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.CustomerRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.CustomerListViewModel;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class CustomerService {  
 private final CustomerRepository repository=CustomerRepository.*getInstance*();  
  
 public static CustomerService getInstance()  
 {  
 return CustomerServiceHolder.*INSTANCE*;  
 }  
  
 private static class CustomerServiceHolder {  
 public static final CustomerService *INSTANCE* =new CustomerService();  
 }  
  
 public Customer checkTickets(List<Customer> c)  
 {  
 List<Customer>customers=repository.getAll();  
 for(Customer customer:customers)  
 {  
 if(customer.equals(c))  
 {  
 return customer;  
 }  
 }  
 return (Customer) c;  
 }  
  
 public ObservableList<CustomerListViewModel>getAllCustomer()  
 {  
 List<Customer>customers=repository.getAll();  
  
 return FXCollections.*observableList*(customers.stream().map(c->new CustomerListViewModel(  
 c.getCustomer\_ID(),c.getCustomer\_Name(),c.getPhone\_Number(),c.getTicket\_Limit())).collect(Collectors.*toList*()));  
 }  
  
 public Customer getCustomerByName(String name)  
 {  
 List<Customer> customers=CustomerRepository.*getInstance*().getAll();  
 Customer temp=new Customer();  
 for (Customer c:customers)  
 {  
 if(c.getCustomer\_Name().equals(name))  
 {  
 temp=c;  
 }  
 }  
 return temp;  
 }  
}

>DistributorService

package bg.tu\_varna.sit.group25.ticketcenter.Application.business.services;  
  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Distributor;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.DistributorRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.DistributorListViewModel;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class DistributorService {  
 private final DistributorRepository repository = DistributorRepository.*getInstance*();  
  
 public static DistributorService getInstance() {  
 return DistributorServiceHolder.*INSTANCE*;  
 }  
  
 private static class DistributorServiceHolder {  
 public static final DistributorService *INSTANCE* = new DistributorService();  
 }  
  
 public ObservableList<DistributorListViewModel>getPass()  
 {  
 List<Distributor>distributors=repository.getAll();  
  
 return FXCollections.*observableList*(distributors.stream().map(d -> new DistributorListViewModel(  
 d.getPassword\_D()  
 )).collect(Collectors.*toList*()));  
 }  
  
 public ObservableList<DistributorListViewModel>getName()  
 {  
 List<Distributor>distributors=repository.getAll();  
  
 return FXCollections.*observableList*(distributors.stream().map(d -> new DistributorListViewModel(  
 d.getUserName\_D()  
 )).collect(Collectors.*toList*()));  
 }  
  
 public ObservableList<DistributorListViewModel> getAllDistributor() {  
 List<Distributor> distributors = repository.getAll();  
  
 return FXCollections.*observableList*(distributors.stream().map(d -> new DistributorListViewModel(d.getDistributor\_ID(),  
 d.getName\_D(),  
 d.getPhone\_Number\_D(),  
 d.getFee\_D(),  
 d.getRating(),  
 (int) d.getRating\_Count(),  
 d.getUserName\_D(),  
 d.getPassword\_D())).collect(Collectors.*toList*()));  
  
  
 }  
}

>FormService

package bg.tu\_varna.sit.group25.ticketcenter.Application.business.services;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Form;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.CustomerRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.FormRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.CustomerListViewModel;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.FormListViewModel;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class FormService  
{  
 private final FormRepository repository= FormRepository.*getInstance*();  
  
 public static FormService getInstance()  
 {  
 return FormService. FormServiceHolder.*INSTANCE*;  
 }  
  
 private static class FormServiceHolder {  
 public static final FormService *INSTANCE* =new FormService();  
 }  
  
 public ObservableList<FormListViewModel> getAllForms()  
 {  
 List<Form> forms=repository.getAll();  
  
 return FXCollections.*observableList*(forms.stream().map(f->new FormListViewModel(  
 f.getSold\_Tickets(),  
 f.getPrice\_Ticket(),  
 f.getDistributor().getDistributor\_ID(),  
 f.getCustomer().getCustomer\_ID(),  
 f.getShow().getShow\_ID()  
 )).collect(Collectors.*toList*()));  
 }  
}

>OrganizerService

package bg.tu\_varna.sit.group25.ticketcenter.Application.business.services;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Organizer;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.OrganizerRepository;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.OrganizerListViewModel;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class OrganizerService  
{  
 private final OrganizerRepository repository = OrganizerRepository.*getInstance*();  
  
 public static OrganizerService getInstance() {  
 return OrganizerService.OrganizerServiceHolder.*INSTANCE*;  
 }  
  
 private static class OrganizerServiceHolder {  
 public static final OrganizerService *INSTANCE* = new OrganizerService();  
 }  
  
  
 public ObservableList<OrganizerListViewModel> getAllOrganizers() {  
 List<Organizer>organizers = repository.getAll();  
  
 return FXCollections.*observableList*(organizers.stream().map(o-> new OrganizerListViewModel(  
 o.getOrganizer\_ID(),  
 o.getName\_O(),  
 o.getPhone\_Number\_O(),  
 o.getFee\_O(),  
 o.getUserName\_O(),  
 o.getPassword\_O())).collect(Collectors.*toList*()));  
 }  
}

>ShowService

package bg.tu\_varna.sit.group25.ticketcenter.Application.business.services;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Show;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.ShowRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.ShowListViewModel;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class ShowService {  
 private final ShowRepository repository= ShowRepository.*getInstance*();  
  
 public static ShowService getInstance()  
 {  
 return ShowService. ShowServiceHolder.*INSTANCE*;  
 }  
  
 private static class ShowServiceHolder {  
 public static final ShowService *INSTANCE* =new ShowService();  
}  
  
public ObservableList<ShowListViewModel>getAllShows()  
{  
 List<Show> shows=repository.getAll();  
  
 return FXCollections.*observableList*(shows.stream().map(s -> new ShowListViewModel(  
 s.getShow\_ID(),  
 s.getShow\_Title(),  
 s.getShow\_Date(),  
 s.getLocation(),  
 s.getMaxPlaces(),  
 s.getTickets(),  
 s.getShow\_status(),  
 s.getShow\_type(),  
 s.getOrganizer(),  
 s.getDistributor()  
 )).collect(Collectors.*toList*()));  
}  
 }

**/Ticket\_Center.idea/src/main/java/bg/tu\_varna/sit/group25/ticketcenter/common**

**>Constants**

package bg.tu\_varna.sit.group25.ticketcenter.Application.common;  
  
public class Constants  
{  
 public static class View  
 {  
 public static final String *HELLO\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/hello-view.fxml";  
 }  
  
 public static class Login  
 {  
 public static final String *LOGIN\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/LoginForm-view.fxml";  
 }  
  
 public static class Configurations  
 {  
 public static final String *LOG4J\_PROPERTIES*="/bg/tu\_varna/sit/group25/ticketcenter/configuration/log4j.properties";  
 }  
  
 public static class Organizer  
 {  
 public static final String *ORGANIZER\_LOGIN*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/OrganizerLogin-view.fxml";  
 }  
  
 public static class Distributor  
 {  
 public static final String *DISTRIBUTOR\_LOGIN*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/DistributorLogin-view.fxml";  
 public static final String *DISTRIBUTOR\_QUERY\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/DistributorQuery-view.fxml";  
 }  
  
 public static class TicketForm  
 {  
 public static final String *TICKET\_FORM\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/TicketForm-view.fxml";  
 }  
  
 public static class RateingForm  
 {  
 public static final String *RATING\_FORM\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/DistributorRateing-view.fxml";  
 }  
  
  
 public static class EventForm  
 {  
 public static final String *EVENT\_FORM\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/EventForm-view.fxml";  
 public static final String *EVENT\_QUERY\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/EventQuery-View.fxml";  
 }  
  
 public static class Warning  
 {  
 public static final String *EMPTY\_FIELDS\_MESSAGE*="Please fill all the fields";  
 public static final String *INVALID\_DATA\_MESSAGE*="Entered data is incorrect";  
 }  
  
 public static class Values  
 {  
 public static final String *TITTLE*="Hello there user";  
 }  
}

**/Ticket\_Center.idea/src/main/java/bg/tu\_varna/sit/group25/ticketcenter/common**

**>Common**

package bg.tu\_varna.sit.group25.ticketcenter.Application.common;  
  
public class Constants  
{  
 public static class View  
 {  
 public static final String *HELLO\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/hello-view.fxml";  
 }  
  
 public static class Login  
 {  
 public static final String *LOGIN\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/LoginForm-view.fxml";  
 }  
  
 public static class Configurations  
 {  
 public static final String *LOG4J\_PROPERTIES*="/bg/tu\_varna/sit/group25/ticketcenter/configuration/log4j.properties";  
 }  
  
 public static class Organizer  
 {  
 public static final String *ORGANIZER\_LOGIN*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/OrganizerLogin-view.fxml";  
 }  
  
 public static class Distributor  
 {  
 public static final String *DISTRIBUTOR\_LOGIN*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/DistributorLogin-view.fxml";  
 public static final String *DISTRIBUTOR\_QUERY\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/DistributorQuery-view.fxml";  
 }  
  
 public static class TicketForm  
 {  
 public static final String *TICKET\_FORM\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/TicketForm-view.fxml";  
 }  
  
 public static class RateingForm  
 {  
 public static final String *RATING\_FORM\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/DistributorRateing-view.fxml";  
 }  
  
  
 public static class EventForm  
 {  
 public static final String *EVENT\_FORM\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/EventForm-view.fxml";  
 public static final String *EVENT\_QUERY\_VIEW*="/bg/tu\_varna/sit/group25/ticketcenter/presentation.views/EventQuery-View.fxml";  
 }  
  
 public static class Warning  
 {  
 public static final String *EMPTY\_FIELDS\_MESSAGE*="Please fill all the fields";  
 public static final String *INVALID\_DATA\_MESSAGE*="Entered data is incorrect";  
 }  
  
 public static class Values  
 {  
 public static final String *TITTLE*="Hello there user";  
 }  
}

