**Технически университет – Варна**

Факултет: ФИТА

Катедра: Софтуерни и интернет технологии (СИТ)

Специалност: СИТ

Тема на проекта:

Билетен център

Изготвили:

Теодор Николаев Николов Факултетен номер: 20621504

Ангел Петров Петров Факултетен номер: 20621510

Структура на документацията

1. Увод
   1. Задание
   2. План за разработка на проекта
2. Анализ на проблема
   1. Функционални изисквания - UML Use Case
   2. Структура на проекта
3. Проектиране на системата
   1. Activity Diagrams
   2. Релационен модел на базата данни
   3. Концептуален модел на базата от данни (ER диаграма (модел на Чен))
   4. GUI и бизнес логика
   5. Class Diagrams и Sequence diagram
4. Реализация на системата
   1. Реализация на базата данни
   2. Реализация на слоя с работа с базата данни
   3. Реализация на бизнес логика и графичен интерфейс
   4. Реализация на модул регистриране на събития в системата
5. Тестови резултати
   1. JUnit тестове

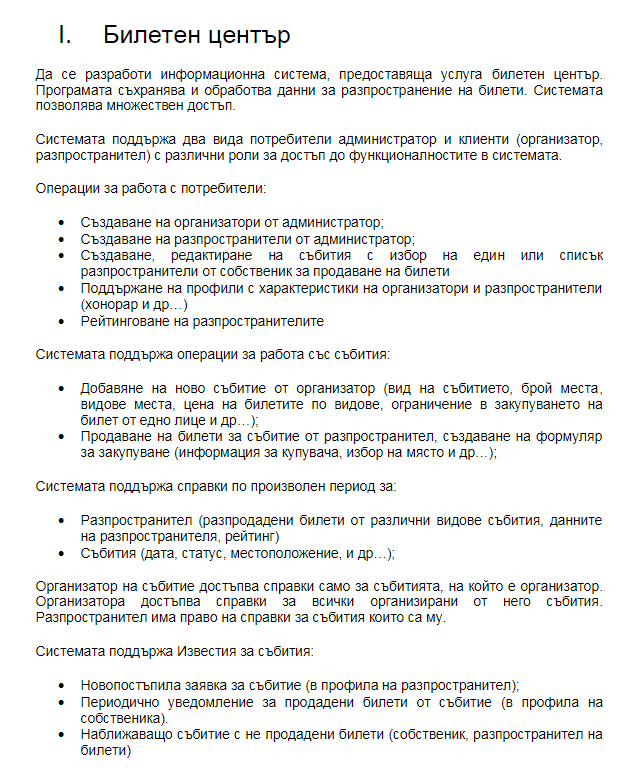
Линк към github: <https://github.com/tedonikolov/OOP2_project>

Използвана литература:

1. [YouTube клип за използване на JavaFX](https://www.youtube.com/watch?v=9XJicRt_FaI&ab_channel=BroCode)
2. [Oracle SQL документация](https://docs.oracle.com/cd/B19306_01/server.102/b14200/toc.htm)
3. [Хеширане на парола](https://howtodoinjava.com/java/java-security/how-to-generate-secure-password-hash-md5-sha-pbkdf2-bcrypt-examples/)
4. [JDBC свързване с базата данни](https://www.geeksforgeeks.org/establishing-jdbc-connection-in-java/)
5. [Log4J документация](https://logging.apache.org/log4j/2.x/)

1.Увод

а. Задание



b. План за разработка на проекта

1. Запознаване със заданието на проекта
2. Анализиране на проблема
3. Създаване на UML use case диаграма
4. Създаване на repository в GitHub
5. Избиране на технологиите за реализация на системата.
6. Проектиране на базата от данни чрез oracle (релационен модел)
7. Генериране на базата данни чрез SQL команди
8. Създаване на интерфейс на системата чрез JavaFX
9. Свързване на базата данни с програмата чрез JDBC
10. Създаване на класовете отговарящи на базата данни
11. Създаване на функционалностите на администратора (създаване на потребители)
12. Създаване на функционалностите на организатора (създаване на събития, назначаване на разпространители, рейтинговане на разпространителите)
13. Създаване на функционалностите на разпространителя
14. Създаване на формуляр за купуване на билети
15. Създаване на справки по произволен период за разпространители
16. Създаване на справки по произволен период за събития
17. Създаване на система за известия на събития
18. Създаване на тестови резултати
19. Създаване на Class Diagram, ER Diagram, Sequence diagram
20. Финализиране на документацията

