



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ ВАРНА  
ФАКУЛТЕТ ПО ИЗЧИСЛИТЕЛНА ТЕХНИКА И АВТОМАТИЗАЦИЯ

# КУРСОВ ПРОЕКТ

по

Обектно – ориентирано програмиране

II част

Айше Ердинч Шабан, факултетен номер: 19621727

Деница Живкова Николова, факултетен номер: 19621781

Специалност: Софтуерни и Интернет Технологии(СИТ), III курс, I група

Тема 14: Билетен център

# Съдържание

Съдържание.....	2
Задание.....	3
1.Анализ на проблема.....	4
1.1 Функционални изисквания.....	4
1.2 Структура на проекта и дефиниция на модулите на системата.....	5
2. Проектиране на системата.....	8
2.1 Use Case диаграма.....	8
2.2 Class диаграми.....	9
2.3 Модел на Чен.....	12
3. Реализация на системата.....	13
3.1 Реализация на базата от данни.....	13
3.2 Реализация на слоя за работа с базата данни.....	16
3.3 Реализация на бизнеслогика и графичен интерфейс.....	17
3.4 Реализация на модул за регистриране на събития в системата.....	17
4. Тестове.....	18
5.Примерно изпълнение на програмата.....	20

## Задание - 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 които е организатор.

Организаторът 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 непродадени билети (собственик, разпространител на билети)

## 1. Анализ на проекта

### 1.1. Функционални изисквания

Необходимо е да се създаде система за добавяне на събития и продажба на билети. При правилна реализация на заданието, системата позволява множествен достъп, като това се постига чрез обхождане на съществуващите в базата от данни таблици, които отговарят на различните профили. Възможните видове потребители са администратор, организатор на събитие и разпространител на билети.

Администраторът е потребителят, който стои в основата на създадената система. Чрез неговите достъпи в системата, става възможно добавянето на нови организатори и разпространители и предоставянето им на достъп до функциите на системата.

При успешно създаване на профил с достъп до системата като организатор, потребителят получава право за добавяне на нови събития и да определя потребителите, с право на достъп като разпространител на билети, които ще имат предоставена възможност за продажба на билети. Също така организаторът има право да определя видовете места и съответно цена на билета по вид. Ако потребителят реши, че е създал представа за работата на конкретен разпространител, той може да добави рейтинг за него като този рейтинг може да е от 0 до 100 на базата на процент. Преди да се добави рейтинг, организаторът може да провери скалата, според която се определя оценката, и да определи правилната числова стойност, която отговаря на неговата представа.

Последният вид достъп, е този, който потребител получава при създаден профил като разпространител на билети. Основното действие, което може да се извършва, е продаване на билети. Разпространителят създава формуляри, съдържащи информация за клиента, събитието и вида на билета. Ако билетите за вида не са разпродадени или клиентът не е купил максимален брой билети за човек, определени за събитието, системата създава формуляра и намалява броя на непроданите билети с 1.

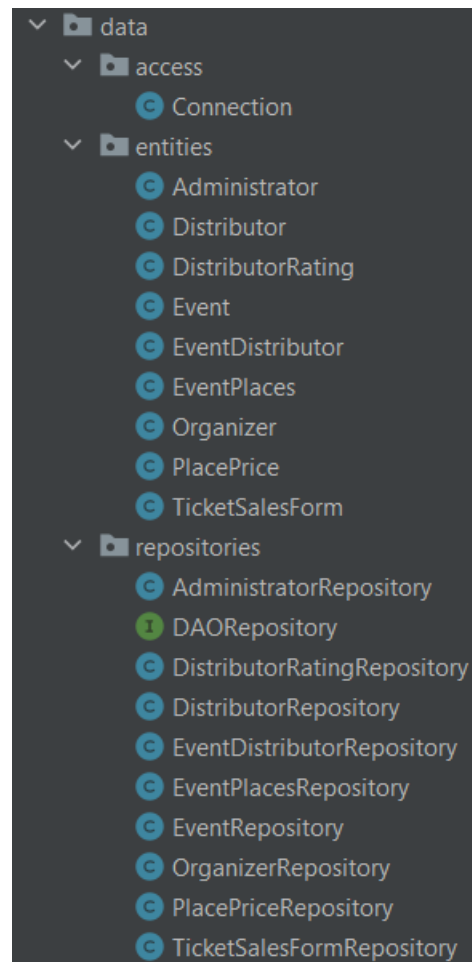
## 1.2. Структура на проекта и дефиниция на модулите на системата

Проектът е разделен на няколко модула. Целта на разделянето е създаване на структуриран проект, по-лесен за четене и разбиране.

### 1.2.1 Даннов слой

Целта от създаването на даннов слой е групиране и отделяне на класовете, чрез които програмата създава връзка с базата от данни. Сложът вътрешно е разделен на три пакета – **Access, Entities, Repositories**.

Пакетът **Access** съдържа един единствен клас **Connection**. В него се създава обект от **SessionFactory**. **Hibernate SessionFactory** е интерфейс, чрез който получаваме сесии и извършваме операции с база данни. **SessionFactory** може да бъде създаден чрез предоставяне на конфигурационен обект, който съдържа всички подробности за свойствата, свързани с базата от данни, извлечени от **hibernate.cfg.xml** файл. Може да се създава една реализация на **SessionFactory** на база данни във всяко приложение. Сесията се използва за получаване на физическа връзка с база данни. Обектът **Session** е проектиран да се инстанцира всеки път, когато е



необходимо взаимодействие с базата данни. Основната функция на сесията е да предлага, създава, чете и изтрива операции за екземпляри на класове обекти.

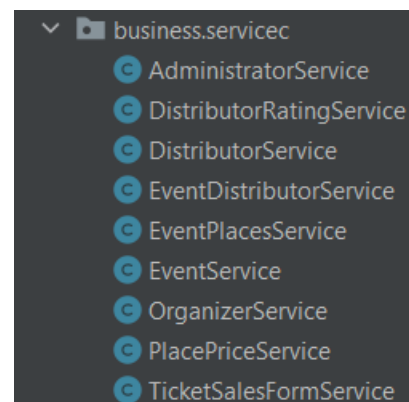
Пакетът **Entities** съдържа 9 класа като всеки един от тях отговаря на таблица от БД. Всеки един клас имплементира интерфейса `Serializable`. Сериализацията е преобразуване на състоянието на обект в поток от байтове. Казано по различен начин, сериализацията е преобразуване на Java обект в статичен поток от байтове, които след това можем да запишем в база данни. Десериализацията е обратният процес, при който потокът от байтове се използва за пресъздаване на действителния Java обект в паметта. В класовете създаване на таблицата се постига чрез анотациите `@Entity` и `@Table` следвано от името на таблицата от БД, а за колоните от таблиците се използва анотацията `@Column` (ако колоната е и едновременно идентификационен номер се добавя и `@ID`). За създаване на различните видове връзки се използват `@OneToMany`, `@ManyToOne`, `@OneToOne` или `@ManyToMany`.

Пакетът **Repositories** съдържа 9 класа и 1 интерфейс. Интерфейсът има 4 метода като те са съответно за създаване на обект, за промяна на обект, за изтриване на обект и за извличане на всички обекти от таблицата в БД. Класовете имплементират тези методи като всеки един клас отговаря на 1 таблица от БД.

### 1.2.2 Бизнес слой

Бизнес слойът съдържа един пакет **Services**, а в този пакет отново се съдържат 9 класа.

В Service класовете са създадени функциите, които са необходими за изпълнението на различните задачи в програмата. Основната идея е информацията, която постъпва от контролерите, с помощта на моделите (ще бъдат разгледани по-късно в документацията), да достигне до Service класа и там да бъде преобразувана според изискванията на заданието и чрез извикване на методи от Repository класовете за отделните таблици тази информация да постъпи в базата от данни. Също така чрез методите на Service класовете може да се изведе в



### 1.2.3 Презентационен слой

**Models** – и в този пакет отново са създадени 9 класа, отговарящи на таблиците в БД. Models класовете са създадени, за да може да се създаде прототип на елемент от таблица на БД. Благодарение на тези класове, извлечените от контролерите информация, може да бъде запазена като обект и предадена към Service класовете за преобразуване и оттам към Repository класовете за съхраняване в БД. Models класовете съдържат частни полета, отговарящи на колоните от таблиците, Set & Get методи за тези полета, подразбиращ се и експлицитен конструктор и ToString() метод.