**/Ticket\_Center.idea/src/main/java/bg/tu\_varna/sit/group25/ticketcenter/data/access**

**>Connection**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.access;  
  
import org.apache.log4j.Logger;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
  
import java.sql.DriverManager;  
  
  
public class Connection  
{  
  
 private static final Logger *log*=Logger.*getLogger*(Connection.class);  
 private static SessionFactory *sessionFactory*;  
  
 static  
 {  
 try {  
 *sessionFactory*=new Configuration().configure().buildSessionFactory();  
 }  
 catch (Throwable ex)  
 {  
 *log*.error("Initial SessionFactory created fail"+ex);  
  
 }  
  
 }  
  
 public static Session openSession()  
 {  
 return *sessionFactory*.openSession();  
 }  
  
 public static void openSessionClose() {  
 *sessionFactory*.close();  
 }  
}

**/Ticket\_Center.idea/src/main/java/bg/tu\_varna/sit/group25/ticketcenter/data/entities**

**Чрез entities достъпваме таблиците в базата данни и обозначаваме на връзките между таблиците в базата данни**

**>Customer** package bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities;  
  
import javax.persistence.\*;  
import java.io.Serializable;  
import java.util.Set;  
  
@Table(name="customer")  
@Entity  
  
public class Customer implements Serializable{  
private static final long *serialVersionUID*=1L;  
  
  
 @Id  
 @GeneratedValue(strategy=GenerationType.*IDENTITY*)  
 @JoinColumn(name="Customer\_ID",nullable = false)  
 private Long Customer\_ID;  
  
  
 @Column(name = "Customer\_Name",table = "customer")  
 private String Customer\_Name;  
  
  
 @Column(name = "Phone\_Number",table = "customer")  
 private String Phone\_Number;  
  
  
 @Column(name = "Ticket\_Limit",table = "customer")  
 private int Ticket\_Limit;  
  
 @OneToMany(mappedBy = "customer")  
 private Set<Form> forms;  
  
  
 public Long getCustomer\_ID() {  
 return Customer\_ID;  
 }  
  
 public void setCustomer\_ID(Long customer\_ID) {  
 Customer\_ID = customer\_ID;  
 }  
  
 public String getCustomer\_Name() {  
 return Customer\_Name;  
 }  
  
 public void setCustomer\_Name(String customer\_Name) {  
 Customer\_Name = customer\_Name;  
 }  
  
 public String getPhone\_Number() {  
 return Phone\_Number;  
 }  
  
 public void setPhone\_Number(String phone\_Number) {  
 Phone\_Number = phone\_Number;  
 }  
  
 public int getTicket\_Limit() {  
 return Ticket\_Limit;  
 }  
  
 public void setTicket\_Limit(int ticket\_Limit) {  
 Ticket\_Limit = ticket\_Limit;  
 }  
  
 public Set<Form> getForms() {  
 return forms;  
 }  
  
 public void setForms(Set<Form> forms) {  
 this.forms = forms;  
 }  
  
 @Override  
 public String toString() {  
 return "Customer{" +  
 "Customer\_ID=" + Customer\_ID +  
 ", Customer\_Name='" + Customer\_Name + '\'' +  
 ", Phone\_Number='" + Phone\_Number + '\'' +  
 ", Ticket\_Limit=" + Ticket\_Limit +  
 ", forms=" + forms +  
 '}';  
 }  
}

**>Distributor**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities;  
import javax.persistence.\*;  
import java.util.List;  
import java.util.Set;  
  
@Table(name="distributor",schema = "ticketcenter")  
@Entity  
public class Distributor {  
 @Id  
 @JoinColumn(name="Distributor\_ID",nullable = false)  
 private int Distributor\_ID;  
  
 @JoinColumn(name="Name\_D",nullable = false)  
 private String Name\_D;  
  
 @JoinColumn(name="Phone\_Number\_D",nullable = false)  
 private String Phone\_Number\_D;  
  
 @Column(name="Fee\_D",nullable = false)  
 private double Fee\_D;  
  
 @Column(name="Rating",nullable = false)  
 private double Rating;  
  
 @Column(name="Rating\_Count",nullable = false)  
 private int Rating\_Count;  
  
 @Column(name = "Sold\_Tickets\_D",nullable = false)  
 private int Sold\_Tickets\_D;  
  
 @Column(name="UserName\_D",nullable = false)  
 private String UserName\_D;  
  
 @Column(name="Password\_D",nullable = false)  
 private String Password\_D;  
  
 @OneToMany(mappedBy ="distributor")  
 private Set<Show> showSet;  
  
 @OneToMany(mappedBy ="distributor")  
 private Set<Form>formSet;  
  
 public Distributor(String userName\_D, String password\_D) {  
 UserName\_D = userName\_D;  
 Password\_D = password\_D;  
 }  
 public Distributor(){}  
  
  
 public int getSold\_Tickets\_D() {  
 return Sold\_Tickets\_D;  
 }  
  
 public void setSold\_Tickets\_D(int sold\_Tickets\_D) {  
 Sold\_Tickets\_D = sold\_Tickets\_D;  
 }  
  
 public Set<Show> getShowSet() {  
 return showSet;  
 }  
  
 public void setShowSet(Set<Show> showSet) {  
 this.showSet = showSet;  
 }  
  
 public Set<Form> getFormSet() {  
 return formSet;  
 }  
  
 public void setFormSet(Set<Form> formSet) {  
 this.formSet = formSet;  
 }  
  
 public int getDistributor\_ID() {  
 return Distributor\_ID;  
 }  
  
 public void setDistributor\_ID(int distributor\_ID) {  
 Distributor\_ID = distributor\_ID;  
 }  
  
 public String getName\_D() {  
 return Name\_D;  
 }  
  
 public void setName\_D(String name\_D) {  
 Name\_D = name\_D;  
 }  
  
 public String getPhone\_Number\_D() {  
 return Phone\_Number\_D;  
 }  
  
 public void setPhone\_Number\_D(String phone\_Number\_D) {  
 Phone\_Number\_D = phone\_Number\_D;  
 }  
  
 public double getFee\_D() {  
 return Fee\_D;  
 }  
  
 public void setFee\_D(double fee\_D) {  
 Fee\_D = fee\_D;  
 }  
  
 public double getRating() {  
 return Rating;  
 }  
  
 public void setRating(double rating) {  
 Rating = rating;  
 }  
  
 public double getRating\_Count() {  
 return Rating\_Count;  
 }  
  
 public void setRating\_Count(int rating\_Count) {  
 Rating\_Count = rating\_Count;  
 }  
  
 public String getUserName\_D() {  
 return UserName\_D;  
 }  
  
 public void setUserName\_D(String userName\_D) {  
 UserName\_D = userName\_D;  
 }  
  
 public String getPassword\_D() {  
 return Password\_D;  
 }  
  
 public void setPassword\_D(String password\_D) {  
 Password\_D = password\_D;  
 }  
  
 @Override  
 public String toString() {  
 return "Distributor{" +  
 "Distributor\_ID=" + Distributor\_ID +  
 ", Name\_D='" + Name\_D + '\'' +  
 ", Phone\_Number\_D='" + Phone\_Number\_D + '\'' +  
 ", Fee\_D=" + Fee\_D +  
 ", Rating=" + Rating +  
 ", Rating\_Count=" + Rating\_Count +  
 ", Sold\_Tickets\_D=" + Sold\_Tickets\_D +  
 ", UserName\_D='" + UserName\_D + '\'' +  
 ", Password\_D='" + Password\_D + '\'' +  
 ", showSet=" + showSet +  
 ", formSet=" + formSet +  
 '}';  
 }  
}

**>Form**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities;  
  
import javax.persistence.\*;  
import java.util.Set;  
  
@Table(name="Form")  
@Entity  
public class Form {  
  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name="Form\_ID",nullable = false)  
 private int Form\_Id;  
  
 @Column(name= "Sold\_Tickets",nullable = false)  
 private int Sold\_Tickets;  
  
 @Column(name="Price\_Ticket",nullable = false)  
 private double Price\_Ticket;  
  
 @ManyToOne  
 @JoinColumn(name="Distributor\_ID",nullable = false)  
 private Distributor distributor;  
  
 @ManyToOne  
 @JoinColumn(name="Customer\_ID",nullable = false)  
 private Customer customer;  
  
 @ManyToOne  
 @JoinColumn(name="Show\_ID",nullable = false)  
 private Show show;  
  
 public Form() {  
 }  
  
 public int getForm\_Id() {  
 return Form\_Id;  
 }  
  
 public void setForm\_Id(int form\_Id) {  
 Form\_Id = form\_Id;  
 }  
  
 public int getSold\_Tickets() {  
 return Sold\_Tickets;  
 }  
  
 public void setSold\_Tickets(int sold\_Tickets) {  
 Sold\_Tickets = sold\_Tickets;  
 }  
  
 public double getPrice\_Ticket() {  
 return Price\_Ticket;  
 }  
  
 public void setPrice\_Ticket(double price\_Ticket) {  
 Price\_Ticket = price\_Ticket;  
 }  
  
 public Distributor getDistributor() {  
 return distributor;  
 }  
  
 public void setDistributor(Distributor distributor) {  
 this.distributor = distributor;  
 }  
  
 public Customer getCustomer() {  
 return customer;  
 }  
  
 public void setCustomer(Customer customer) {  
 this.customer = customer;  
 }  
  
 public Show getShow() {  
 return show;  
 }  
  
 public void setShow(Show show) {  
 this.show = show;  
 }  
  
 @Override  
 public String toString() {  
 return "Form{" +  
 "Form\_Id=" + Form\_Id +  
 ", Sold\_Tickets=" + Sold\_Tickets +  
 ", Price\_Ticket=" + Price\_Ticket +  
 ", distributor=" + distributor +  
 ", customer=" + customer +  
 ", show=" + show +  
 '}';  
 }  
}

**>Organizer**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities;  
  
import javax.persistence.\*;  
import java.io.Serializable;  
import java.util.List;  
@Entity  
@Table(name="organizer")  
public class Organizer implements Serializable {  
 private static final long *serialVersionUID*=1L;  
  
  
 @Id  
 @GeneratedValue(strategy=GenerationType.*IDENTITY*)  
 @JoinColumn(name = "Organizer\_ID",nullable = false)  
 private int Organizer\_ID;  
  