2. Анализ на проблема

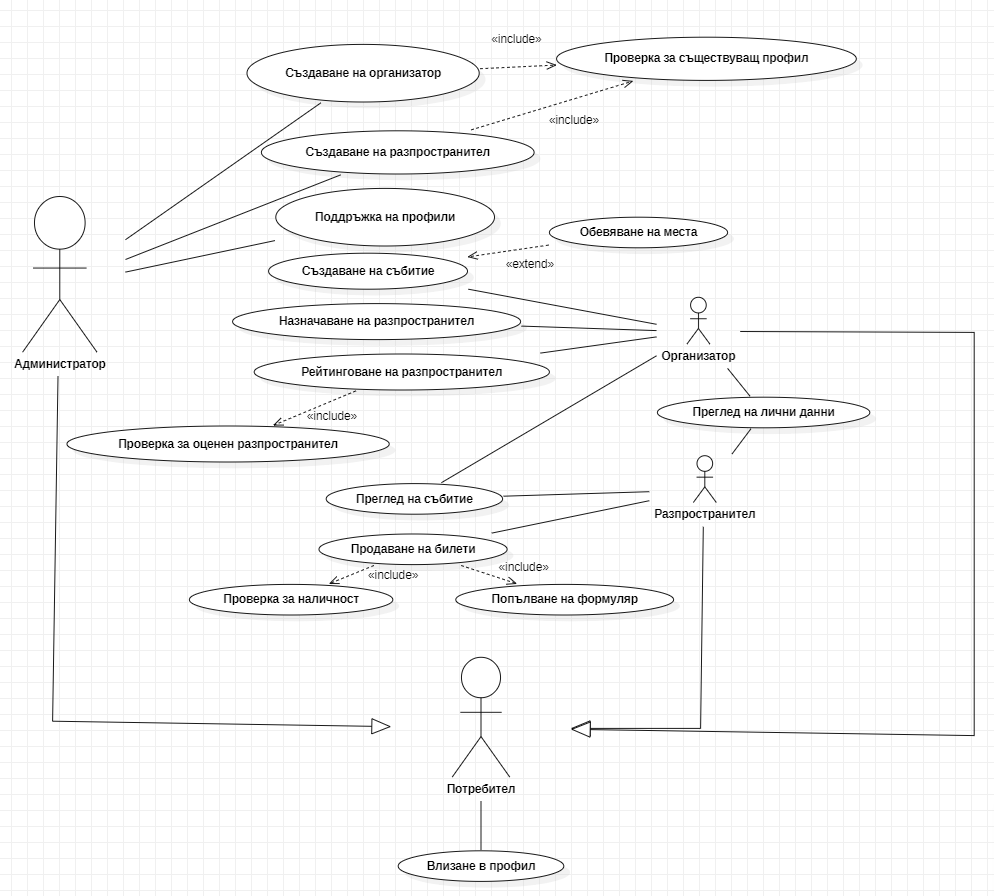
Информационна система предоставя услугата билетен център. Програмата съхранява и обработва данни за разпространение на билети. Като системата поддържа два вида потребители: администратор и клиенти (организатор, разпространител) с различни роли за достъп до функционалностите в системата. За целта е използван начален екран за логване, като в зависимост от ролята на акаунта, който е съхранен в базата данни се активиран един от трите интерфейса: админски, оранизаторски или разпорстранителски.

1. Функционални изисквания.

Операциите, с които разполага администратора са: Създаване на организатори, разпространители; Проверка на същестуващите профили; Промяна на парола; Изтриване на съществуващ профил, който е неактивен( не е изполван в базата данни).

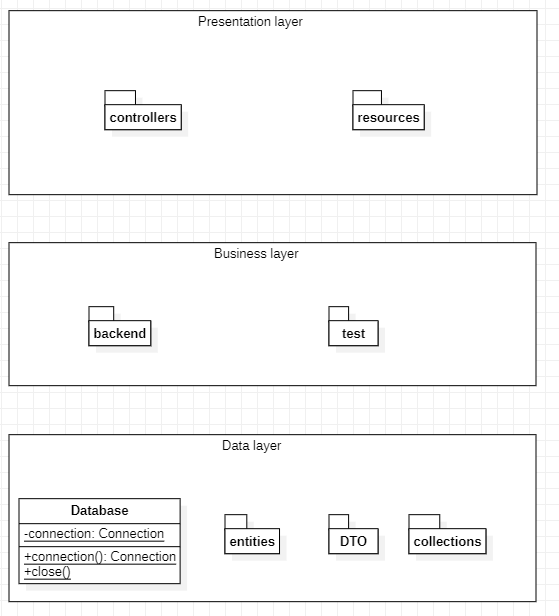
Операциите, с които разполага организтора са: Проверка на профила и коригиране на лични данни; Създаване на събития; Редактиране на събития чрез добавяне на места за продаване; Назначаване на разпространител; Оценяване на разпространител; Справки за събития и разпространители.

Операциите, с които разполага разпространителя са: Проверка на профила и коригиране на лични данни; Продаване на билети чрез формуляр; Справки за събтия.



* 1. Структура на проекта

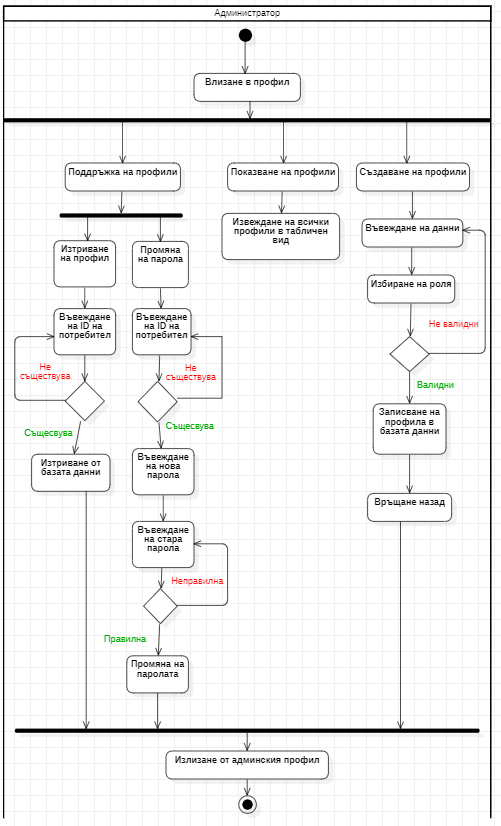
За реализацията на системата е използвана многослойна (трислойна) архитектура. Тъй като по този начин интерфейса, обработката на приложението и съхраняването на данните в база от данни са разделени на отделни модули. Тъй като в случая отделните слоеве могат да бъдат включени в отделни проекти: Презентационият слой се състой от потребителския интерфейс и контролерите, които го управлияват; Бизнес логиката контролира функционалността на приложението като извършва различни процеси по обработката на данните; Слоя от данни се състой от сървър база данни, тук се обработва информацията от базата данни.