**Views** пакетът съдържа **fxml** файлове. FXML файловете са базирани на XML езика, който осигурява структурата за изграждане на потребителски интерфейс, отделно от логиката на приложението. Чрез тези файлове се създават различните сцени, необходими за

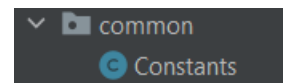
изграждане на приложението. В файловете се описват разположението на контролерите в сцената, техните размери, цветове и т.н. За да бъде възможна връзката между fxml-файл и Controller клас е необходимо към файла да се добави името на класа за връзка чрез `fx:controller="път към класа"`, а връзка между контролерите се създава като към всеки един от тях се добави `id`, предхождащо се от `fx:`. Имената на различните контролери във файла трябва да отговарят на имената на `@FXML` контролерите в класа.



**Controllers** пакетът съдържа класове, които са свързани с fxml файловете, и правят възможна комуникацията между потребителя и системата. Те създават различните сцени и реализират прехода между тях. Също така извличат информацията, въведена от потребителя, и я предават като създават обекти от Models класовете. Ключовите моменти при създаването на тези класове е правилното предаване на пътя до fxml файла, който отговаря на съответната сцена и съответствието на имената на контролерите между файла и класа (използва се @FXML преди декларирането на контролер в класа).

## 1.2.4 Utilities слой

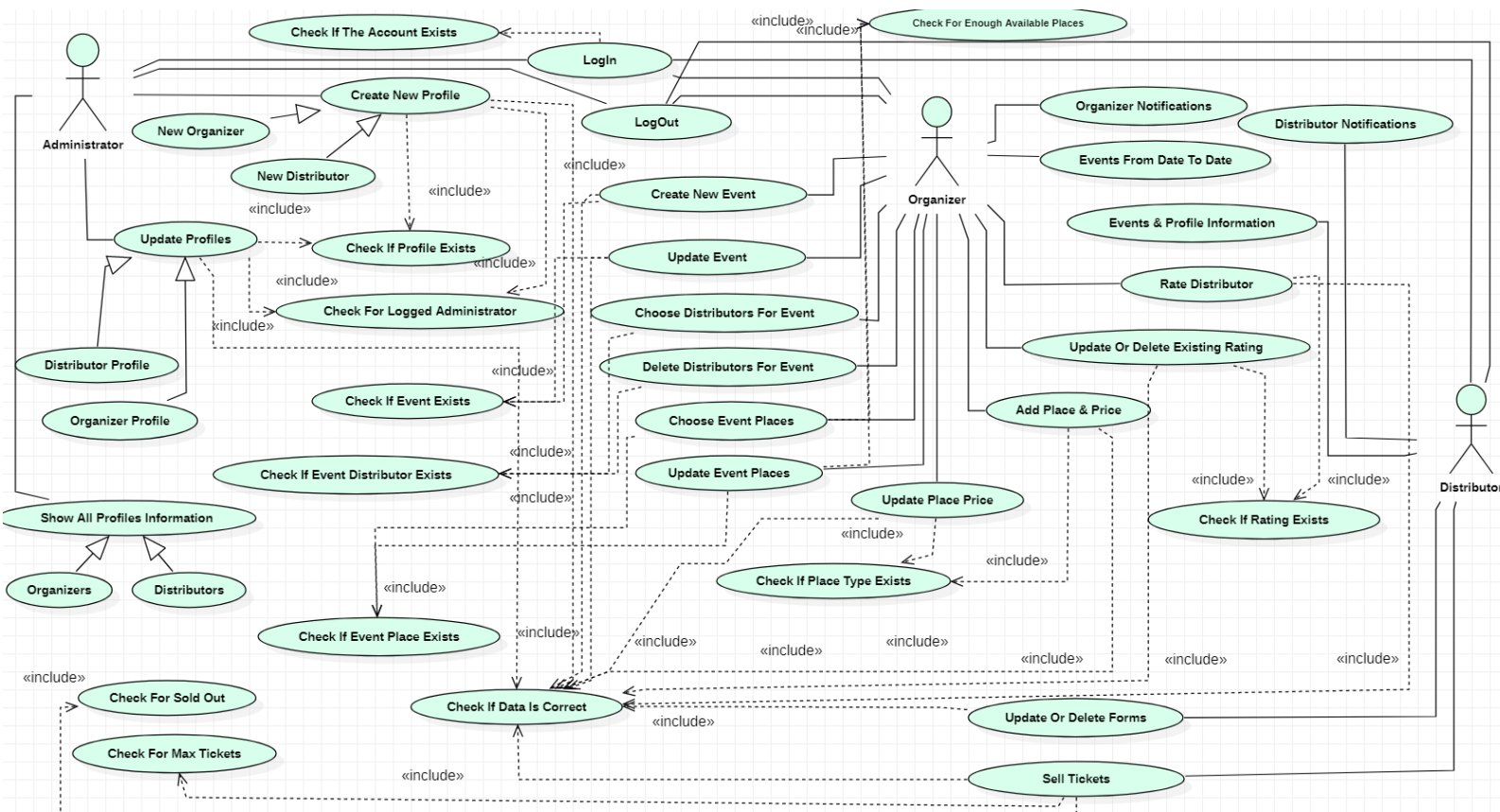
В този слой се съдържа един единствен клас **Constants**. В него се съхранява информация, под формата на константи, за пътя до всеки един fxml файл от Views пакета. Тези константи се използват в Controller класовете.