 @Column(name = "Name\_O",nullable = false)  
 private String Name\_O;  
  
 @Column(name = "Phone\_Number\_O",nullable = false)  
 private String Phone\_Number\_O;  
  
 @Column(name = "Fee\_O",nullable = false)  
 private double Fee\_O;  
  
 @Column(name = "UserName\_O",nullable = false)  
 private String UserName\_O;  
  
 @Column(name = "Password\_O",nullable = false)  
 private int Password\_O;  
  
 @OneToMany(mappedBy = "organizer")  
 private List<Show> shows;  
  
 public Organizer(int organizer\_ID, String name\_O, String phone\_Number\_O, double fee\_O, String userName\_O, int password\_O) {  
 this.Organizer\_ID = organizer\_ID;  
 this.Name\_O = name\_O;  
 this.Phone\_Number\_O = phone\_Number\_O;  
 this.Fee\_O = fee\_O;  
 this.UserName\_O = userName\_O;  
 this.Password\_O = password\_O;  
 }  
  
 public Organizer() {  
 }  
  
 public int getOrganizer\_ID() {  
 return Organizer\_ID;  
 }  
  
 public void setOrganizer\_ID(int organizer\_ID) {  
 Organizer\_ID = organizer\_ID;  
 }  
  
 public String getName\_O() {  
 return Name\_O;  
 }  
  
 public void setName\_O(String name\_O) {  
 Name\_O = name\_O;  
 }  
  
 public String getPhone\_Number\_O() {  
 return Phone\_Number\_O;  
 }  
  
 public void setPhone\_Number\_O(String phone\_Number\_O) {  
 Phone\_Number\_O = phone\_Number\_O;  
 }  
  
 public double getFee\_O() {  
 return Fee\_O;  
 }  
  
 public void setFee\_O(double fee\_O) {  
 Fee\_O = fee\_O;  
 }  
  
 public String getUserName\_O() {  
 return UserName\_O;  
 }  
  
 public void setUserName\_O(String userName\_O) {  
 UserName\_O = userName\_O;  
 }  
  
 public int getPassword\_O() {  
 return Password\_O;  
 }  
  
 public void setPassword\_O(int password\_O) {  
 Password\_O = password\_O;  
 }  
  
 public List<Show> getShows() {  
 return shows;  
 }  
  
 public void setShows(List<Show> shows) {  
 this.shows = shows;  
 }  
  
 @Override  
 public String toString() {  
 return "Organizer{" +  
 "Organizer\_ID=" + Organizer\_ID +  
 ", Name\_O='" + Name\_O + '\'' +  
 ", Phone\_Number\_O='" + Phone\_Number\_O + '\'' +  
 ", Fee\_O=" + Fee\_O +  
 ", UserName\_O='" + UserName\_O + '\'' +  
 ", Password\_O='" + Password\_O + '\'' +  
 ", shows=" + shows +  
 '}';  
 }  
}

**>Show**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.common.Constants;  
  
import java.util.GregorianCalendar;  
import javax.persistence.\*;  
  
@Entity  
@Table(name = "show")  
public class Show {  
 private static final long *serialVersionUID*=1L;  
  
  
 @Id  
 @GeneratedValue(strategy=GenerationType.*IDENTITY*)  
 @Column(name = "Show\_ID",table = "show")  
 private int Show\_ID;  
  
 @Column(name = "Show\_Title",table ="show")  
 private String Show\_Title;  
  
 @Column(name = "Show\_Date",table = "show")  
 private GregorianCalendar Show\_Date;  
  
 @Column(name = "Location",table = "show")  
 private String Location;  
  
 @Column(name = "MaxPlaces",table = "show")  
 private int MaxPlaces;  
  
 @Column(name = "Tickets",table = "show")  
 private int Tickets;  
  
 @ManyToOne  
 @JoinColumn(name = "Show\_Status\_ID",nullable = false)  
 private Show\_Status show\_status;  
  
 @ManyToOne  
 @JoinColumn(name = "Show\_Type\_ID",nullable = false)  
 private Show\_Type show\_type;  
  
 @ManyToOne  
 @JoinColumn(name = "Organizer\_ID",nullable = false)  
 private Organizer organizer;  
  
 @ManyToOne  
 @JoinColumn(name = "Distributor\_ID",nullable = false)  
 private Distributor distributor;  
  
 public int getShow\_ID() {  
 return Show\_ID;  
 }  
  
 public void setShow\_ID(int show\_ID) {  
 Show\_ID = show\_ID;  
 }  
  
 public String getShow\_Title() {  
 return Show\_Title;  
 }  
  
 public void setShow\_Title(String show\_Title) {  
 Show\_Title = show\_Title;  
 }  
  
 public GregorianCalendar getShow\_Date() {  
 return Show\_Date;  
 }  
  
 public void setShow\_Date(GregorianCalendar show\_Date) {  
 Show\_Date = show\_Date;  
 }  
  
 public String getLocation() {  
 return Location;  
 }  
  
 public void setLocation(String location) {  
 Location = location;  
 }  
  
 public int getMaxPlaces() {  
 return MaxPlaces;  
 }  
  
 public void setMaxPlaces(int maxPlaces) {  
 MaxPlaces = maxPlaces;  
 }  
  
 public int getTickets() {  
 return Tickets;  
 }  
  
 public void setTickets(int tickets) {  
 Tickets = tickets;  
 }  
  
 public Show\_Status getShow\_status() {  
 return show\_status;  
 }  
  
 public void setShow\_status(Show\_Status show\_status) {  
 this.show\_status = show\_status;  
 }  
  
 public Show\_Type getShow\_type() {  
 return show\_type;  
 }  
  
 public void setShow\_type(Show\_Type show\_type) {  
 this.show\_type = show\_type;  
 }  
  
 public Organizer getOrganizer() {  
 return organizer;  
 }  
  
 public void setOrganizer(Organizer organizer) {  
 this.organizer = organizer;  
 }  
  
 public Distributor getDistributor() {  
 return distributor;  
 }  
  
 public void setDistributor(Distributor distributor) {  
 this.distributor = distributor;  
 }  
  
 @Override  
 public String toString() {  
 return "Show{" +  
 "Show\_ID=" + Show\_ID +  
 ", Show\_Title='" + Show\_Title + '\'' +  
 ", Show\_Date=" + Show\_Date +  
 ", Location='" + Location + '\'' +  
 ", MaxPlaces=" + MaxPlaces +  
 ", Tickets=" + Tickets +  
 ", show\_status=" + show\_status +  
 ", show\_type=" + show\_type +  
 ", organizer=" + organizer +  
 ", distributor=" + distributor +  
 '}';  
 }  
}

**>Show\_Status**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities;  
import javax.persistence.\*;  
import java.io.Serializable;  
import java.util.Set;  
  
@Table(name="show\_status")  
@Entity  
public class Show\_Status implements Serializable {  
 private static final long *serialVersionUID*=1L;  
  
 @Id  
 @GeneratedValue(strategy=GenerationType.*IDENTITY*)  
 @JoinColumn(name="Show\_Status\_ID",nullable = false)  
 private long Show\_Status\_ID;  
  
 @Column(name="Status",nullable = false)  
 private String Status;  
  
  
 @OneToMany(mappedBy = "show\_status")  
 private Set<Show> showSet;  
  
 public long getShow\_Status\_ID() {  
 return Show\_Status\_ID;  
 }  
  
 public void setShow\_Status\_ID(long show\_Status\_ID) {  
 Show\_Status\_ID = show\_Status\_ID;  
 }  
  
 public String getStatus() {  
 return Status;  
 }  
  
 public void setStatus(String status) {  
 Status = status;  
 }  
  
 public Set<Show> getShowSet() {  
 return showSet;  
 }  
  
 public void setShowSet(Set<Show> showSet) {  
 this.showSet = showSet;  
 }  
  
 @Override  
 public String toString() {  
 return "Show\_Status{" +  
 "Show\_Status\_ID=" + Show\_Status\_ID +  
 ", Status='" + Status + '\'' +  
 ", showSet=" + showSet +  
 '}';  
 }  
}

**>Show\_Type**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities;  
  
import javax.persistence.\*;  
import java.io.Serializable;  
import java.util.List;  
import java.util.Set;  
  
@Table(name="show\_type")  
@Entity  
public class Show\_Type implements Serializable {  
 private static final long *serialVersionUID*=1L;  
  
 @Id  
 @GeneratedValue(strategy=GenerationType.*IDENTITY*)  
 @JoinColumn(name="Show\_Type\_ID",nullable = false)  
 private int Show\_Type\_ID;  
  
 @Column(name="Type",nullable = false)  
 private int Type;  
  
 @OneToMany(fetch = FetchType.*LAZY*,mappedBy = "show\_type")  
 private Set<Show> showSet;  
  
 public int getShow\_Type\_ID() {  
 return Show\_Type\_ID;  
 }  
  
 public void setShow\_Type\_ID(int show\_Type\_ID) {  
 Show\_Type\_ID = show\_Type\_ID;  
 }  
  
 public int getType() {  
 return Type;  
 }  
  
 public void setType(int type) {  
 Type = type;  
 }  
  
 public Set<Show> getShowSet() {  
 return showSet;  
 }  
  
 public void setShowSet(Set<Show> showSet) {  
 this.showSet = showSet;  
 }  
  
 @Override  
 public String toString() {  
 return "Show\_Type{" +  
 "Show\_Type\_ID=" + Show\_Type\_ID +  
 ", Type=" + Type +  
 ", showSet=" + showSet +  
 '}';  
 }  
}

**/Ticket\_Center.idea/src/main/java/bg/tu\_varna/sit/group25/ticketcenter/data/repositories**

**Всяко репоситори, имплементира интерфесът DAORepository**

**>CustomerRepository**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.access.Connection;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import javafx.concurrent.Task;  
import org.apache.log4j.Logger;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
import java.util.LinkedList;  
import java.util.List;  
import java.util.Optional;  
  
public class CustomerRepository implements DAORepository<Customer>{  
 private static final Logger *log*=Logger.*getLogger*(CustomerRepository.class);  
  
 public static CustomerRepository getInstance()  
 {  
 return CustomerRepository.CustomerRepositoryHolder.*INSTANCE*;  
 }  
  
 private static class CustomerRepositoryHolder  
 {  
 public static final CustomerRepository *INSTANCE*=new CustomerRepository();  
 }  
  