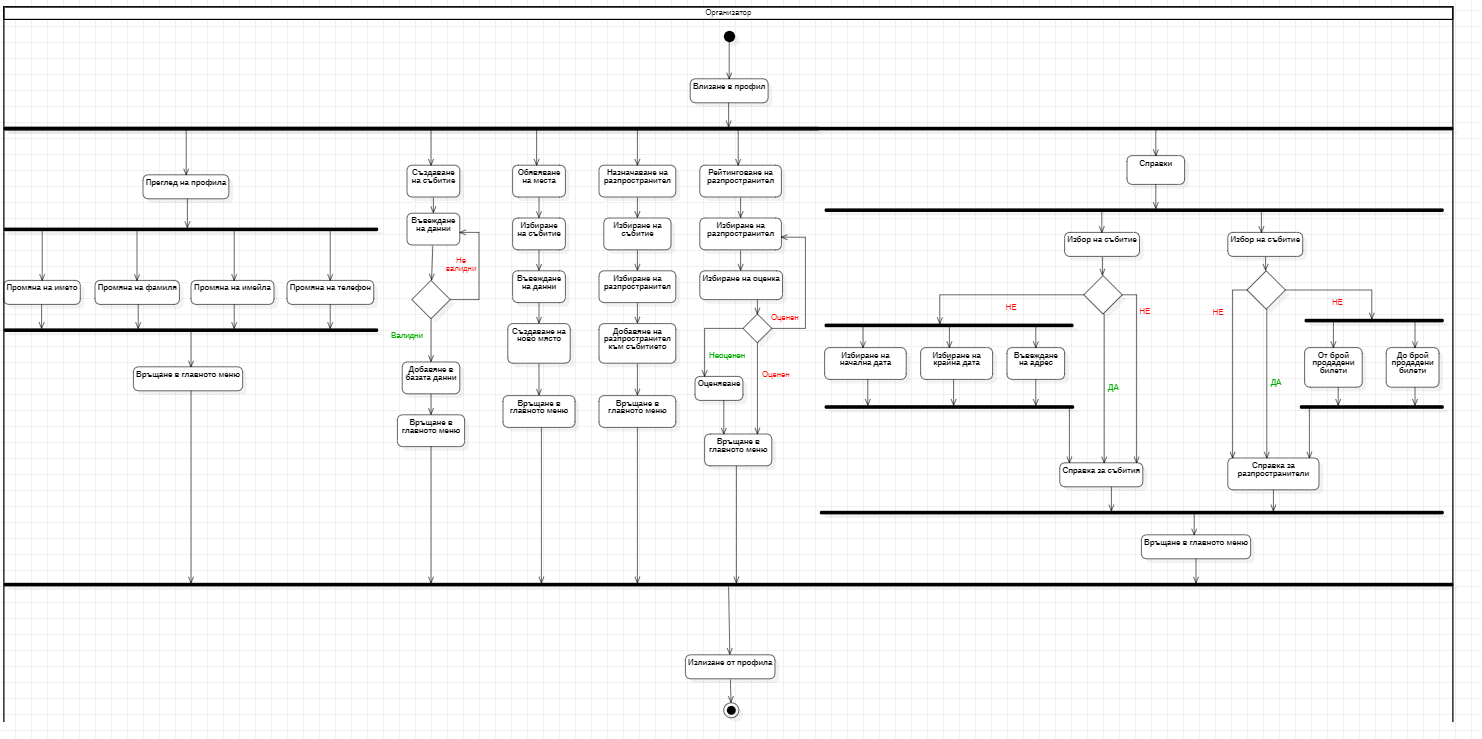


3. Проектиране на системата.