## 2. Проектиране на системата

### 2.1. UML диаграми

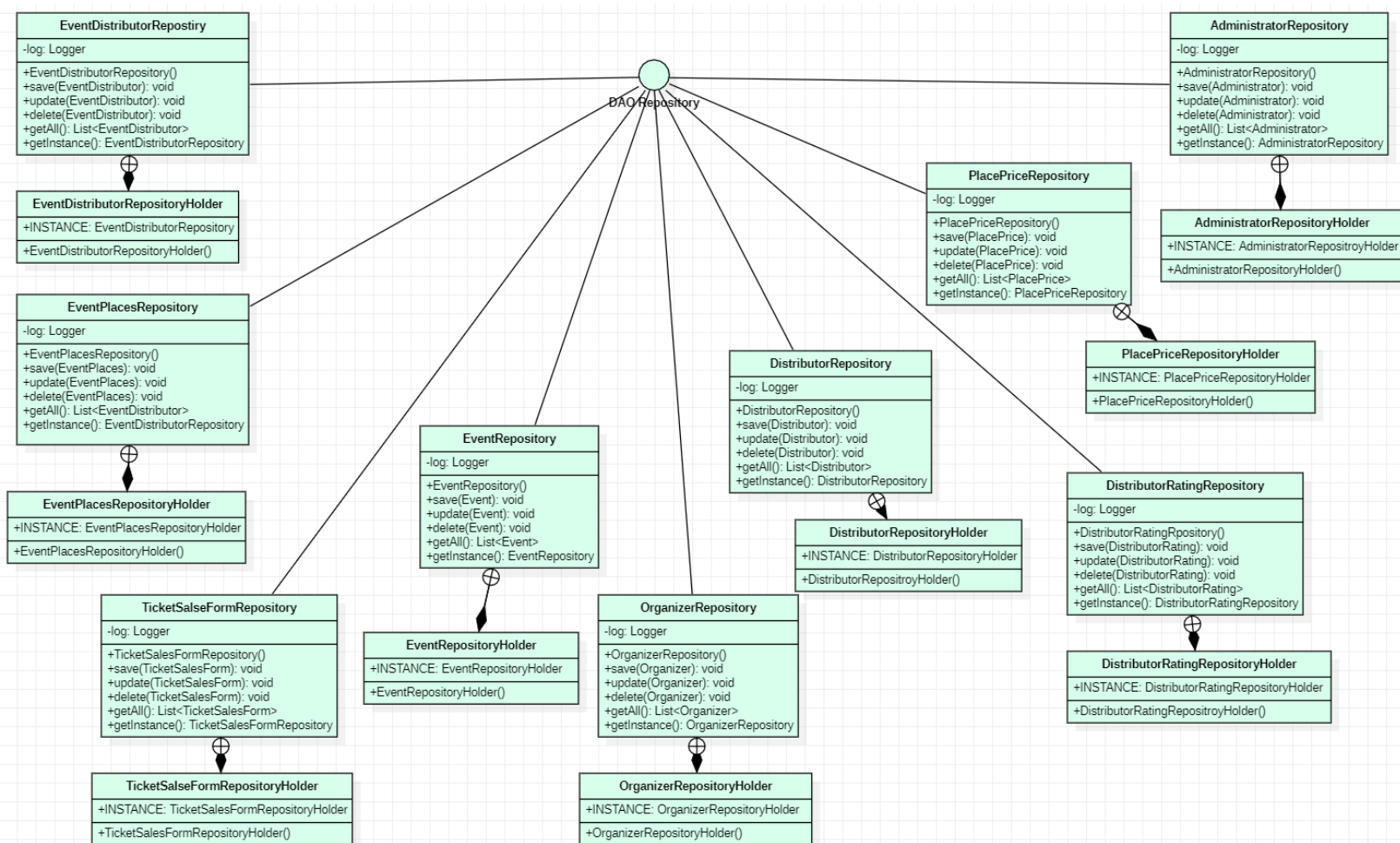
#### 2.1.1. Use Case



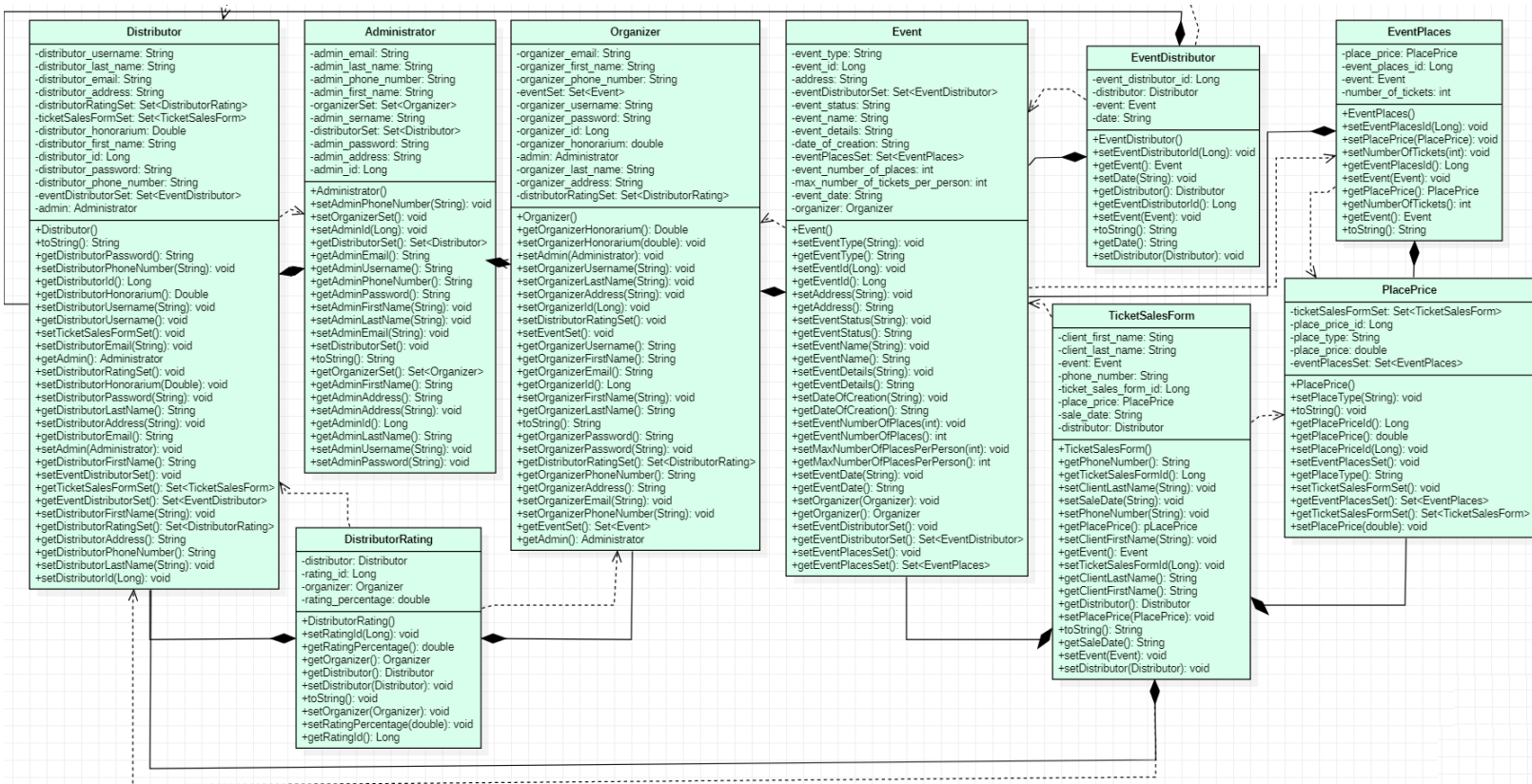


## 2.1.2. Class диаграми

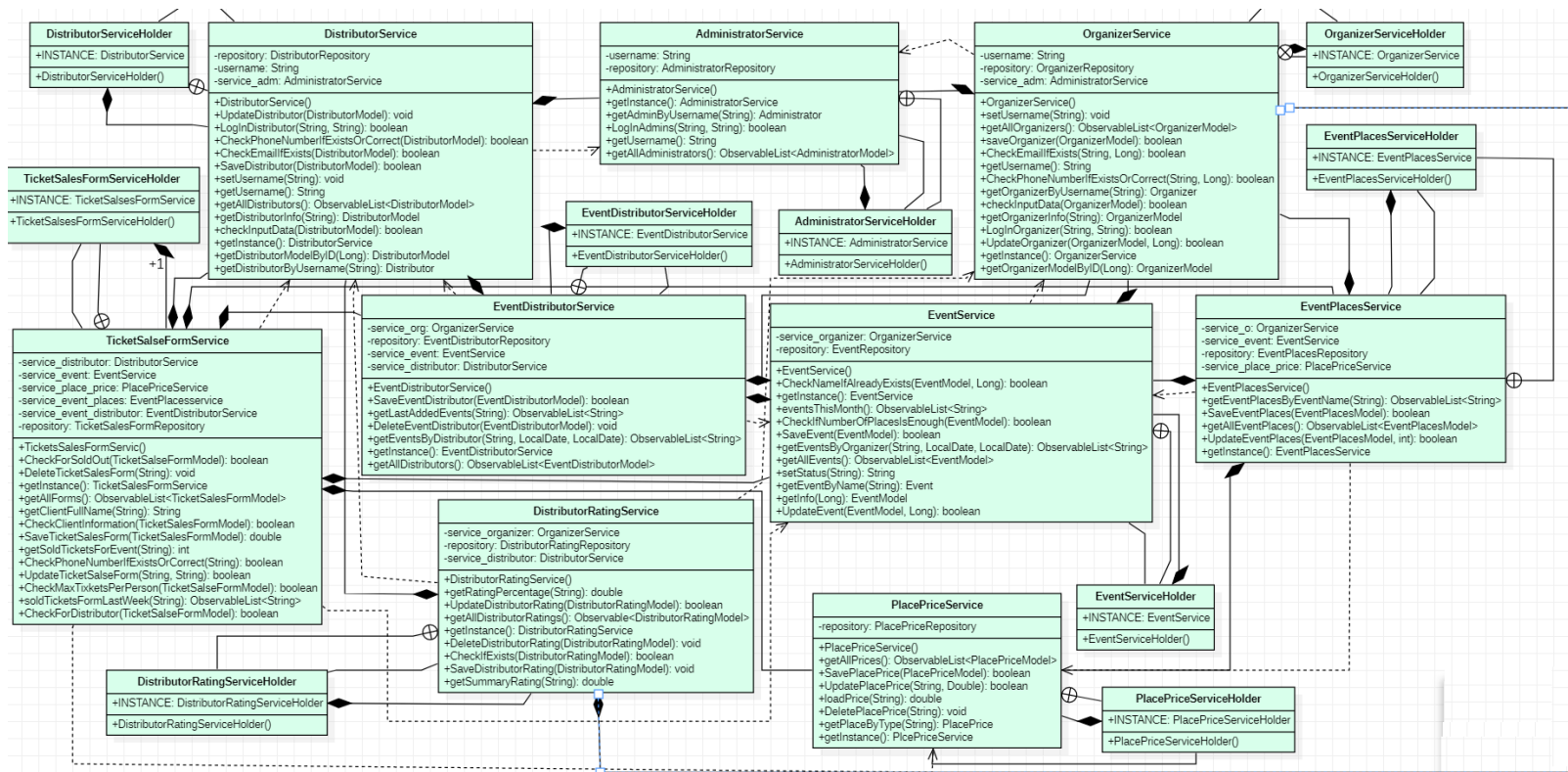
### 2.1.2.1. Class Diagram – Repositories