 @Override  
 public void save(Customer obj)  
 {  
 Session session= Connection.*openSession*();  
 Transaction transaction=session.beginTransaction();  
 try {  
 session.save(obj);  
 *log*.info("Customer saved successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Customer save error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public void update(Customer obj) {  
 Session session1= Connection.*openSession*();  
 Transaction transaction=session1.beginTransaction();  
 try {  
 session1.update(obj);  
 *log*.info("Customer updated successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Customer update error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
  
 }  
  
 @Override  
 public void delete(Customer obj) {  
 Session session3= Connection.*openSession*();  
 Transaction transaction=session3.beginTransaction();  
 try {  
 session3.delete(obj);  
 *log*.info("Customer deleted successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Customer deleted error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
  
 }  
  
 @Override  
 public Optional<Customer> getById(int id) {  
 return Optional.*empty*();  
 }  
  
  
  
 @Override  
 public List<Customer> getAll() {  
 Session session=Connection.*openSession*();  
 Transaction transaction= session.beginTransaction();  
 List<Customer> customers=new LinkedList<>();  
 try {  
 String jpql = "SELECT t FROM Customer t";  
 customers.addAll(session.createQuery(jpql, Customer.class).getResultList());  
 *log*.info("get all customers");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Get Customer error: "+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 return customers;  
 }  
}

**>DAORepository**

void save(T obj);  
void update(T obj);  
void delete(T obj);  
Optional<T> getById(int id);  
List<T>getAll();

**>DistributorRepository**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.common.Constants;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.access.Connection;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Distributor;  
import org.apache.log4j.Logger;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
import java.util.LinkedList;  
import java.util.List;  
import java.util.Optional;  
  
public class DistributorRepository implements DAORepository<Distributor>  
{  
 private static final Logger *log*=Logger.*getLogger*(CustomerRepository.class);  
  
 public static DistributorRepository getInstance()  
 {  
 return DistributorRepository.DistributorRepositoryHolder.*INSTANCE*;  
 }  
 private static class DistributorRepositoryHolder  
 {  
 public static final DistributorRepository *INSTANCE*=new DistributorRepository();  
 }  
  
 @Override  
 public void save(Distributor obj) {  
 Session session= Connection.*openSession*();  
 Transaction transaction=session.beginTransaction();  
 try {  
 session.save(obj);  
 *log*.info("Distributor saved successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Distributor save error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public void update(Distributor obj) {  
 Session session1= Connection.*openSession*();  
 Transaction transaction=session1.beginTransaction();  
 try {  
 session1.update(obj);  
 *log*.info("Distributor updated successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Distributor update error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public void delete(Distributor obj) {  
 Session session3= Connection.*openSession*();  
 Transaction transaction=session3.beginTransaction();  
 try {  
 session3.delete(obj);  
 *log*.info("Distributor deleted successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Distributor deleted error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
  
 }  
  
 @Override  
 public Optional<Distributor> getById(int id) {  
 return Optional.*empty*();  
 }  
  
 @Override  
 public List<Distributor> getAll() {  
  
 Session session=Connection.*openSession*();  
 Transaction transaction= session.beginTransaction();  
 List<Distributor> distributors=new LinkedList<>();  
 try {  
 String jpql = "SELECT t FROM Distributor t";  
 distributors.addAll(session.createQuery(jpql, Distributor.class).getResultList());  
 *log*.info("get all customers");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Get Customer error: "+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 return distributors;  
 }  
  
  
  
}

**>FormRepository**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.access.Connection;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Form;  
import org.apache.log4j.Logger;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
import javax.persistence.EntityManager;  
import javax.persistence.PersistenceContext;  
import java.util.LinkedList;  
import java.util.List;  
import java.util.Optional;  
  
public class FormRepository implements DAORepository<Form>  
{  
 private static final Logger *log*=Logger.*getLogger*(CustomerRepository.class);  
  
 public static FormRepository getInstance()  
 {  
 return FormRepository. FormRepositoryHolder.*INSTANCE*;  
 }  
  
 private static class FormRepositoryHolder  
 {  
 public static final FormRepository *INSTANCE*=new FormRepository();  
 }  
  
 @PersistenceContext  
 private EntityManager entityManager;  
  
 public void insertForm(Form form)  
 {  
 entityManager.createNativeQuery("INSERT INTO FROM (Sold\_Tickets,Price\_Ticket,Distributor\_ID,Customer\_ID,Show\_ID) VALUES (?,?,?,?,?)")  
 .setParameter(1,form.getSold\_Tickets())  
 .setParameter(2,form.getPrice\_Ticket())  
 .setParameter(3,form.getDistributor().getDistributor\_ID())  
 .setParameter(4,form.getCustomer().getCustomer\_ID())  
 .setParameter(5,form.getShow().getShow\_ID())  
 .executeUpdate();  
 }  
  
 public void insertF(Form form)  
 {  
 this.entityManager.persist(form);  
 }  
  
  
 @Override  
 public void save(Form obj) {  
 Session session= Connection.*openSession*();  
 Transaction transaction=session.beginTransaction();  
 try {  
 session.save(obj);  
 *log*.info("Form saved successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error(" Form save error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public void update(Form obj) {  
 Session session1= Connection.*openSession*();  
 Transaction transaction=session1.beginTransaction();  
 try {  
 session1.update(obj);  
 *log*.info("Form updated successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Form update error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
  
 }  
  
 @Override  
 public void delete(Form obj) {  
 Session session3= Connection.*openSession*();  
 Transaction transaction=session3.beginTransaction();  
 try {  
 session3.delete(obj);  
 *log*.info("Form deleted successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Form deleted error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
  
 }  
  
 @Override  
 public Optional<Form> getById(int id) {  
 return Optional.*empty*();  
 }  
  
 @Override  
 public List<Form> getAll() {  
 Session session=Connection.*openSession*();  
 Transaction transaction= session.beginTransaction();  
 List<Form> forms=new LinkedList<>();  
 try {  
 String jpql = "SELECT t FROM Form t";  
 forms.addAll(session.createQuery(jpql, Form.class).getResultList());  
 *log*.info("get all Forms");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Get Form error: "+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 return forms;  
 }  
}

**>OrganizerRepository**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.access.Connection;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Organizer;  
import org.apache.log4j.Logger;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
import java.util.LinkedList;  
import java.util.List;  
import java.util.Optional;  
  
public class OrganizerRepository implements DAORepository<Organizer>  
{  
 private static class OrganizerRepositoryHolder  
 {  
 public static final OrganizerRepository *INSTANCE*=new OrganizerRepository();  
 }  
 public static OrganizerRepository getInstance()  
 {  
 return OrganizerRepository.OrganizerRepositoryHolder.*INSTANCE*;  
 }  
 private static final Logger *log*=Logger.*getLogger*(OrganizerRepository.class);  
  
 @Override  
 public void save(Organizer obj) {  
 Session session= Connection.*openSession*();  
 Transaction transaction=session.beginTransaction();  
 try {  
 session.save(obj);  
 *log*.info("Organizer saved successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Organizer save error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public void update(Organizer obj) {  
 Session session1= Connection.*openSession*();  
 Transaction transaction=session1.beginTransaction();  
 try {  
 session1.update(obj);  
 *log*.info("Organizer updated successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Organizer update error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public void delete(Organizer obj) {  
 Session session3= Connection.*openSession*();  
 Transaction transaction=session3.beginTransaction();  
 try {  
 session3.delete(obj);  
 *log*.info("Distributor deleted successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Distributor deleted error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public Optional<Organizer> getById(int id) {  
 return Optional.*empty*();  
 }  
  
 @Override  
 public List<Organizer> getAll() {  
 Session session=Connection.*openSession*();  
 Transaction transaction= session.beginTransaction();  
 List<Organizer> organizers=new LinkedList<>();  
 try {  
 String jpql = "SELECT t FROM Organizer t";  
 organizers.addAll(session.createQuery(jpql, Organizer.class).getResultList());  
 *log*.info("get all customers");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Get Customer error: "+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 return organizers;  
 }  
  
  
  
}

**>ShowRepository**

package bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.access.Connection;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Organizer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Show;  
import org.apache.log4j.Logger;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
import java.net.PortUnreachableException;  
import java.util.LinkedList;  
import java.util.List;  
import java.util.Optional;  
  
public class ShowRepository implements DAORepository<Show>  
{  
  
 private static class ShowRepositoryHolder  
 {  
 public static final ShowRepository *INSTANCE*=new ShowRepository();  
 }  
 public static ShowRepository getInstance()  
 {  
 return ShowRepository.ShowRepositoryHolder.*INSTANCE*;  
 }  
 private static final Logger *log*=Logger.*getLogger*(ShowRepository.class);  
  
 @Override  
 public void save(Show obj) {  
 Session session= Connection.*openSession*();  
 Transaction transaction=session.beginTransaction();  
 try {  
 session.save(obj);  
 *log*.info("Show saved successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Show save error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public void update(Show obj) {  
 Session session= Connection.*openSession*();  
 Transaction transaction=session.beginTransaction();  
 try {  
 session.update(obj);  
 *log*.info("Show saved successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Show save error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public void delete(Show obj) {  
 Session session= Connection.*openSession*();  
 Transaction transaction=session.beginTransaction();  
 try {  
 session.delete(obj);  
 *log*.info("Show saved successfully");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Show save error"+ex.getMessage());  
 }  
 finally {  
 transaction.commit();  
 }  
 }  
  
 @Override  
 public Optional<Show> getById(int id) {  
 return Optional.*empty*();  
 }  
  
  
 @Override  
 public List<Show> getAll()  
 {  
 Session session1=Connection.*openSession*();  
 Transaction transaction= session1.beginTransaction();  
 List<Show> shows=new LinkedList<>();  
 try  
 {  
 String jpql = "SELECT s FROM Show s ";  
 shows.addAll(session1.createQuery(jpql, Show.class).getResultList());  
 *log*.info("get all customers");  
 }  
 catch (Exception ex)  
 {  
 *log*.error("Get Customer error: "+ex.getMessage());  
 }  
 finally  
 {  
 transaction.commit();  
 }  
 return shows;  
 }  
}