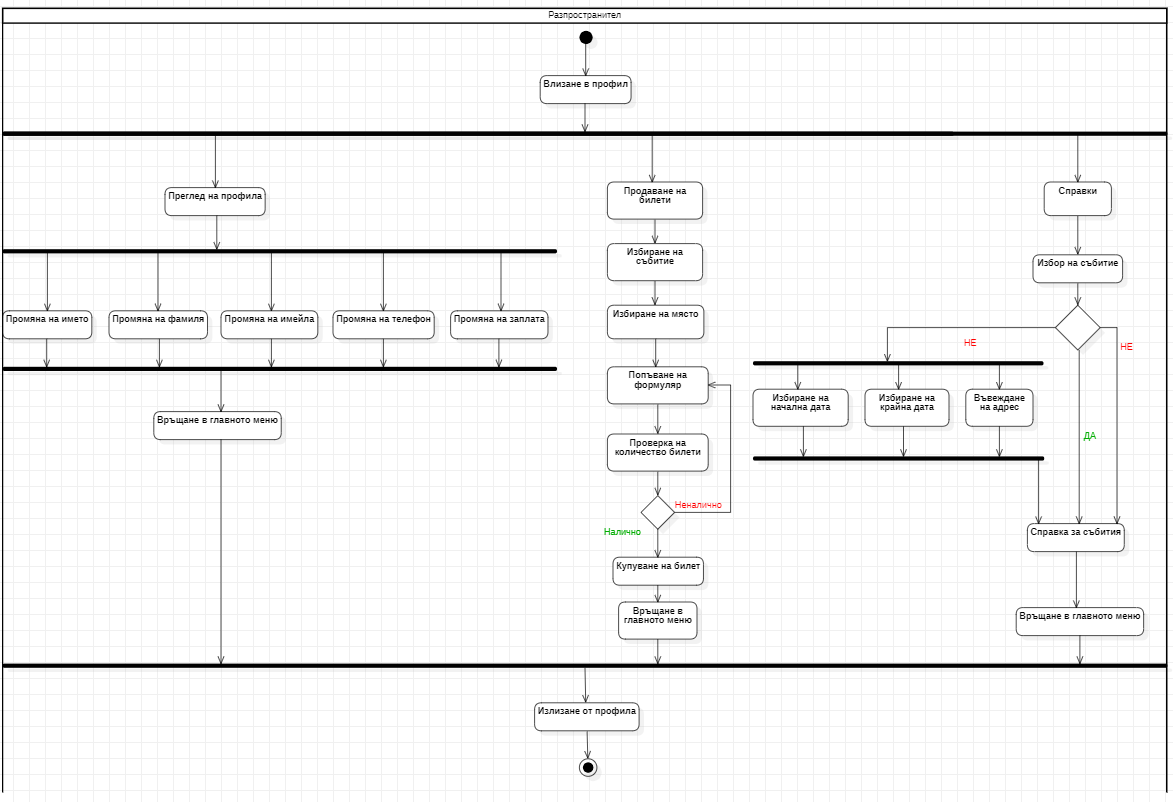
a. Activity Diagrams

– Фукционалността на администратора:



– Фукционалността на организатора:

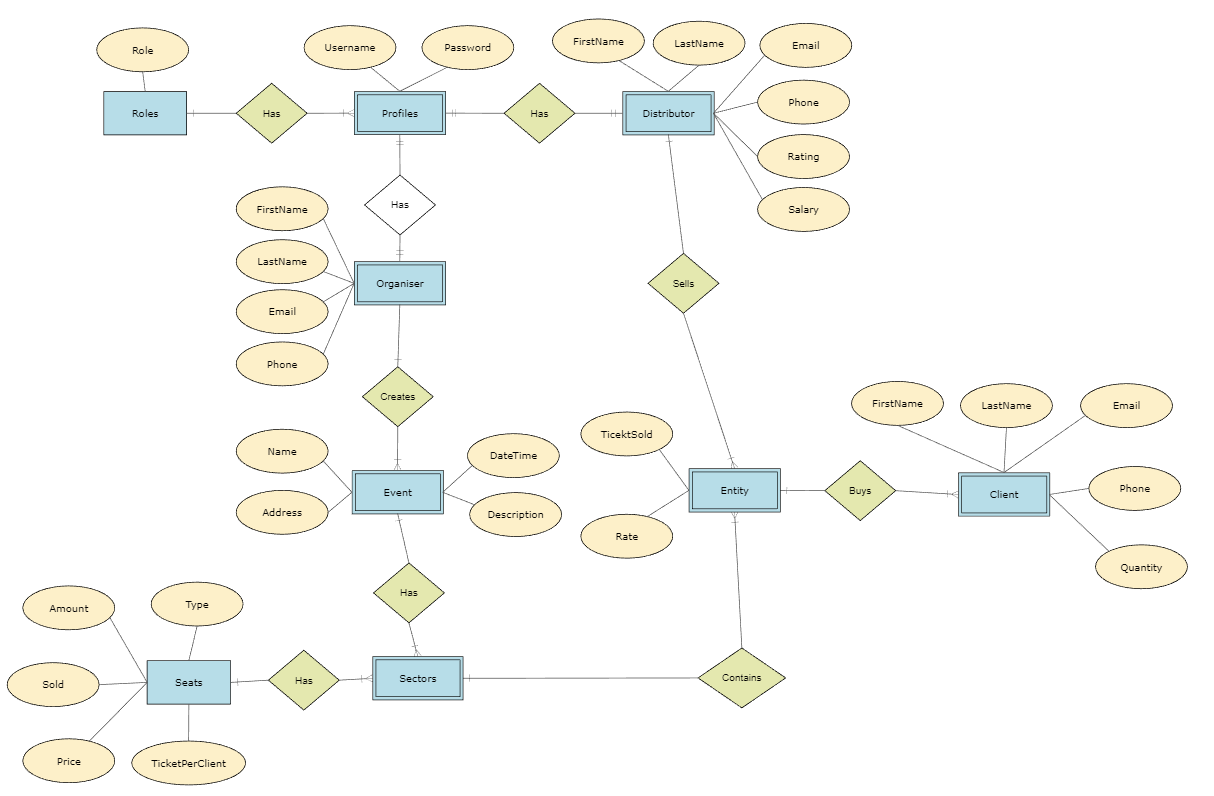
– Фукционалността на разпространителя:

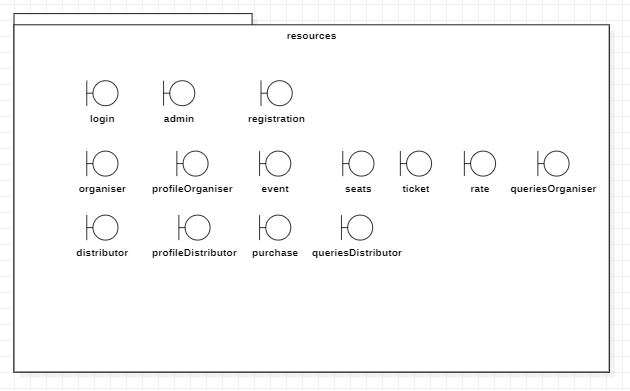


1. Релационен модел на базата данни

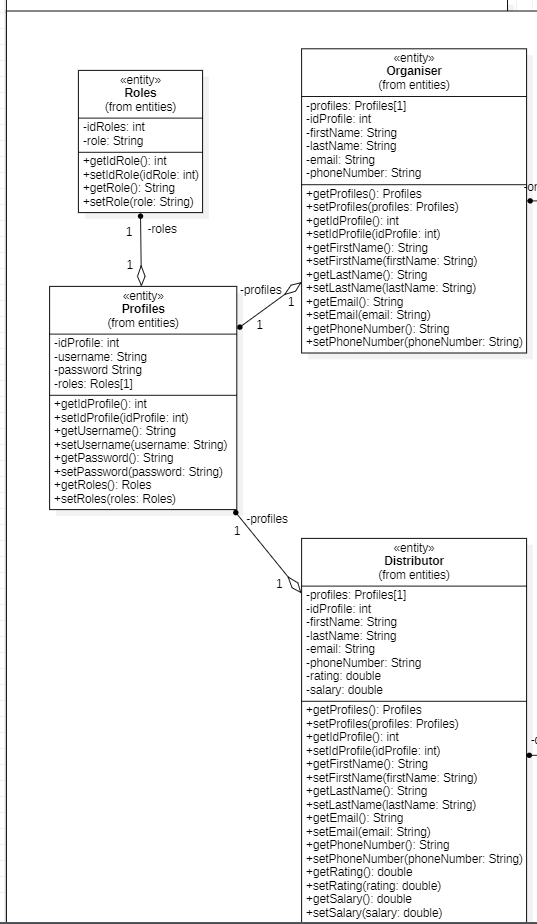
Diagram

Description automatically generated

1. Концептуален модел на базата от данни (ER диаграма (модел на Чен))
2. GUI и бизнес логика



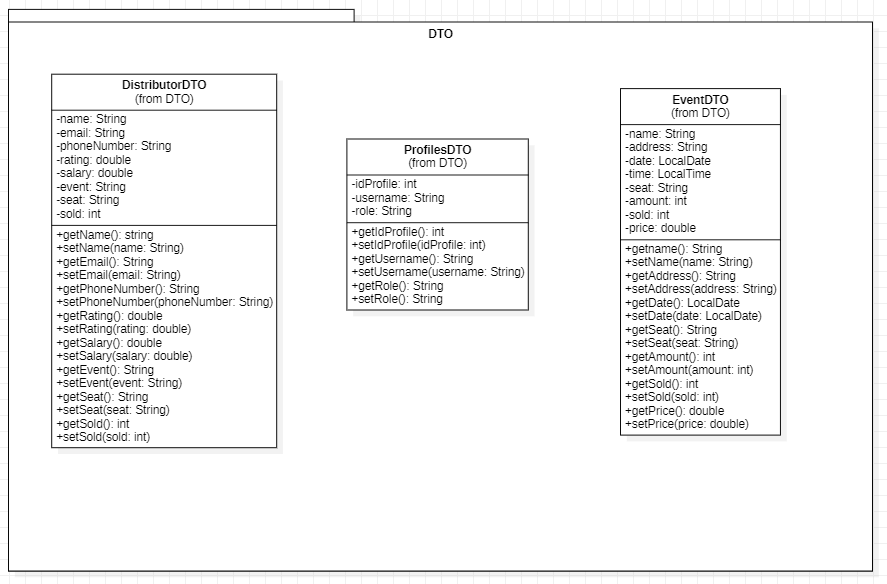
– Класове отговарящи на таблиците от базата данни.



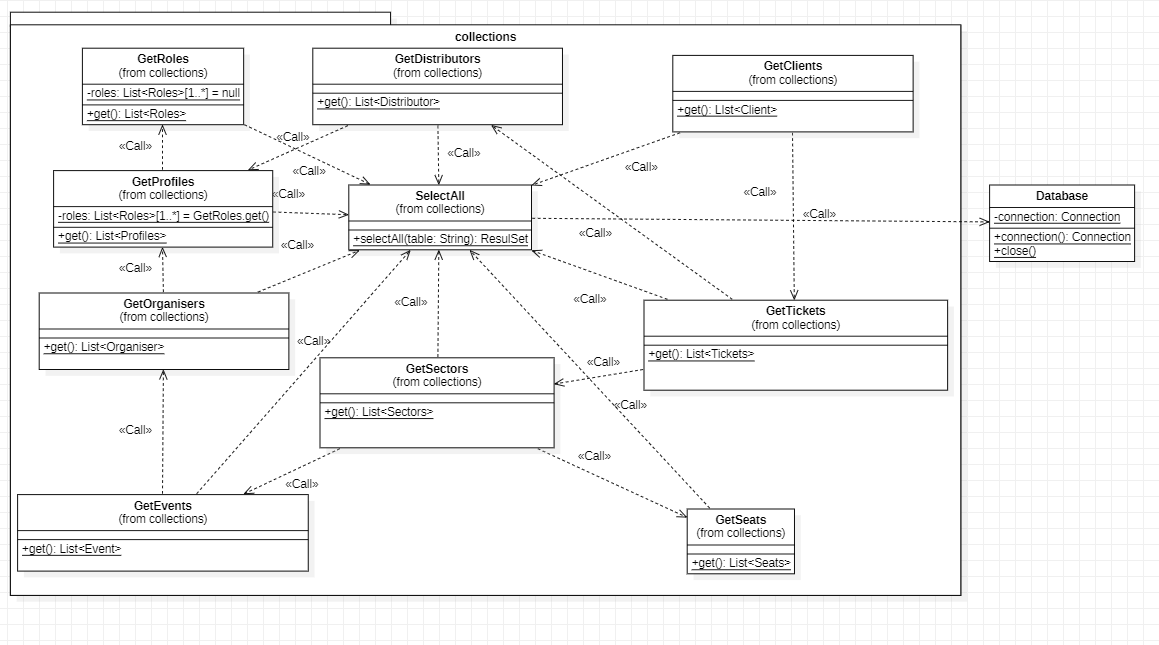
A diagram of a flowchart

Description automatically generated with low confidence

* Класове отговарящи за създаването на таблици в интерфейса.