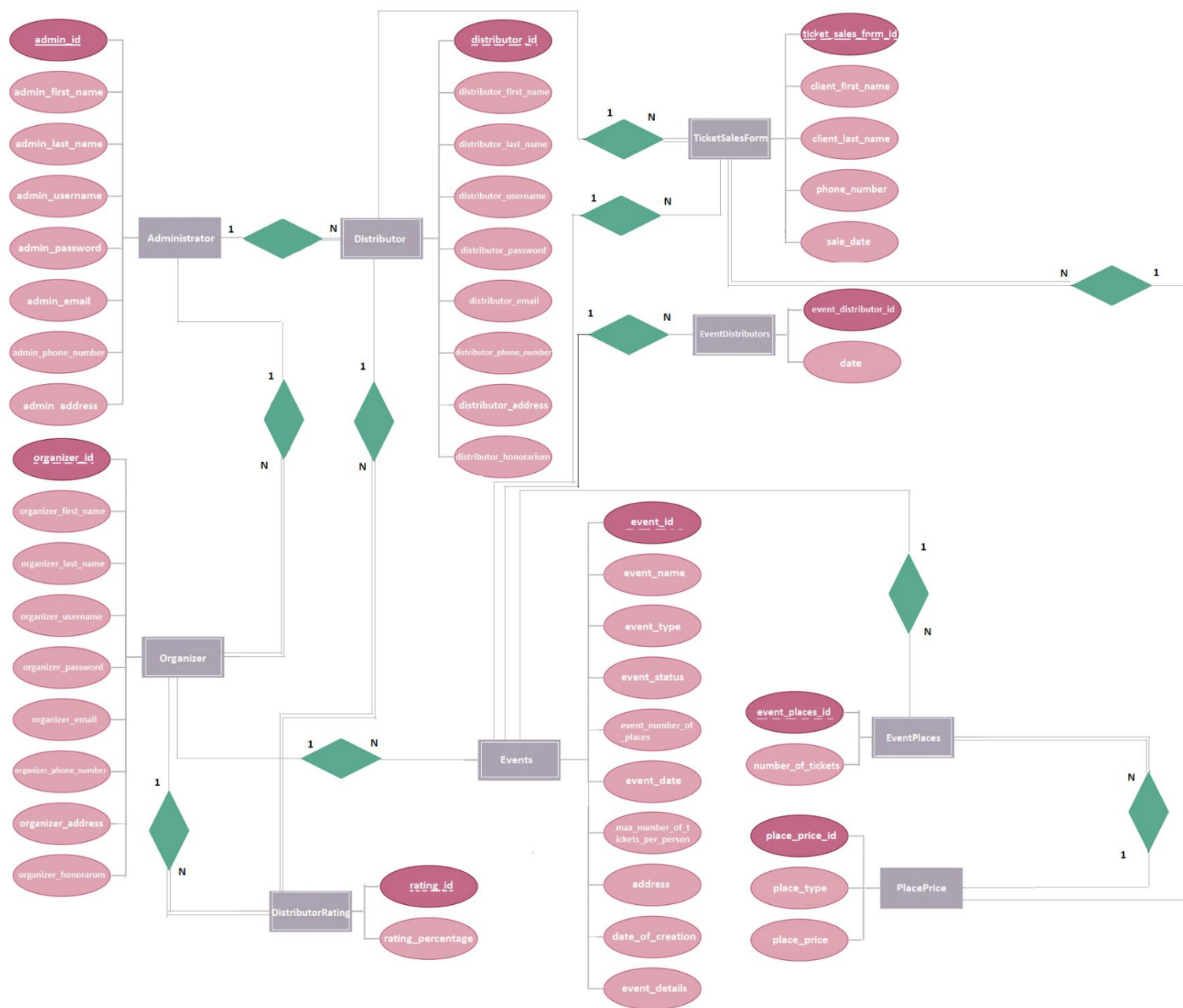
## 2.1.2.2. Class Diagram – Entities



### 2.1.2.3. Class Diagram – Services

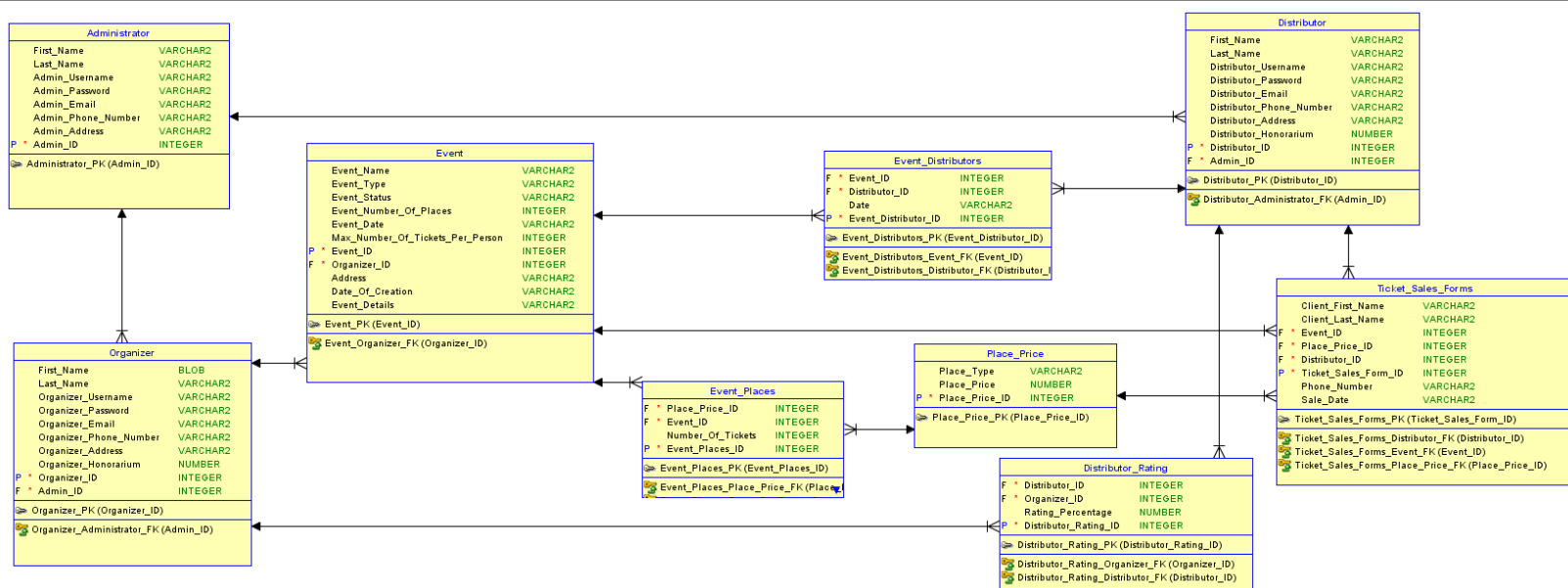


## 2.2. Концептуален модел на базата от данни – ER диаграма (Модел на Чен)



### 3. Реализация на системата

#### 3.1. Реализация на базата от данни – PostgreSQL – релационна схема и описание на таблиците



##### 3.1.1. Таблицы – в базата от данни са създадени 9 таблици

- 1) **Administrators** – таблицата съдържа колони, които съхраняват информация за администратор (име, фамилия, потребителско име, парола, имейл, телефонен номер, адрес и идентификационен номер).
- 2) **Organizers** – тази таблица съхранява същата информация както за администратора, само че за организатор. Има още една добавена колона – за хонорар на организатора.
- 3) **Distributors** – идентичен е на таблицата за организаторите.
- 4) **Events** – съхранява информация за създадените събития (име, вид, статус, брой места, дата на събитието, максимален брой билети за един клиент, адрес, дата на създаване на събитието, детайли за събитието, идентификационен номер на събитието и номер на организатора, който е създал събитието).
- 5) **Event Distributors** – съхранява информация за дистрибутори, отговорни за продажбата на билетите на дадено събитие. Колоните в таблицата отговарят на идентификационен

номер на дистрибутор и номер на събитие. ID на таблицата е зададено да се добавя автоматично.

- 6) **Place Prices** – съхранява информация за видовете места и тяхната цена. Колоните в таблицата отговарят на името на вида място и съответната цената. ID на таблицата е зададено да се добавя автоматично.
- 7) **Event Places** - съхранява информация за видовете места в дадено събитие. Колоните в таблицата отговарят на идентификационен номер на място, номер на събитие и брой места за вид. ID на таблицата е зададено да се добавя автоматично.
- 8) **Distributor Ratings** - съхранява информация за рейтингите на дистрибутор. Колоните в таблицата отговарят на идентификационен номер на дистрибутор, идентификационен номер на организатор, който оценява дистрибутора и стойността на рейтинга (0-100). ID на таблицата е зададено да се добавя автоматично.
- 9) **Ticket Sales Forms** – съхранява информация за продажба на билети. Колоните съхраняват информация за името и фамилията на клиента, както и телефонният му номер, дата на продажбата, номер на събитието, номер на вида на мястото и номер на дистрибутора. ID на таблицата е зададено да се добавя автоматично.