**/Ticket\_Center.idea/src/main/java/bg/tu\_varna/sit/group25/ticketcenter/presentation/controllers**

**>DistributorLoginController**

**Чрез DistributorLoginController достъпваме стредата за вписване, като Разпространител.** package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.common.Constants;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.access.Connection;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.DistributorRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.DistributorListViewModel;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.LoginUtls;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.DistributorService;  
import javafx.collections.ObservableList;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.PasswordField;  
import javafx.scene.control.TextField;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
import java.net.URL;  
import java.sql.DatabaseMetaData;  
import java.util.ResourceBundle;  
  
public class DistributorLoginController implements Initializable {  
 Session session8 = Connection.*openSession*();  
 Transaction transaction = session8.beginTransaction();  
 public static DistributorService *service* = new DistributorService();  
 private static boolean *log* = true;  
  
  
 @FXML  
 private TextField tfUserDis;  
 @FXML  
 private PasswordField tfPassDis;  
 @FXML  
 private Label lbDisWarning;  
 @FXML  
 private Button DistributorLoginButton;  
 @Override  
 public void initialize(URL url, ResourceBundle resourceBundle) {  
 DistributorLoginButton.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 // DistributorListViewModel distributorListViewModel=new DistributorListViewModel();  
 // ObservableList<DistributorListViewModel> distributor= service.getName();  
 // ObservableList<DistributorListViewModel> distributor1= service.getPass();  
 if (tfPassDis.getText().isBlank()==false&&tfUserDis.getText().isBlank()==false) {  
 // if (tfUserDis.getText().equals(service.getName())) {  
  
  
 LoginUtls.*DisLogin*(event, Constants.TicketForm.*TICKET\_FORM\_VIEW*, "Ticket Form");  
 // }  
 }  
 else {  
 lbDisWarning.setText(Constants.Warning.*EMPTY\_FIELDS\_MESSAGE*);  
 }  
 }  
 });  
 }  
  
}

**>DistributorQueryController**

**Чрез DistributorQueryController**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.FormService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.FormRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.FormListViewModel;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.FormUtils;  
import javafx.collections.ObservableList;  
import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.Label;  
import javafx.scene.control.ListView;  
  
import java.lang.ref.PhantomReference;  
import java.net.URL;  
import java.util.ResourceBundle;  
  
public class DistributorQueryController implements Initializable {  
  
 private final FormService service=FormService.*getInstance*();  
 @FXML  
 private Label lbDisQry;  
 @FXML  
 private ListView<FormListViewModel> lvDistributor;  
  
 private String disName;  
 @Override  
 public void initialize(URL url, ResourceBundle resourceBundle) {  
  
 FormUtils.*DisQuery*(disName);  
 }  
  
 public void SetUserInfo(String username)  
 {  
 lbDisQry.setText("Distributor: "+username);  
 disName=username;  
  
 }  
}

**>DistributorRatingController**

**Чрез DistributorRatingController въпеждаме оценка на разпространителя, който ни е продал билетите само след, като заявката за от TicketFormControler е изпънена**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.common.Constants;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.FormUtils;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.LoginUtls;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.Button;  
import javafx.scene.control.TextField;  
  
import java.net.URL;  
import java.util.ResourceBundle;  
  
public class DistributorRateingController implements Initializable {  
 @FXML  
 private Button buttonSubmmitRating;  
 @FXML  
 private TextField tfRateing;  
  
 private String disName;  
 @Override  
 public void initialize(URL url, ResourceBundle resourceBundle) {  
 buttonSubmmitRating.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 if (tfRateing.getText()!="") {  
 FormUtils.*DisRateing*(tfRateing.getText(),disName);  
 }  
 LoginUtls.*chooseLogin*(event, Constants.Login.*LOGIN\_VIEW*,"Login",null,null);  
 }  
 });  
 }  
 public void GetInfo(String name)  
 {  
 disName=name;  
 }  
}

**>EventFormController**

**Катп използваме EventFormController,ние задаваме заявка към базата данни за запис на ново събитие, което може да бъде направено само от профил на Организатор**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.CustomerService;  
//import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.FormService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.ShowService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.common.Constants;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.access.Connection;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Form;  
// bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.FormRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Show;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.FormUtils;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.LoginUtls;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.ShowListViewModel;  
import javafx.collections.ObservableList;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
import org.w3c.dom.events.MouseEvent;  
  
import java.net.URL;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.\*;  
  
public class EventFormController implements Initializable {  
  
 private final ShowService service=ShowService.*getInstance*();  
  
 @FXML  
 private TextField tfEventLocation;  
 @FXML  
 private TextField tfEventType;  
 @FXML  
 private TextField tfEventName;  
 @FXML  
 private TextField tfEventStatus;  
 @FXML  
 private TextField tfEventDate;  
 @FXML  
 private TextField tfEventTickets;  
 @FXML  
 private TextField tfEventTicketLimit;  
 @FXML  
 private TextField tfEventID;  
 @FXML  
 private Label lbEventFormWarning;  
 @FXML  
 private Button buttonCreate;  
 @FXML  
 private Button buttonUpdate;  
 @FXML  
 private Button buttonQuery;  
 @FXML  
 private Button buttonLoginReturn;  
 @Override  
 public void initialize(URL url, ResourceBundle resourceBundle) {  
 buttonCreate.setOnAction(new EventHandler<ActionEvent>() {  
  
 @Override  
 public void handle(ActionEvent event) {  
 if (tfEventLocation.getText()!=""&&tfEventType.getText()!=""&&tfEventType.getText()!=""&&tfEventStatus.getText()!=""&&tfEventDate.getText()!=""  
 &&tfEventTickets.getText()!=""&&tfEventTicketLimit.getText()!=""&&tfEventID.getText()!="")  
 {  
 //FormUtils.EventFormCreate(tfEventLocation.getText(),Integer.parseInt(tfEventType.getText()),tfEventName.getText(), Integer.parseInt(tfEventStatus.getText()), tfEventDate.getText(),  
 // Integer.parseInt(tfEventTickets.getText()), Integer.parseInt(tfEventTicketLimit.getText()),Integer.parseInt(tfEventID.getText()));  
 }  
 else  
 {  
 lbEventFormWarning.setText(Constants.Warning.*EMPTY\_FIELDS\_MESSAGE*);  
 }  
  
 }  
 });  
  
 buttonUpdate.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 if (tfEventLocation.getText()!=""&&tfEventType.getText()!=""&&tfEventName.getText()!=""&&tfEventStatus.getText()!=""&&tfEventDate.getText()!=""  
 &&tfEventTickets.getText()!=""&&tfEventTicketLimit.getText()!=""&&tfEventID.getText()!="")  
 {  
 FormUtils.*EventFormUpdate*(tfEventLocation.getText(),tfEventType.getText(),tfEventName.getText(),tfEventStatus.getText(),tfEventDate.getText(),  
 tfEventTickets.getText(), tfEventTicketLimit.getText(),tfEventID.getText());  
 }  
 else  
 {  
 lbEventFormWarning.setText(Constants.Warning.*EMPTY\_FIELDS\_MESSAGE*);  
 }  
 }  
 });  
 buttonQuery.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 if (true)  
 {  
 LoginUtls.*EventQueryEntry*(event,Constants.EventForm.*EVENT\_QUERY\_VIEW*,"Event Query");  
 }  
 else  
 {  
 lbEventFormWarning.setText(Constants.Warning.*EMPTY\_FIELDS\_MESSAGE*);  
 }  
 }  
 });  
 buttonLoginReturn.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 LoginUtls.*chooseLogin*(event,Constants.Login.*LOGIN\_VIEW*,"Login choice",null,null);  
 }  
 });  
 }  
  
}

**>EventQueryController**

**>HеlloControler**

**HеlloControler използваме за визуализация на прототипните заявки, към базата данни**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.CustomerService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.FormService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.ShowService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.\*;  
import javafx.collections.ObservableList;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.scene.Node;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.ListView;  
import javafx.scene.input.MouseEvent;  
  
public class HelloController implements EventHandler <MouseEvent> {  
  
 private final ShowService service=ShowService.*getInstance*();  
  
 private final FormService service1=FormService.*getInstance*();  
  
 private final CustomerService service2=CustomerService.*getInstance*();  
 @FXML  
 private Label welcomeText;  
  
 @FXML  
 private Button lobbyButton;  
  
 @FXML  
 private ListView<FormListViewModel>listView;  
  
  
 @FXML  
 private final LobbyModel model;  
  
 public HelloController()  
 {  
 this.model=new LobbyModel();  
 }  
 @FXML  
 private void initialize()  
 {  
 lobbyButton.setOnMouseClicked(this::handle);  
 }  
  
 @Override  
 public void handle(MouseEvent mouseEvent) {  
 welcomeText.setText(model.getWelcomeMessage());  
  
 ObservableList<FormListViewModel>formListViewModels= service1.getAllForms();  
 listView.setItems(formListViewModels);  
  
 //ObservableList<ShowListViewModel>showsListViewModels=service.getAllShows();  
 //listView.setItems(showsListViewModels);  
  
 }  
}

**>LoginFormControllet**

**LoginFormControllet е средата в която си избираме дали да се логнем, каро Дистрибутор или Организатор**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.OrganizerService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.common.Constants;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Distributor;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Organizer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.OrganizerRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.DistributorListViewModel;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.LoginUtls;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.Button;  
  
import java.net.URL;  
import java.util.ResourceBundle;  
  
public class LoginFormController implements Initializable {  
  
 private Organizer organizer=new Organizer();  
 private Distributor distributor=new Distributor();  
  
 @FXML  
 private Button OrganizerLoginButton;  
 @FXML  
 private Button DistributorLoginButton;  
 @Override  
 public void initialize(URL url, ResourceBundle resourceBundle) {  
  
 OrganizerLoginButton.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent actionEvent) {  
 LoginUtls.*chooseLogin*(actionEvent, Constants.Organizer.*ORGANIZER\_LOGIN*,  
 "OrganizerLogin", String.*valueOf*(organizer.getPassword\_O()), organizer.getUserName\_O());  
 }  
 });  
 DistributorLoginButton.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent actionEvent) {  
  