– Извличане на данните от базата данни



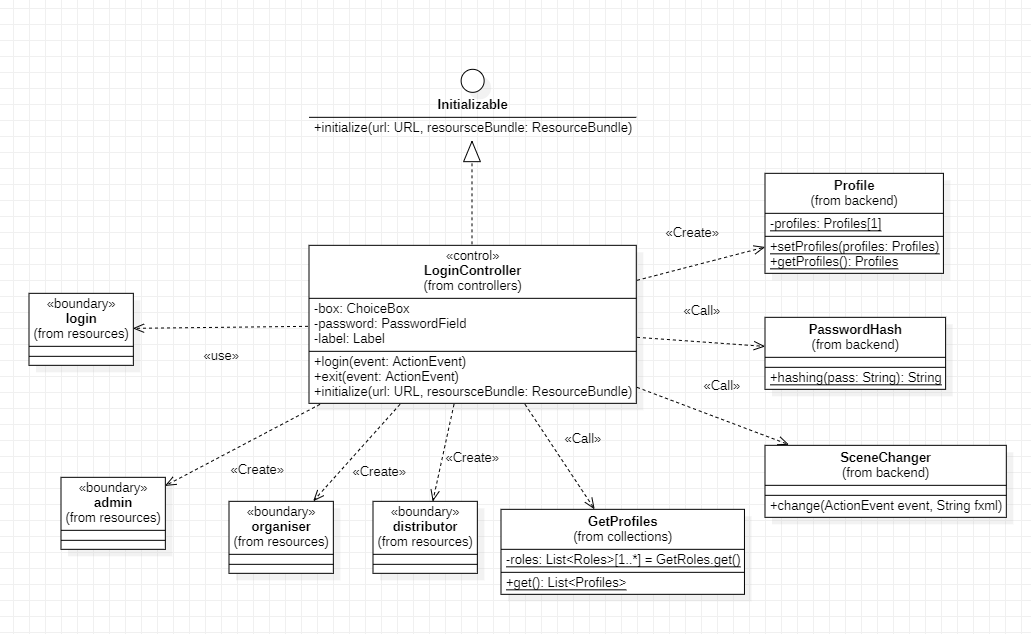
Diagram

Description automatically generatedDiagram

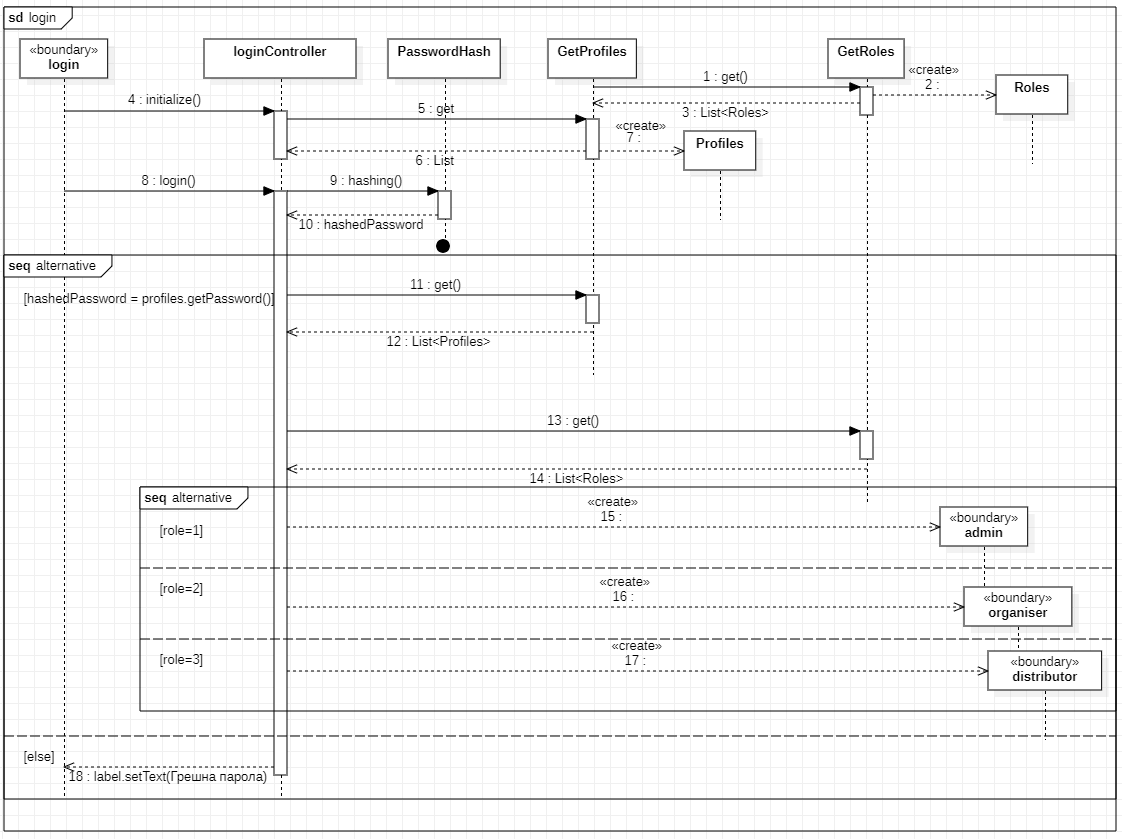
Description automatically generated– Реализиране на бизнес логика