### 3.1.2. Връзки

- 1) **Администратор – Организатор** – връзката е One-To-Many, защото един администратор може да има много организатори, но организаторът може да има само един администратор.
- 2) **Администратор – Дистрибутор** - връзката е One-To-Many, защото един администратор може да има много дистрибутори, но дистрибуторът може да има само един администратор.

- 3) **Организатор – Събитие** - връзката е One-To-Many, защото един организатор може да има много събития, но събитието може да има само един организатор.
- 4) **Организатор – Рейтинг на дистрибутор** - връзката е One-To-Many, защото един организатор може да рейтингова много дистрибутори, но един рейтинг може да има само един организатор.
- 5) **Дистрибутор – Рейтинг на дистрибутор** – връзката е One-To-Many, защото един дистрибутор може да има много рейтинги, един рейтинг може да има само един дистрибутор.
- 6) **Дистрибутор – Формуляр за продажба на билети** – връзката е One-To-Many, защото един дистрибутор може да има много формуляри, един формуляр може да има само един дистрибутор.
- 7) **Дистрибутор – Дистрибутори на събитие** – връзката е One-To-Many, защото един дистрибутор може да е дистрибутор на много събития.
- 8) **Цена на билет за вид – Видове места за събитие** – връзката е One-To-Many, защото даден вид може да има много събития.
- 9) **Цена на билет за вид – Формуляр за продажба на билети** – връзката е One-To-Many, защото даден вид може да има много формуляри за продажба, един формуляр може да има само един вид на място.
- 10) **Събитие – Дистрибутори на събитие** – връзката е One-To-Many, защото едно събитие може да има много дистрибутори.
- 11) **Събитие – Видове места за събитието** – връзката е One-To-Many, защото събитие може да има много видове места.



12) **Събитие – Формуляр за продажба на билети** – връзката е One-To-Many, защото едно събитие може да има много формуляри за продажба, докато един формуляр може да има само един организатор.

### 3.1.3. Реализация – PostgreSQL

Таблиците и връзките са създадени чрез PostgreSQL.

**PostgreSQL**- мощна обектно-реляционна база данни с отворен код, която използва и разширява SQL езика, комбиниран с много функции, които безопасно съхраняват и мащабират най-сложните работни натоварвания с данни.

## 3.2. Реализация на слоя за работа с базата от данни –Hibernate

**Hibernate** е една от най-популярните Java ORM frameworks, които се използват днес. ORM е процесът на преобразуване на Java обекти в таблици на база данни.

Основната функция на Hibernate е преобразуване от Java класове към таблици на база данни и преобразуване от типове данни на Java към типове данни на SQL. Hibernate също така предоставя средства за запитване и извличане на данни.

Конфигурационен файл :

```
<?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"
<hibernate-configuration>
  <session-factory>

    <property name="hibernate.dialect"> org.hibernate.dialect.PostgreSQL82Dialect</property>
    <property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
    <property name="hibernate.connection.url">jdbc:postgresql://localhost/ticketCenter</property>
    <property name="hibernate.connection.username">postgres</property>
    <property name="hibernate.connection.password">napravih@registraci13A</property>
    <property name="hibernate.show_sql">true</property>
    <property name="hibernate.hbm2ddl.auto">update</property>

    <mapping class="bg.tu_varna.sit.ticket_center.data.entities.Administrator" />
    <mapping class="bg.tu_varna.sit.ticket_center.data.entities.Distributor" />
    <mapping class="bg.tu_varna.sit.ticket_center.data.entities.Organizer" />
    <mapping class="bg.tu_varna.sit.ticket_center.data.entities.Event" />
    <mapping class="bg.tu_varna.sit.ticket_center.data.entities.EventDistributor" />
    <mapping class="bg.tu_varna.sit.ticket_center.data.entities.EventPlaces" />
    <mapping class="bg.tu_varna.sit.ticket_center.data.entities.PlacePrice" />
    <mapping class="bg.tu_varna.sit.ticket_center.data.entities.TicketSalesForm" />
    <mapping class="bg.tu_varna.sit.ticket_center.data.entities.DistributorRating" />

  </session-factory>
</hibernate-configuration>
```





Properties:

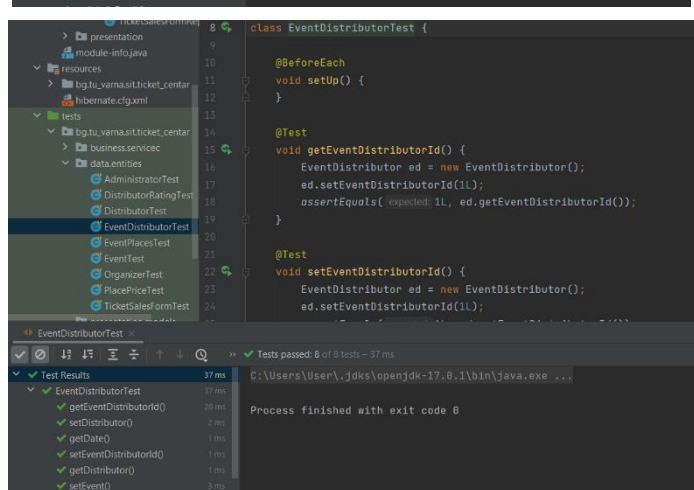
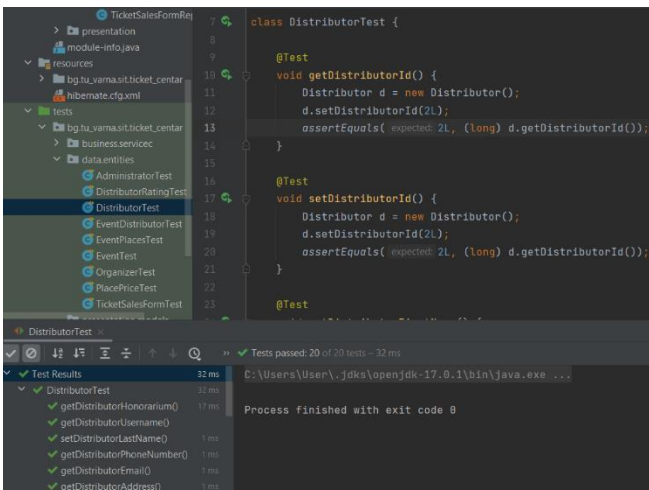
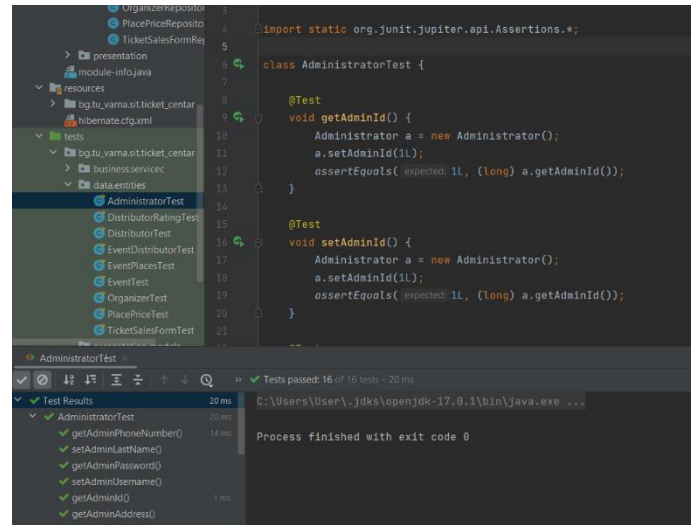
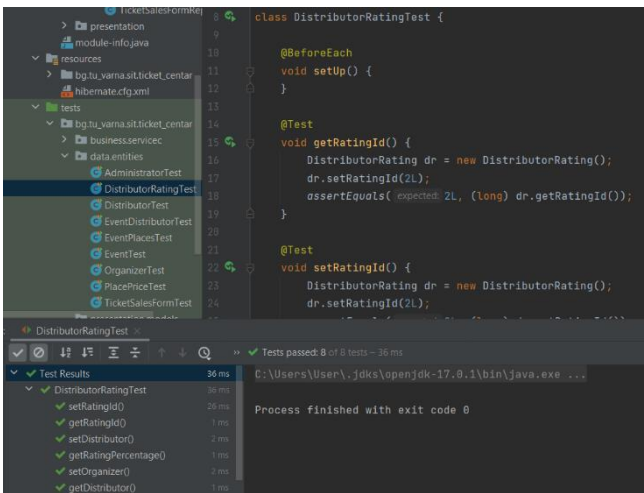
```
log4j.rootLogger=ERROR, console, file

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
```