 LoginUtls.*chooseLogin*(actionEvent, Constants.Distributor.*DISTRIBUTOR\_LOGIN*,  
 "DistributorLogin", distributor.getUserName\_D(), distributor.getPassword\_D());  
  
 }  
 });  
  
 }  
  
  
  
  
}

**>OrganizerLoginControler**

**Средата за вписване, като Организатор**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.common.Constants;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.LoginUtls;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.PasswordField;  
import javafx.scene.control.TextField;  
  
import java.net.URL;  
import java.util.ResourceBundle;  
  
public class OrganizerLoginController implements Initializable {  
 @FXML  
 private Button OrganizerLoginButton;  
 @FXML  
 private TextField tfUserOrg;  
 @FXML  
 private PasswordField tfPassOrg;  
 @FXML  
 private Label lbOrgWarning;  
 @Override  
 public void initialize(URL url, ResourceBundle resourceBundle) {  
 OrganizerLoginButton.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 if (tfPassOrg.getText()!=""&&tfUserOrg.getText()!="")  
 {  
 LoginUtls.*OrgLogin*(event,Constants.EventForm.*EVENT\_FORM\_VIEW*,"Event Form",tfUserOrg.getText(),tfPassOrg.getText());  
 }  
 else {  
 lbOrgWarning.setText(Constants.Warning.*EMPTY\_FIELDS\_MESSAGE*);  
 }  
 }  
 });  
 }  
  
 public void setUserInformation(String username,String pass)  
 {  
 username=tfUserOrg.getText();  
 pass=tfPassOrg.getText();  
 }  
}

**>TicketFormControler**

**Средата, коятято ни осигурява възможност да изпратим заявка, към базата данни за да, запазим продажба**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.CustomerService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.common.Constants;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.access.Connection;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Form;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.CustomerRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.FormRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.FormUtils;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models.LoginUtls;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
import org.hibernate.PropertyValueException;  
  
import java.net.URL;  
import java.time.format.FormatStyle;  
import java.util.ResourceBundle;  
  
public class TicketFormController implements Initializable {  
  
 CustomerService customerService=new CustomerService();  
 FormRepository formRepository=new FormRepository();  
 CustomerRepository customerRepository=new CustomerRepository();  
  
  
 @FXML  
 private Label DistributorInForm;  
 @FXML  
 private TextField tfClientName;  
 @FXML  
 private TextField tfClientPhonenumber;  
 @FXML  
 private TextField tfEvent;  
 @FXML  
 private TextField tfTicketCount;  
 @FXML  
 private TextField tfTicketPrice;  
 @FXML  
 private Button buttonSubmit;  
 @FXML  
 private Button buttonDisQuery;  
 @FXML  
 private Label lbWarning;  
  
 private String disName;  
 @Override  
  
 public void initialize(URL url, ResourceBundle resourceBundle) {  
 buttonSubmit.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 if (tfClientName.getText() != "" && tfClientPhonenumber.getText() != "" && tfTicketCount.getText() != "" && tfEvent.getText() != ""&&tfTicketPrice.getText()!="") {  
 FormUtils.*TicketForm*(tfClientName.getText(),tfClientPhonenumber.getText(),tfEvent.getText(),tfTicketCount.getText(),tfTicketPrice.getText());  
  
 LoginUtls.*DisRateing*(event,Constants.RateingForm.*RATING\_FORM\_VIEW*,"Rating Form",disName);  
 } else {  
 lbWarning.setText(Constants.Warning.*EMPTY\_FIELDS\_MESSAGE*);  
 }  
 }  
  
 });  
 buttonDisQuery.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 LoginUtls.*DisQueryEntry*(event,Constants.Distributor.*DISTRIBUTOR\_QUERY\_VIEW*,"Rating Form",disName);  
 }  
 });  
 }  
 public void SetDisName(String name)  
 {  
 DistributorInForm.setText(DistributorInForm.getText()+" "+name);  
 disName=name;  
 }  
}

**/Ticket\_Center.idea/src/main/java/bg/tu\_varna/sit/group25/ticketcenter/peresentation.models**

**Чрез /---/ListViewModel задаваме формата под който данните ще бъдата показани след изппълнението на заявката “SELECT”**

**>CustomerListViewModel**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models;  
  
public class CustomerListViewModel {  
 private long id;  
 private String ime;  
 private String nomer;  
 private int limit;  
  
 public CustomerListViewModel(long id, String ime, String nomer, int limit) {  
 this.id = id;  
 this.ime = ime;  
 this.nomer = nomer;  
 this.limit = limit;  
 }  
  
  
 @Override  
 public String toString() {  
 return String.*format*( "%d | %s | %s | %d",id, ime, nomer,limit);  
 }  
}

**>DistributorListViewModel**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models;  
  
public class DistributorListViewModel  
{  
 private int id;  
 private String name;  
 private String phone;  
 private double fee;  
 private double rating;  
 private int rating\_count;  
 private String username;  
 private String password;  
  
 public DistributorListViewModel(int id, String name, String phone, double fee, double rating, int rating\_count, String username, String password) {  
 this.id = id;  
 this.name = name;  
 this.phone = phone;  
 this.fee = fee;  
 this.rating = rating;  
 this.rating\_count = rating\_count;  
 this.username = username;  
 this.password = password;  
 }  
  
 public DistributorListViewModel() {  
  
 }  
  
 public DistributorListViewModel(String userName\_d) {  
 }  
  
 public String toString()  
 {return String.*format*(" %d | %s | %s | %f | %f | %d | %s | %s |",id,name,phone,fee,rating,rating\_count,username,password);}  
}

**>FormListViewModel**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Distributor;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Show;  
  
public class FormListViewModel  
{  
 private int sold;  
 private double price;  
 private int distributor;  
 private int customer;  
 private int show;  
  
 public FormListViewModel(int sold, double price, int distributor, int customer, int show) {  
 this.sold = sold;  
 this.price = price;  
 this.distributor = distributor;  
 this.customer = customer;  
 this.show = show;  
 }  
  
 public FormListViewModel(int sold\_tickets, double price\_ticket, Distributor distributor, Customer customer, Show show) {  
 }  
  
 public FormListViewModel(long sold\_tickets, double price\_ticket, int distributor\_id, Long customer\_id, int show\_id) {  
 }  
  
 public String toString()  
 {  
 return String.*format*(" %d | %f | %d | %d | %d |",sold,price,distributor,customer,show);  
 }  
}

**>FormUtils**

**Ползваме FormUtils за да извличаме функцийте за съответния контролер от там и за да предадем малко по изчистен вид при контролерите**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.CustomerService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.ShowService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Form;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.CustomerRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.repositories.FormRepository;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer;  
import javafx.collections.ObservableList;  
import javafx.fxml.FXML;  
import javafx.scene.control.ListView;  
  
import java.util.GregorianCalendar;  
  
public class FormUtils {  
  
 public static void TicketForm(String name,String phoneNumber,String eventName,String ticketCount,String ticketPrice)  
 {//submit the related data to the DB in the correct places  
  
  
  
 }  
  
 public static void EventFormCreate(String Location, int Type, String Name, int Status, String Date, int Tickets, int Ticket\_limit, int ID)  
 {//make DB connection and insert data  
  
  
  
 }  
  
 public static void EventFormUpdate(String Location,String Type,String Name,String Status,String Date,String Tickets,String Ticket\_limit,String ID)  
 {//make DB connection and update data at specified ID  
  
 }  
  
 public static void EventQry()  
 {//make DB data display by eventName  
  
  
  
 }  
  
 public static void DisRateing(String Rateing,String disName)  
 {//add the rateing to disName in the DB  
  
 }  
  
 public static void DisQuery(String name)  
 {//make query by name in the DB  
  
 }  
}

**>LobbyModel**

**Модел от прототипа**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models;  
  
public class LobbyModel  
{  
 private final String welcomeMessage;  
  
 public LobbyModel()  
 {  
 this.welcomeMessage="Hello User, let us get to work";  
 }  
  
 public String getWelcomeMessage()  
 {  
 return welcomeMessage;  
 }  
}

**>LoginUtls**

**В LoginUtls се съдържат връзките за проверка на потребителко име и парола на дистрибуро и организатор, отново за да може в техните контролери да е по изчистено**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.controllers.\*;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.access.Connection;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers.\*;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.DistributorService;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.business.services.OrganizerService;  
import javafx.collections.ObservableList;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Node;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.PasswordField;  
import javafx.scene.control.TextField;  
import javafx.stage.Stage;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
import java.io.IOException;  
  
  
//commented the 5th line //import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Organizer; remove the "//" when fixed  
public class LoginUtls {  
 @FXML  
 private static TextField *tfUserDis*;  
 @FXML  
 private static PasswordField *tfPassDis*;  
 Session session8 = Connection.*openSession*();  
 Transaction transaction = session8.beginTransaction();  
 public static DistributorService *service* = new DistributorService();  
 public static OrganizerService *service1* = new OrganizerService();  
 private static boolean *log* = true;  
  
 public static void EventQueryEntry(ActionEvent event,String fxmlFile,String title)  
 {  
 Parent root = null;  
 try {  
 FXMLLoader loader = new FXMLLoader(LoginUtls.class.getResource(fxmlFile));  
 EventQueryController eventQueryController=loader.getController();  
 root = loader.load();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();  
 stage.setTitle(title);  
 stage.setScene((new Scene(root, 600, 400)));  
 stage.show();  
 }  
 public static void chooseLogin(ActionEvent event, String fxmlFile, String title, String username, String pass) {  
 Parent root = null;  
 if (username != null && pass != null) {  
  
 try {  
 FXMLLoader loader = new FXMLLoader(LoginUtls.class.getResource(fxmlFile));  
 LoginFormController loginFormController = loader.getController();  
 root = loader.load();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 try {  
 root = FXMLLoader.*load*(LoginUtls.class.getResource(fxmlFile));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
 Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();  
 stage.setTitle(title);  
 stage.setScene((new Scene(root, 600, 400)));  
 stage.show();  
 }  
  
  
 public static void OrgLogin(ActionEvent event, String fxmlFile, String title, String username, String pass)  
 {//add DB distributor user and pass check  
 OrganizerListViewModel organizerListViewModel = new OrganizerListViewModel();  
 ObservableList<OrganizerListViewModel> organizer = *service1*.getAllOrganizers();  
 for (OrganizerListViewModel o : organizer) {  
 if (o.equals(organizer)) {  
  