1. Class Diagrams и Sequence diagram

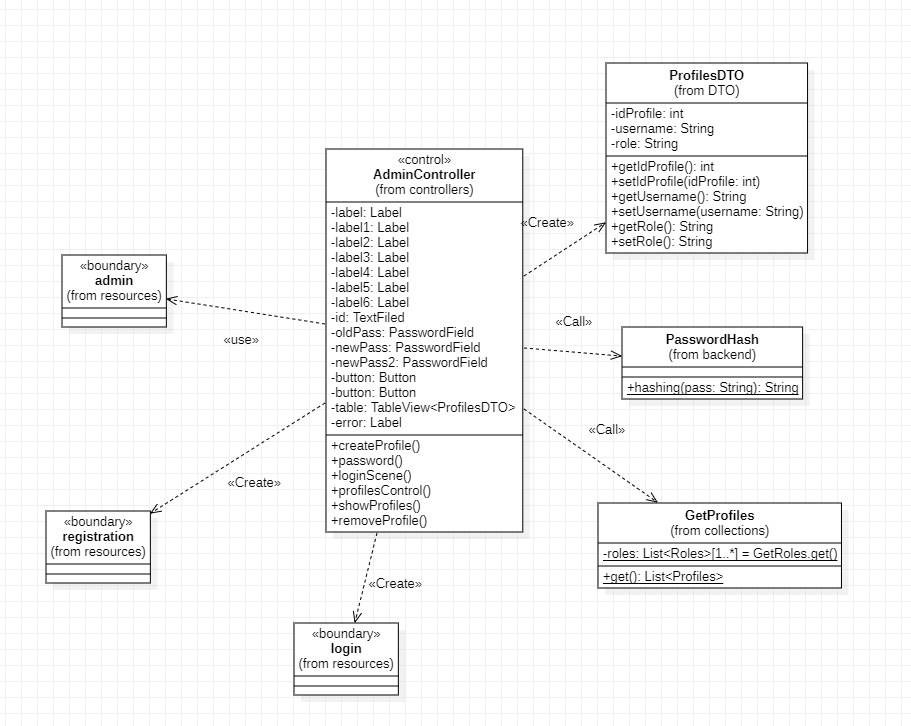
– LoginController клас, който реализира интерфейса на логина