#### 4. Тестови резултати

**JUnit тестове** - Unit Testing се използва за проверка на малка част от кода чрез създаване на път, функция или метод. За Java разработчиците е полезно да пишат и изпълняват повторяеми тестове, за да проверят достоверността на кода, който са написали.



```
class EventPlacesTest {
    @BeforeEach
    void setUp() {
    }

    @Test
    void getEventPlacesId() {
        EventPlaces ep = new EventPlaces();
        ep.setEventPlacesId(1L);
        assertEquals( expected: 1L, ep.getEventPlacesId());
    }

    @Test
    void setEventPlacesId() {
        EventPlaces ep = new EventPlaces();
        ep.setEventPlacesId(1L);
    }
}
```

Test Results: 39 ms, 8 tests passed.

```
class EventTest {
    @BeforeEach
    void setUp() {
    }

    @Test
    void getEventId() {
        Event e = new Event();
        e.setEventId(1L);
        assertEquals( expected: 1L, e.getEventId());
    }

    @Test
    void setEventId() {
        Event e = new Event();
        e.setEventId(1L);
    }
}
```

Test Results: 52 ms, 22 tests passed.

```
class OrganizerTest {
    @Test
    void getOrganizerId() {
        Organizer o = new Organizer();
        o.setOrganizerId(1L);
        assertEquals( expected: 1L, o.getOrganizerId());
    }

    @Test
    void setOrganizerId() {
        Organizer o = new Organizer();
        o.setOrganizerId(1L);
        assertEquals( expected: 1L, o.getOrganizerId());
    }
}
```

Test Results: 29 ms, 20 tests passed.

```
class PlacePriceTest {
    @BeforeEach
    void setUp() {
    }

    @Test
    void getPlacePriceId() {
        PlacePrice pp = new PlacePrice();
        pp.setPlacePriceId(4L);
        assertEquals( expected: 4L, pp.getPlacePriceId());
    }

    @Test
    void setPlacePriceId() {
        PlacePrice pp = new PlacePrice();
        pp.setPlacePriceId(4L);
    }
}
```

Test Results: 23 ms, 6 tests passed.

```
class TicketSalesFormTest {
    @BeforeEach
    void setUp() {
    }

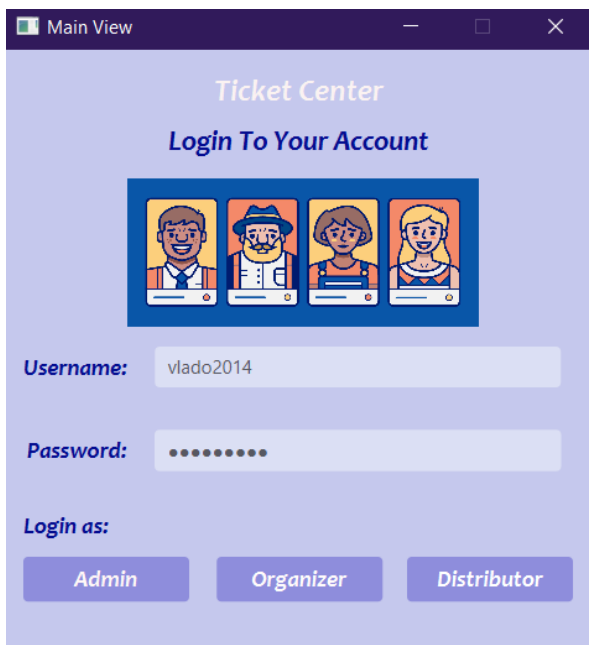
    @Test
    void getTicketSalesFormId() {
        TicketSalesForm t = new TicketSalesForm();
        t.setTicketSalesFormId(7L);
        assertEquals( expected: 7L, t.getTicketSalesFormId());
    }

    @Test
    void setTicketSalesFormId() {
        TicketSalesForm t = new TicketSalesForm();
        t.setTicketSalesFormId(7L);
    }
}
```

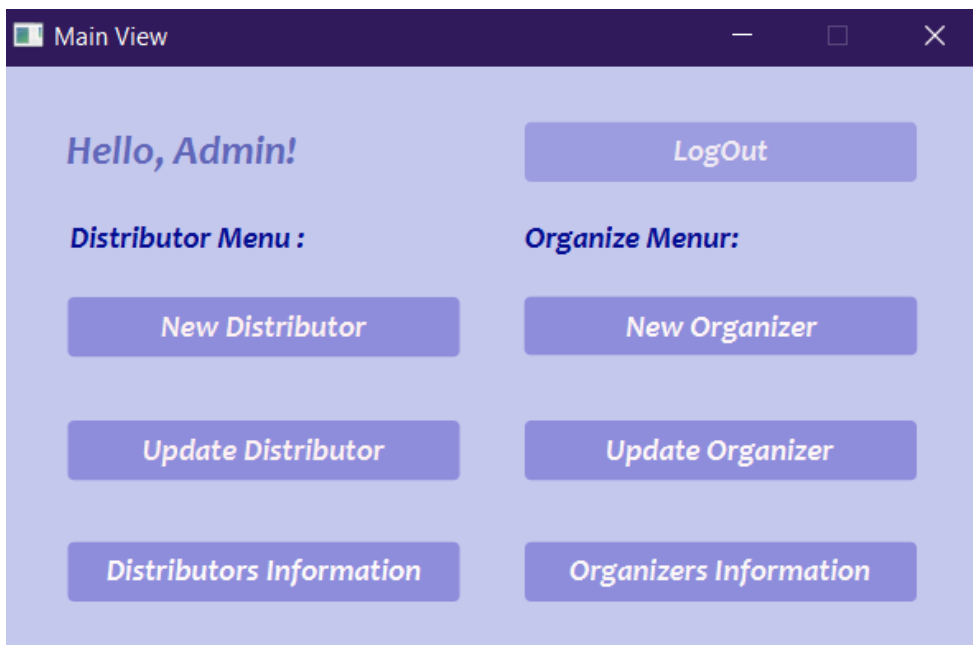
Test Results: 47 ms, 16 tests passed.

## 5. Примерно изпълнение на програмата

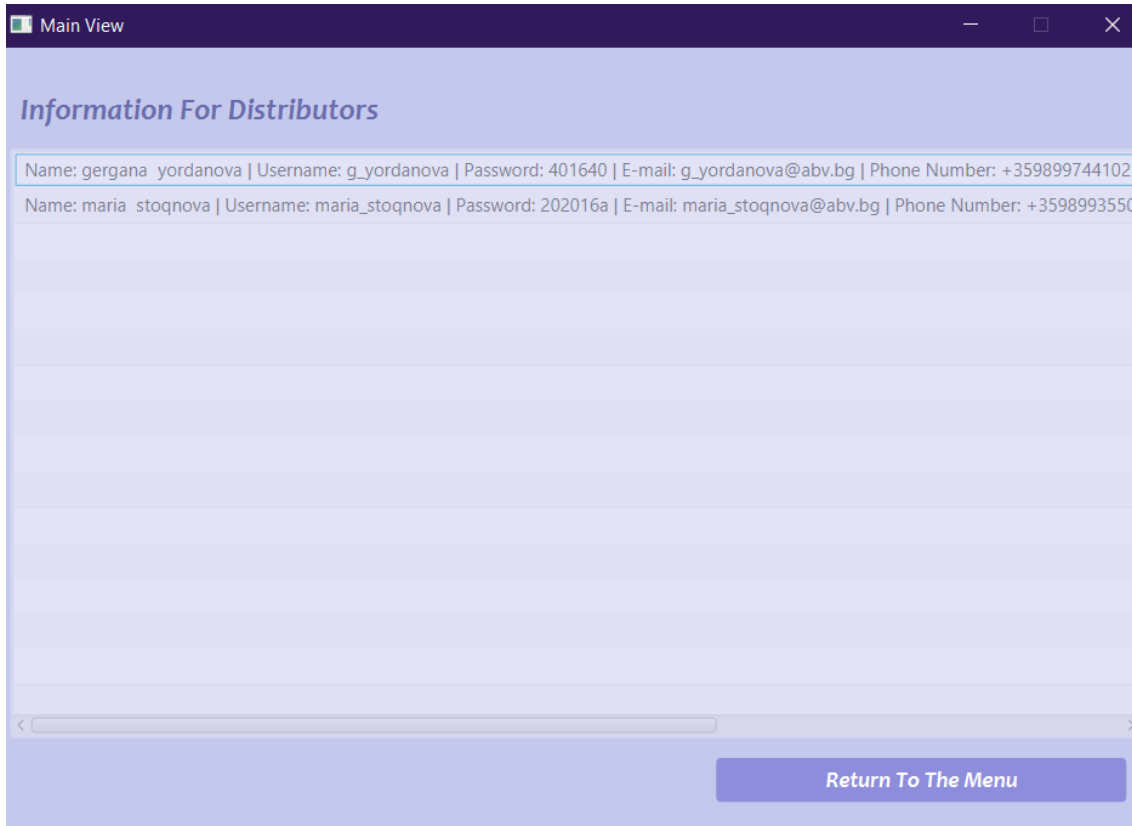
### 5.1. Профил на администратор



The screenshot shows a web application window titled "Main View". The page has a light purple background and is titled "Ticket Center" with the subtitle "Login To Your Account". Below the title, there are four cartoon avatars of people. Underneath the avatars, there are two input fields: "Username:" with the text "vlado2014" and "Password:" with a masked password ".....". Below these fields, there is a "Login as:" label and three buttons: "Admin", "Organizer", and "Distributor".