 *log* = true;  
 }  
 }  
 if (*log* == true) {  
 Parent root1 = null;  
 try {  
 FXMLLoader loader = new FXMLLoader(LoginUtls.class.getResource(fxmlFile));  
 LoginFormController loginFormController= loader.getController();  
 root1 = loader.load();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();  
 stage.setTitle(title);  
 stage.setScene((new Scene(root1, 600, 400)));  
 stage.show();  
 }  
 }  
  
  
 public static void DisLogin (ActionEvent event, String fxmlFile, String title, String username, String pass) //add DB distributor user and pass check  
 {  
 DistributorListViewModel distributorListViewModel = new DistributorListViewModel();  
 ObservableList<DistributorListViewModel> distributor = *service*.getAllDistributor();  
 for (DistributorListViewModel o : distributor) {  
 if (o.equals(distributor))  
 {  
  
 *log* = true;  
 }  
 }  
 if (*log* == true) {  
 Parent root = null;  
 try {  
 FXMLLoader loader = new FXMLLoader(LoginUtls.class.getResource(fxmlFile));  
 TicketFormController ticketFormController=loader.getController();  
 ticketFormController.SetDisName(username);  
 root = loader.load();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 finally {  
 Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();  
 stage.setTitle(title);  
 stage.setScene((new Scene(root, 600, 400)));  
 stage.show();  
 }  
 }  
 }  
  
 public static void DisRateing(ActionEvent event,String fxmlFile,String title,String disName)  
 {  
 Parent root = null;  
 if (disName != null ) {  
 try {  
 FXMLLoader loader = new FXMLLoader(LoginUtls.class.getResource(fxmlFile));  
 DistributorRateingController distributorRateingController=loader.getController();  
 distributorRateingController.GetInfo(disName);  
 root = loader.load();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();  
 stage.setTitle(title);  
 stage.setScene((new Scene(root, 600, 400)));  
 stage.show();  
 }  
 }  
 public static void DisQueryEntry(ActionEvent event,String fxmlFile,String title, String disName)  
 {  
 Parent root = null;  
 if (disName != null ) {  
 try {  
 FXMLLoader loader = new FXMLLoader(LoginUtls.class.getResource(fxmlFile));  
 DistributorQueryController distributorQueryController=loader.getController();  
 distributorQueryController.SetUserInfo(disName);  
 root = loader.load();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();  
 stage.setTitle(title);  
 stage.setScene((new Scene(root, 600, 400)));  
 stage.show();  
 }  
 }  
  
 public static void DisLogin(ActionEvent event, String ticketFormView, String ticket\_form) {  
 }  
}

**>OrganizerListViewModel**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models;  
  
public class OrganizerListViewModel  
{  
 private int id;  
 private String name;  
 private String phone;  
 private double fee;  
 private String username;  
 private int password;  
  
 public OrganizerListViewModel(int id, String name, String phone, double fee, String username, int password) {  
 this.id = id;  
 this.name = name;  
 this.phone = phone;  
 this.fee = fee;  
 this.username = username;  
 this.password = password;  
 }  
  
 public OrganizerListViewModel(int organizer\_id, String name\_o, String getPhone\_number\_o, double fee\_o, double fee\_o1, String userName\_o\_o, int password\_o) {  
 }  
  
 public OrganizerListViewModel() {  
  
 }  
  
 public String toString()  
 {  
 return String.*format*("%d | %s | %s | %f | %s | %d |",id,name,phone,fee,username,password);  
 }  
}

**>ShowListViewModel**

package bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.models;  
  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Distributor;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Organizer;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Show\_Status;  
import bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Show\_Type;  
  
import java.util.GregorianCalendar;  
  
public class ShowListViewModel  
{  
 public int s\_id;  
 public String title;  
 public GregorianCalendar date;  
 public String location;  
 public int max;  
 public int tickets;  
 public int status;  
 public int s\_type;  
 public int organizer;  
 public int distributor;  
  
 public ShowListViewModel(int s\_id, String title, GregorianCalendar date, String location, int max, int tickets, int status, int s\_type, int organizer, int distributor) {  
 this.s\_id = s\_id;  
 this.title = title;  
 this.date = date;  
 this.location = location;  
 this.max = max;  
 this.tickets = tickets;  
 this.status = status;  
 this.s\_type = s\_type;  
 this.organizer = organizer;  
 this.distributor = distributor;  
 }  
  
 public ShowListViewModel(int show\_id, String show\_title, GregorianCalendar show\_date, String location, int maxPlaces, int tickets, Show\_Status show\_status, Show\_Type show\_type, Organizer organizer, Distributor distributor) {  
 }  
  
 public String toString()  
 {  
 return String.*format*(" %d | %s | %t | %s | %d | %d | %d | %d | %d | %d ",s\_id,title,date,location,max,title,status,s\_type,organizer,distributor);  
 }  
}

**src/main/java/module-info.java**

module bg.tu\_varna.sit.group25.ticketcenter {  
 requires javafx.controls;  
 requires javafx.fxml;  
 requires javafx.web;  
  
 requires org.controlsfx.controls;  
 requires com.dlsc.formsfx;  
 requires validatorfx;  
 requires org.kordamp.ikonli.javafx;  
 requires org.kordamp.bootstrapfx.core;  
 requires eu.hansolo.tilesfx;  
 requires log4j;  
 requires org.hibernate.orm.core;  
 requires java.persistence;  
 requires java.naming;  
 requires java.sql;  
  
 opens bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities to org.hibernate.orm.core;  
 exports bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities;  
  
 opens bg.tu\_varna.sit.group25.ticketcenter.Application.data.access to org.hibernate.orm.core;  
 exports bg.tu\_varna.sit.group25.ticketcenter.Application.data.access ;  
  
 exports bg.tu\_varna.sit.group25.ticketcenter.Application.application;  
 opens bg.tu\_varna.sit.group25.ticketcenter.Application.application to javafx.fxml;  
  
 exports bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers;  
 opens bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers to javafx.fxml;  
}

**Ticket\_Center/src/main/resources/bg/tu\_varna/sit/group25/ticketcenter/configuration**

log4j.rootLogger=Error,console,file  
  
log4j.appender.console=org.apache.log4j.ConsoleAppender  
log4j.appender.console.layout=org.apache.log4j.PatternLayout  
log4j.appender.console.layout.ConversionPattern=[%t] %-5p %c %x -%m%n  
  
log4j.appender.file=org.apache.log4j.RollingFileAppender  
log4j.appender.file.layout=org.apache.log4j.PatternLayout  
log4j.appender.file.layout.ConversionPattern=[%t] %-5p %c %x -%m%n  
log4j.appender.file.File=application.log

**Ticket\_Center/src/main/resources/bg/tu\_varna/sit/group25/ticketcenter/presentation.views**

**Чрез /----/-view.fxml създаваме UI за осигуряване на по лесна работа са програмата и нейното разбиране, щом е стартирана**

**>DistributorLogin-view.fxml**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="289.0" prefWidth="477.0"  
 xmlns="http://javafx.com/javafx/11.0.2"  
 xmlns:fx="http://javafx.com/fxml/1"  
 fx:controller="bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers.DistributorLoginController">  
 <children>  
 <Label layoutX="59.0" layoutY="90.0" text="Username">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="59.0" layoutY="134.0" text="Password">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Label>  
 <TextField fx:id="tfUserDis" layoutX="159.0" layoutY="88.0">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </TextField>  
 <PasswordField fx:id="tfPassDis" layoutX="159.0" layoutY="136.0">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </PasswordField>  
 <Button fx:id="DistributorLoginButton" layoutX="190.0" layoutY="216.0" mnemonicParsing="false" prefHeight="25.0" prefWidth="142.0" text="Login">  
 <font>  
 <Font size="18.0" />  
 </font></Button>  
 <Label fx:id="lbDisWarning" layoutX="166.0" layoutY="184.0" textFill="RED">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 </children>  
</AnchorPane>

**>DistributorQuery-view.fxml**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/11.0.2"  
 xmlns:fx="http://javafx.com/fxml/1"  
 fx:controller="bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers.DistributorQueryController">  
 <children>  
 <ListView fx:id="lvDistributor" layoutX="53.0" layoutY="162.0" prefHeight="200.0" prefWidth="297.0" />  
 <Label fx:id="lbDisQry" layoutX="44.0" layoutY="58.0" text="Distributor: ">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="53.0" layoutY="133.0" text="Related info">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 </children>  
</AnchorPane>

**>DistributorRateing-view.fxml**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/11.0.2"  
 xmlns:fx="http://javafx.com/fxml/1"  
 fx:controller="bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers.DistributorRateingController">  
 <children>  
 <Button fx:id="buttonSubmmitRating" layoutX="158.0" layoutY="266.0" mnemonicParsing="false" text="Submit">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font></Button>  
 <TextField fx:id="tfRateing" layoutX="172.0" layoutY="188.0">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font></TextField>  
 </children>  
</AnchorPane>