The screenshot shows the "Main View" dashboard for an administrator. It features a "Hello, Admin!" greeting and a "LogOut" button. Below the greeting, there are two columns of buttons. The left column is titled "Distributor Menu :" and contains buttons for "New Distributor", "Update Distributor", and "Distributors Information". The right column is titled "Organize Menur:" and contains buttons for "New Organizer", "Update Organizer", and "Organizers Information".



The screenshot shows the "Main View" window displaying the "Information For Distributors" page. The page has a light purple background and a title "Information For Distributors". Below the title, there is a table with two rows of distributor information. The first row shows "Name: gergana yordanova | Username: g\_yordanova | Password: 401640 | E-mail: g\_yordanova@abv.bg | Phone Number: +359899744102 |". The second row shows "Name: maria stoqnova | Username: maria\_stoqnova | Password: 202016a | E-mail: maria\_stoqnova@abv.bg | Phone Number: +3598993550 |". Below the table, there is a "Return To The Menu" button.

Name: gergana yordanova   Username: g_yordanova   Password: 401640   E-mail: g_yordanova@abv.bg   Phone Number: +359899744102
Name: maria stoqnova   Username: maria_stoqnova   Password: 202016a   E-mail: maria_stoqnova@abv.bg   Phone Number: +3598993550

Main View

### Information For Organizers

Name: georgi bogdanov	Username: g_bogdanov	Password: 1111a	E-mail: g_bogdanov@abv.bg	Phone Number: +359899300514	Address: prosve
Name: ivan ivanov	Username: ivan1436	Password: 1234ert	E-mail: ivan1600@abv.bg	Phone Number: +359899655024	Address: prosve
Name: ivaylo bogdanov	Username: ivo_bogdanov	Password: 1234	E-mail: ivo_bogdanov	Phone Number: +359899366874	Address: add
Name: ivan vladimirov	Username: i_v	Password: 1222	E-mail: i_v@abv.bg	Phone Number: +359899366052	Address: addres   Honorariu

Return To The Menu

Main View

### New Distributor

Added Successfully!

<b>First Name:</b> <input type="text" value="Mariqn"/>	<b>Phone Number:</b> <input type="text" value="+359"/> <input type="text" value="899744745"/>
<b>Last Name:</b> <input type="text" value="Iliev"/>	<b>Address:</b> <input type="text" value="addresss"/>
<b>Username:</b> <input type="text" value="m_iliev"/>	<b>Distributor ID:</b> <input type="text" value="5445"/> <b>Honorarium:</b> <input type="text" value="150"/>
<b>Password:</b> <input type="text" value="481248aB"/>	<input type="button" value="Create Distributor"/>
<b>E-mail:</b> <input type="text" value="m_iliev@abv.bg"/>	<input type="button" value="Return To The Menu"/>

Main View

Update Distributor

Enter ID Of The Distributor You Want To Select:

5445

First Name:

mariqn

Last Name:

iliev

Load Information

Username:

m\_iliev

Password:

481248aB

Information To Update:

Email:

m\_iliev@abv.bg

Phone Number:

+359

899999020

Update

Honorarium:

140.0

Address:

addresssss

Return To The Menu

Main View

New Organizer

Added Successfully!

First Name:

Bogdan

Address:

addrreess

Last Name:

Vasilev

Phone Number:

+359

899744587

Username:

b\_vasilev

Organizer ID:

7711

Honorarium:

180

Password:

20005544a

Create Organizer

E-mail:

b\_vasilev@abv.bg

Return To The Menu

Main View

## Update Organizer

Enter ID Of The Organizer You Want To Select:

7711

**First Name:** bogdan **Last Name:** vasiliev **Load Information**

**Username:** b\_vasilev **Password:** 20005544a **UpdatedSuccessfully!**

**Information To Update:**

**Honorarium:** 160.0 **Address:** addreess **Update**

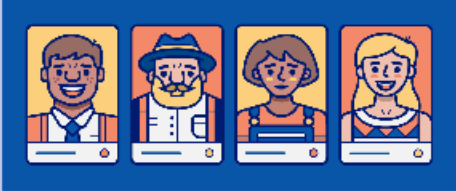
**Email:** b\_vasilev@abv.bg **Phone Number:** +359 899744587 **Return To The Menu**

## 5.2. Профил на организатор

Main View

## Ticket Center

### Login To Your Account



**Username:** ivan1436

**Password:** ••••••••

**Login as:**

**Admin** **Organizer** **Distributor**

Main View

Hello, Organizer!

New Event

Add Event Distributors

Update Event

Delete Event Distributors

Add Event Places

Add Place Prices

Update Event Places

Update Place Prices

Rate Distributor

Events

Update Or Delete Distributor Rating

Notifications

LogOut

Main View

New Event

Added Successfully!

Name:

new event

Type:

type

Details:

details

Event ID:

6622

Date:

1.03.2022 r.

Number Of Places:

500

Max Number Of Tickets Per Person:

5

Address:

address

Create Event

Return To The Menu



Main View

Update Event

Event Number: 6622
Load Information

Information To Update:

Event Date: 1.03.2022 r.

Event Name: new event
Event Type: type
Event Status: scheduled

Number Of Places: 300
Event Details: details

Max Number Of Tickets: 2
Address: address

Successfully Updated!
Update
Return

Main View

Event Distributors

Saved Successfully!

Distributor Username : maria\_stoqnova
Event Name : new event

maria\_stoqnova | event1  
maria\_stoqnova | event2  
g\_yordanova | event1  
g\_yordanova | event2  
maria\_stoqnova | event3  
maria\_stoqnova | event4  
maria\_stoqnova | new event

Save
Show

Return To The Menu

Main View

Delete Event Distributors

Event Name new event
Distributor username g\_yordanova

Delete
Return

Main View

Event Places

Event Name:

new event

Saved Successfully!

Place Type:

place\_type1

Save

Number Of Tickets:

200

Return To The Menu

Main View

Successfully Added!

Update Event Places

Event Name:

new event

Place Type:

place\_type1

Number Of Places:

150

Type: place\_type2 Number Of Places:100 Price: 20.0

Type: place\_type1 Number Of Places:150 Price: 28.5

Load Event Places

Update

Return

Main View

Rate Distributor

Distributor Username :

maria\_stoqnova

Rating Percentage (0-100%):

75

Percentage (%) - Interpretation:

0 - 25	Very Low
26 - 50	Low
51 - 75	Medium
76 - 90	High
91 - 100	Very High

Rate

Return To The Menu

Main View

Update / Delete Distributor Rating

Distributor Username:

maria\_stoqnova

Load Percentage

Return

Rating Percentage:

80

Update

Delete

Main View

**Events**

**From Date :** 1.02.2022 r.
**To Date :** 2.03.2022 r.

Event : event1 2022-02-19 Status: scheduled Address: address1 Number Of Tickets : 250 Sold Tickets: 9  
Event : event3 2022-02-26 Status: scheduled Address: address3 Number Of Tickets : 200 Sold Tickets: 0  
Event : event4 2022-02-27 Status: scheduled Address: address Number Of Tickets : 350 Sold Tickets: 0  
Event : new event 2022-03-01 Status: scheduled Address: address Number Of Tickets : 300 Sold Tickets: 0

Show
Return To The Menu

Main View

Today is : 2022-02-14

**Notifications**

**Sold Tickets:**

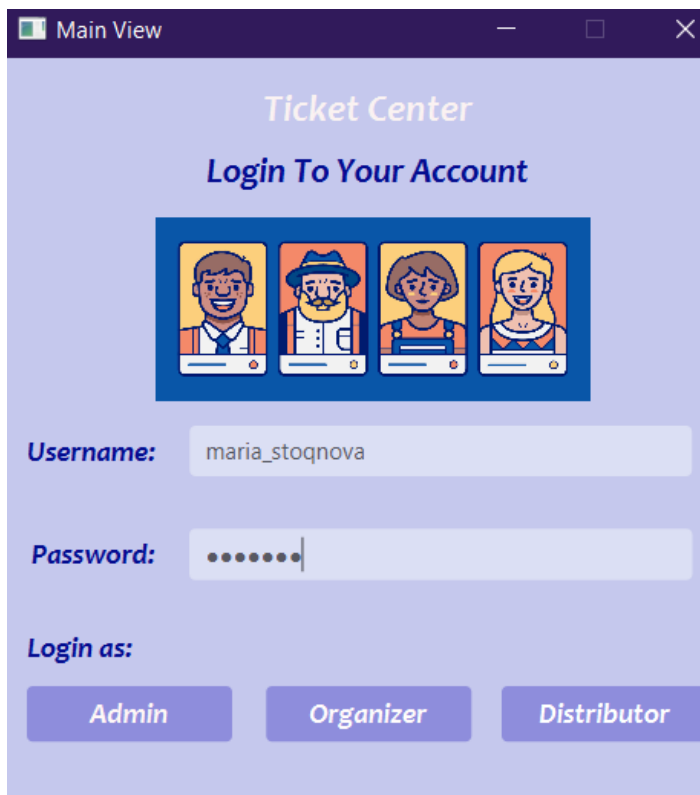
Date : 2022-02-11 Event: event1 Client Name: georgi marinov Phone Number: +359899677458  
Date : 2022-02-11 Event: event1 Client Name: mariqn vladimirov Phone Number: +359899366587  
Date : 2022-02-11 Event: event1 Client Name: mariqn vladimirov Phone Number: +359899366587  
Date : 2022-02-11 Event: event1 Client Name: mariqn vladimirov Phone Number: +359899366587  
Date : 2022-02-11 Event: event1 Client Name: mariqn vladimirov Phone Number: +359899366587  
Date : 2022-02-11 Event: event1 Client Name: mariqn vladimirov Phone Number: +359899366587  
Date : 2022-02-11 Event: event1 Client Name: mariqna marinova Phone Number: +359899366500

**Events This Month:**

Event Name : event1 Date : 2022-02-19 Number Of Places : 250 Sold: 9  
Event Name : event3 Date : 2022-02-26 Number Of Places : 200 Sold: 0  
Event Name : event4 Date : 2022-02-27 Number Of Places : 350 Sold: 0

Return

### 5.3. Профил на дистрибутор



The screenshot shows a web application window titled "Main View". The main content area has a light purple background and is titled "Ticket Center" in a light blue font. Below the title, the text "Login To Your Account" is displayed in a dark blue font. There is a row of four cartoon avatars: a man with a beard and tie, a man with a beard and hat, a woman with brown hair, and a woman with blonde hair. Below the avatars, there are two input fields: "Username:" with the text "maria\_stoqnova" and "Password:" with a masked password represented by dots. At the bottom, there is a "Login as:" label followed by three buttons: "Admin", "Organizer", and "Distributor".

Main View

*Ticket Center*

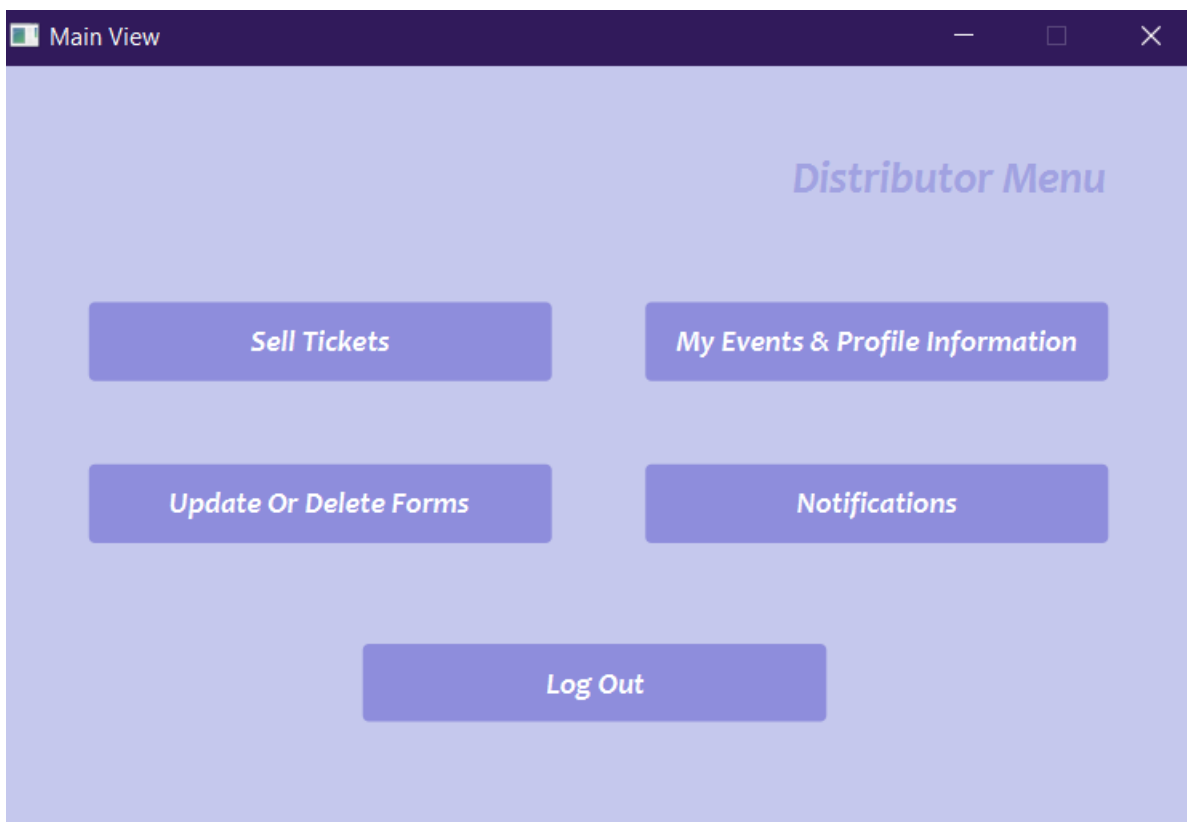
**Login To Your Account**

Username: maria\_stoqnova

Password: .....

Login as:

Admin Organizer Distributor



The screenshot shows a web application window titled "Main View". The main content area has a light purple background and is titled "Distributor Menu" in a light blue font. There are four buttons arranged in a 2x2 grid: "Sell Tickets", "My Events & Profile Information", "Update Or Delete Forms", and "Notifications". At the bottom center, there is a "Log Out" button.

Main View

*Distributor Menu*

Sell Tickets

My Events & Profile Information

Update Or Delete Forms

Notifications

Log Out

Main View

## Ticket Sales Form

Saved Successfully!

**Client First Name :**  
Ivayla

**Event Name :**  
new event

**Client Last Name :**  
Georgieva

**Place Type :**  
place\_type1

**Client Phone Number :**  
+359 877455471

**Price :**  
28.5

OK

Return

Main View

## Update / Delete Forms

**Client Name :**  
ivayla georgieva

Load Client Name

**Enter Phone Number :**  
+359 877455471

Delete

**New Phone Number :**  
+359 877655412

Update

Return To The Menu

Main View

### Profile Information And Rating:

**From Date :** 1.02.2022 r.

**To Date:** 1.03.2022 r.

**Main Information:**

Name: maria stoqnova | Username: maria\_stoqnova | Password: 202016a | E-mail: maria\_stoqnova@abv.bg | Phone Number: +3598

**Events:**

**Summary Rating:** 62.5

Distributor: maria\_stoqnova Event: event1 2022-02-19 Number Of Tickets:250 Sold Tickets: 9  
Distributor: maria\_stoqnova Event: event3 2022-02-26 Number Of Tickets:200 Sold Tickets: 0  
Distributor: maria\_stoqnova Event: event4 2022-02-27 Number Of Tickets:350 Sold Tickets: 0

Show

Return To The Menu

Main View

Today is : 2022-02-14

Notifications

**New Events :**

Added: 2022-02-11 Event: event1 Date: 2022-02-19 Number Of Places: 250 Sold: 9  
Added: 2022-02-11 Event: event2 Date: 2023-02-17 Number Of Places: 20 Sold: 0  
Added: 2022-02-11 Event: event3 Date: 2022-02-26 Number Of Places: 200 Sold: 0  
Added: 2022-02-11 Event: event4 Date: 2022-02-27 Number Of Places: 350 Sold: 0  
Added: 2022-02-14 Event: new event Date: 2022-03-01 Number Of Places: 300 Sold: 1

**Events This Month :**

Event Name : event1 Date : 2022-02-19 Number Of Places : 250 Sold: 9  
Event Name : event3 Date : 2022-02-26 Number Of Places : 200 Sold: 0  
Event Name : event4 Date : 2022-02-27 Number Of Places : 350 Sold: 0

Return