**>EventForm-view.fxml**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="472.0" prefWidth="731.0" xmlns="http://javafx.com/javafx/11.0.2" xmlns:fx="http://javafx.com/fxml/1" fx:controller="bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers.EventFormController">  
 <children>  
 <Label layoutX="52.0" layoutY="43.0" text="Location">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="52.0" layoutY="78.0" text="Type">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="52.0" layoutY="107.0" text="Name">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="52.0" layoutY="143.0" text="Status">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="52.0" layoutY="180.0" text="Date">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="50.0" layoutY="255.0" text="Ticket limit">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="50.0" layoutY="219.0" text="Tickets ">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Label>  
 <TextField fx:id="tfEventLocation" layoutX="137.0" layoutY="41.0" />  
 <TextField fx:id="tfEventType" layoutX="110.0" layoutY="76.0" />  
 <TextField fx:id="tfEventName" layoutX="110.0" layoutY="108.0" />  
 <TextField fx:id="tfEventStatus" layoutX="110.0" layoutY="144.0" />  
 <TextField fx:id="tfEventDate" layoutX="101.0" layoutY="181.0" />  
 <TextField fx:id="tfEventTickets" layoutX="110.0" layoutY="220.0" />  
 <TextField fx:id="tfEventTicketLimit" layoutX="147.0" layoutY="256.0" />  
 <Button fx:id="buttonCreate" layoutX="44.0" layoutY="323.0" mnemonicParsing="false" text="Create">  
 <font>  
 <Font size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="buttonUpdate" layoutX="144.0" layoutY="323.0" mnemonicParsing="false" text="Update">  
 <font>  
 <Font name="Arial" size="14.0" />  
 </font>  
 </Button>  
 <Label fx:id="lbEventFormWarning" layoutX="89.0" layoutY="290.0" textFill="RED">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="531.0" layoutY="45.0" text="ID">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <TextField fx:id="tfEventID" layoutX="520.0" layoutY="76.0" prefHeight="25.0" prefWidth="48.0" />  
 <Button fx:id="buttonLoginReturn" layoutX="413.0" layoutY="317.0" mnemonicParsing="false" text="Return to login">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Button>  
 <Button fx:id="buttonQuery" layoutX="244.0" layoutY="321.0" mnemonicParsing="false" text="Query">  
 <font>  
 <Font name="Arial" size="14.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>

**>EventQuery-View.fxml**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/11.0.2"  
 xmlns:fx="http://javafx.com/fxml/1"  
 fx:controller="bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers.EventQueryController">  
 <children>  
 <Label fx:id="lbEvent" layoutX="41.0" layoutY="29.0" text="Event: ">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <ListView fx:id="lvEvent" layoutX="41.0" layoutY="169.0" prefHeight="200.0" prefWidth="387.0" />  
 <Label layoutX="41.0" layoutY="133.0" text="Related info">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 </children>  
</AnchorPane>

**>hello-view.fxml**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.geometry.Insets?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.layout.VBox?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.ListView?>  
<VBox alignment="CENTER" spacing="20.0" xmlns:fx="http://javafx.com/fxml"  
 fx:controller="bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers.HelloController">  
 <padding>  
 <Insets bottom="20.0" left="20.0" right="20.0" top="20.0"/>  
 </padding>  
  
 <Label fx:id="welcomeText"/>  
 <Button fx:id="lobbyButton" text="hello there!"/>  
 <ListView fx:id="listView"/>  
  
</VBox>

**>LoginForm-view.fxml**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="400.0" prefWidth="600.0"  
 xmlns="http://javafx.com/javafx/11.0.2"  
 xmlns:fx="http://javafx.com/fxml/1"  
 fx:controller="bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers.LoginFormController">  
 <children>  
 <Button fx:id="OrganizerLoginButton" layoutX="98.0" layoutY="246.0" mnemonicParsing="false" text="OrganizerLogin">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font></Button>  
 <Button fx:id="DistributorLoginButton" layoutX="365.0" layoutY="246.0" mnemonicParsing="false" text="DistributorLogin">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font></Button></children>  
 <Label layoutX="220.0" layoutY="170.0" text="Choose login type">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Label>  
 </AnchorPane>

**>OrganizerLogin-view.fxml**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/11.0.2" xmlns:fx="http://javafx.com/fxml/1" fx:controller="bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers.OrganizerLoginController">  
 <children>  
 <Label layoutX="137.0" layoutY="156.0" text="Username">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="137.0" layoutY="200.0" text="Password">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Label>  
 <TextField fx:id="tfUserOrg" layoutX="237.0" layoutY="154.0">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </TextField>  
 <TextField fx:id="tfPassOrg" layoutX="237.0" layoutY="202.0">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </TextField>  
 <Button fx:id="OrganizerLoginButton" layoutX="264.0" layoutY="291.0" mnemonicParsing="false" prefHeight="25.0" prefWidth="142.0" text="Login">  
 <font>  
 <Font size="18.0" />  
 </font></Button>  
 <Label fx:id="lbOrgWarning" layoutX="190.0" layoutY="255.0" textFill="RED">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 </children>  
</AnchorPane>

**>TicketForm-view.fxml**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="414.0" prefWidth="489.0" xmlns="http://javafx.com/javafx/11.0.2" xmlns:fx="http://javafx.com/fxml/1" fx:controller="bg.tu\_varna.sit.group25.ticketcenter.Application.presentation.controllers.TicketFormController">  
 <children>  
 <Label layoutX="50.0" layoutY="82.0" text="Name">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="50.0" layoutY="153.0" text="Event">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="50.0" layoutY="195.0" text="Number of tickets">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <Label fx:id="DistributorInForm" layoutX="56.0" layoutY="27.0" text="Distributor">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <Label layoutX="50.0" layoutY="119.0" text="Phonenumber">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <TextField fx:id="tfClientName" layoutX="117.0" layoutY="80.0" />  
 <TextField fx:id="tfClientPhonenumber" layoutX="173.0" layoutY="117.0" />  
 <TextField fx:id="tfEvent" layoutX="117.0" layoutY="151.0" />  
 <TextField fx:id="tfTicketCount" layoutX="206.0" layoutY="193.0" />  
 <Button fx:id="buttonSubmit" layoutX="166.0" layoutY="325.0" mnemonicParsing="false" text="Submit">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Button>  
 <Label fx:id="lbWarning" layoutX="75.0" layoutY="255.0" textFill="RED">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <TextField layoutX="152.0" layoutY="232.0" />  
 <Label layoutX="50.0" layoutY="234.0" text="Ticket price">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Label>  
 <Button layoutX="335.0" layoutY="15.0" mnemonicParsing="false" text="Query">  
 <font>  
 <Font name="Arial" size="18.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>

**Ticket\_Center/src/main/resources/hibernate.cfg.xml**

**hibernate.cfg.xml служи за създавенето на връзка между базата данни и JavaFX проекта**

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">  
<!-- Version 8 MySQL hiberante-cfg.xml example for Hibernate 5 -->  
<hibernate-configuration>  
 <session-factory>  
 <property name="connection.driver\_class">com.mysql.cj.jdbc.Driver</property>  
 <!-- property name="connection.driver\_class">com.mysql.jdbc.Driver</property -->  
 <property name="connection.url">jdbc:mysql://127.0.0.1:3306/TicketCenter</property>  
 <property name="dialect">org.hibernate.dialect.MySQL8Dialect</property>  
 <property name="connection.username">root</property>  
 <property name="connection.password">Kyuubi279</property>  
 <!--property name="dialect">org.hibernate.dialect.MySQLDialect</property-->  
 <property name="current\_session\_context\_class">thread</property>  
 <property name="show\_sql">true</property>  
 <property name="format\_sql">true</property>  
 <property name="hbm2ddl.auto">update</property>  
 <!-- mapping class="com.mcnz.jpa.examples.Player" / -->  
  
 <mapping class="bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Customer"/>  
 <mapping class="bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Distributor"/>  
 <mapping class="bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Form"/>  
 <mapping class="bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Organizer"/>  
 <mapping class="bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Show"/>  
 <mapping class="bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Show\_Status"/>  
 <mapping class="bg.tu\_varna.sit.group25.ticketcenter.Application.data.entities.Show\_Type"/>  
 </session-factory>  
</hibernate-configuration>

**Ticket\_Center/pom.xml**

**В po.xml се съдържат всички депендансита, които са необходими за проекта** <?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>bg.tu\_varna.sit.group25</groupId>  
 <artifactId>TicketCenter</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <name>TicketCenter</name>  
  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <junit.version>5.7.1</junit.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-controls</artifactId>  
 <version>16</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-fxml</artifactId>  
 <version>16</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-web</artifactId>  
 <version>16</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.controlsfx</groupId>  
 <artifactId>controlsfx</artifactId>  
 <version>11.1.0</version>  
 </dependency>  
  
 <dependency>  
 <groupId>com.dlsc.formsfx</groupId>  
 <artifactId>formsfx-core</artifactId>  
 <version>11.3.2</version>  
 <exclusions>  
 <exclusion>  
 <groupId>org.openjfx</groupId>  
 <artifactId>\*</artifactId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
 <dependency>  
 <groupId>net.synedra</groupId>  
 <artifactId>validatorfx</artifactId>  
 <version>0.1.13</version>  
 <exclusions>  
 <exclusion>  
 <groupId>org.openjfx</groupId>  
 <artifactId>\*</artifactId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
 <dependency>  
 <groupId>org.kordamp.ikonli</groupId>  
 <artifactId>ikonli-javafx</artifactId>  
 <version>12.2.0</version>  
 </dependency>  
 <dependency>  
 <groupId>org.kordamp.bootstrapfx</groupId>  
 <artifactId>bootstrapfx-core</artifactId>  
 <version>0.4.0</version>  
 </dependency>  
 <dependency>  
 <groupId>eu.hansolo</groupId>  
 <artifactId>tilesfx</artifactId>  
 <version>11.48</version>  
 <exclusions>  
 <exclusion>  
 <groupId>org.openjfx</groupId>  
 <artifactId>\*</artifactId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>log4j</groupId>  
 <artifactId>log4j</artifactId>  
 <version>1.2.17</version>  
 </dependency>  
 <dependency>  
 <groupId>org.hibernate</groupId>  
 <artifactId>hibernate-core</artifactId>  
 <version>5.6.1.Final</version>  
 </dependency>  
 <dependency>  
 <groupId>com.h2database</groupId>  
 <artifactId>h2</artifactId>  
 <version>2.0.204</version>  
 </dependency>  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>8.0.27</version>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.8.1</version>  
 <configuration>  
 <source>16</source>  
 <target>16</target>  
 </configuration>  
 </plugin>  
 <plugin>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-maven-plugin</artifactId>  
 <version>0.0.7</version>  
 <executions>  
 <execution>  
 <!-- Default configuration for running with: mvn clean javafx:run -->  
 <id>default-cli</id>  
 <configuration>  
 <mainClass>  
 bg.tu\_varna.sit.group25.ticketcenter/bg.tu\_varna.sit.group25.ticketcenter.Application.application.HelloApplication  
 </mainClass>  
 </configuration>  
 </execution>  
 </executions>  
 </plugin>  
 </plugins>  
 </build>  
</project>

База данни:

Graphical user interface

Description automatically generated