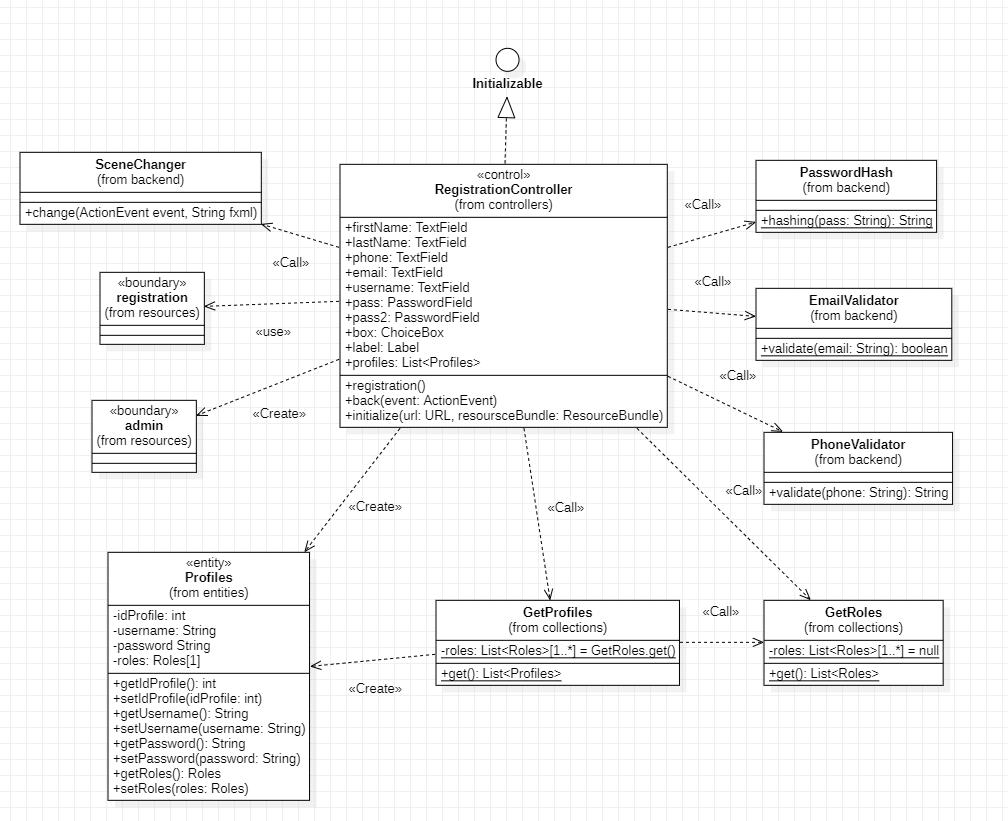


– LoginController диаграма на последователностите

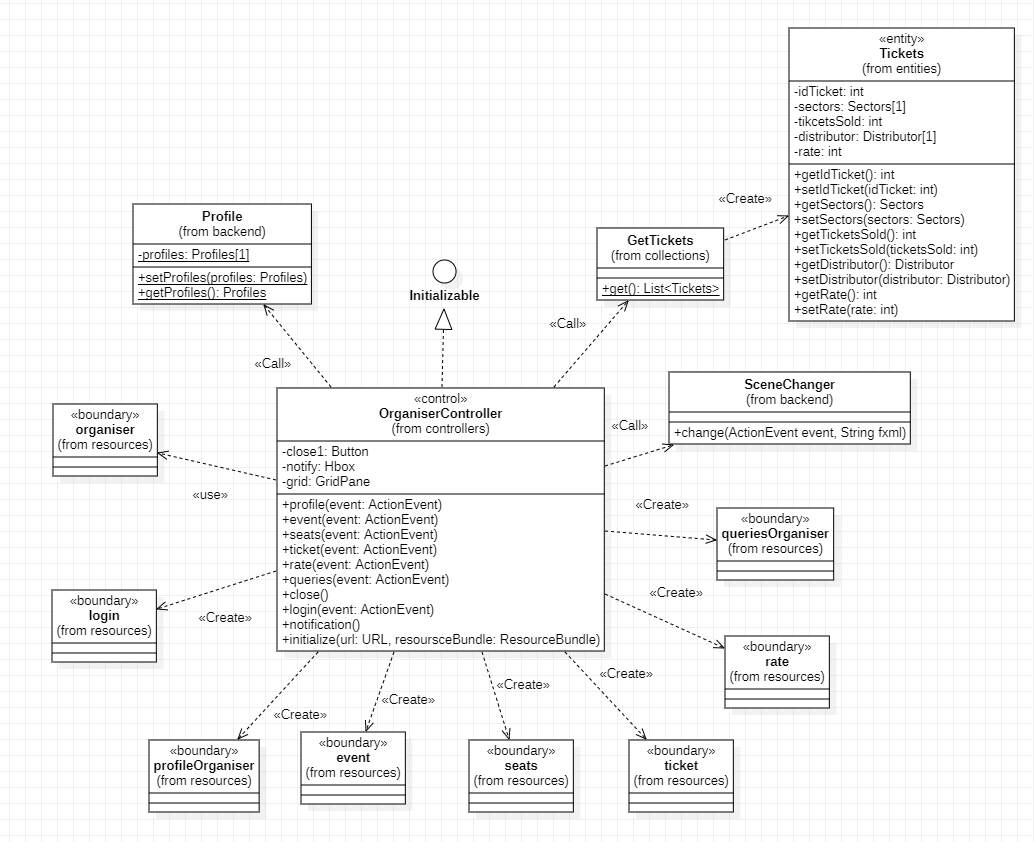


– AdminController клас, който реализира интерфейса на логина

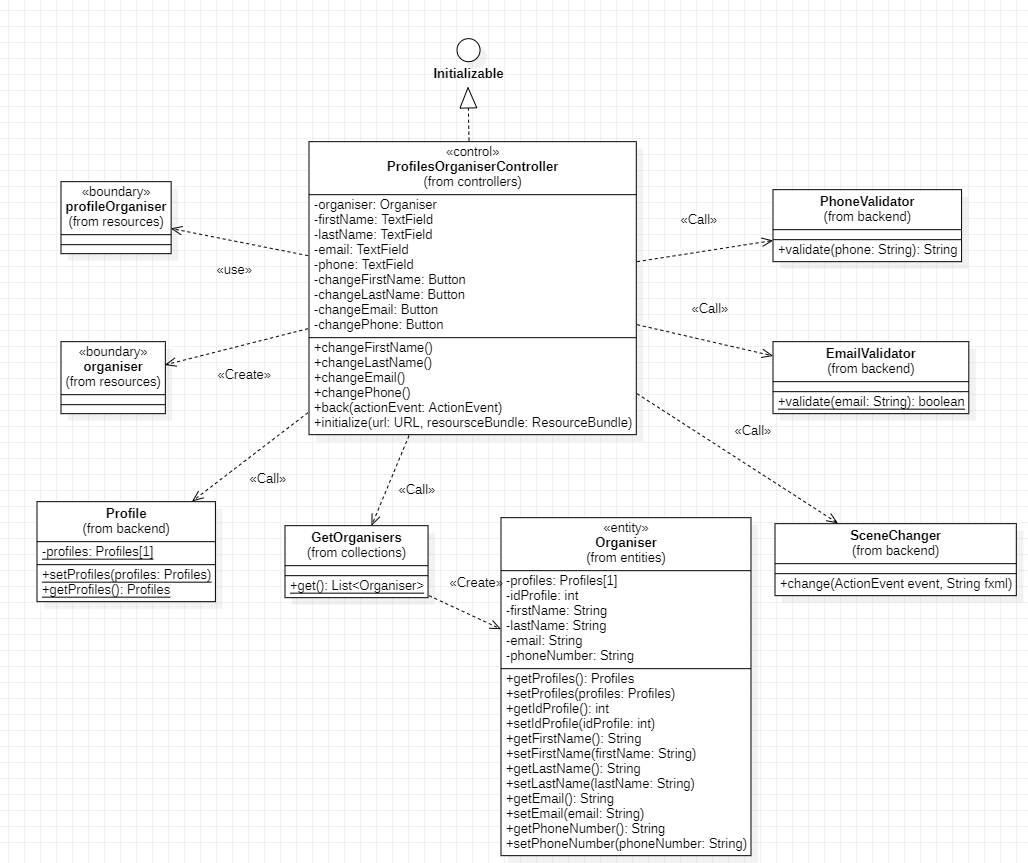


– RegitrationController клас, който реализира интерфейса за създаване на нови профили

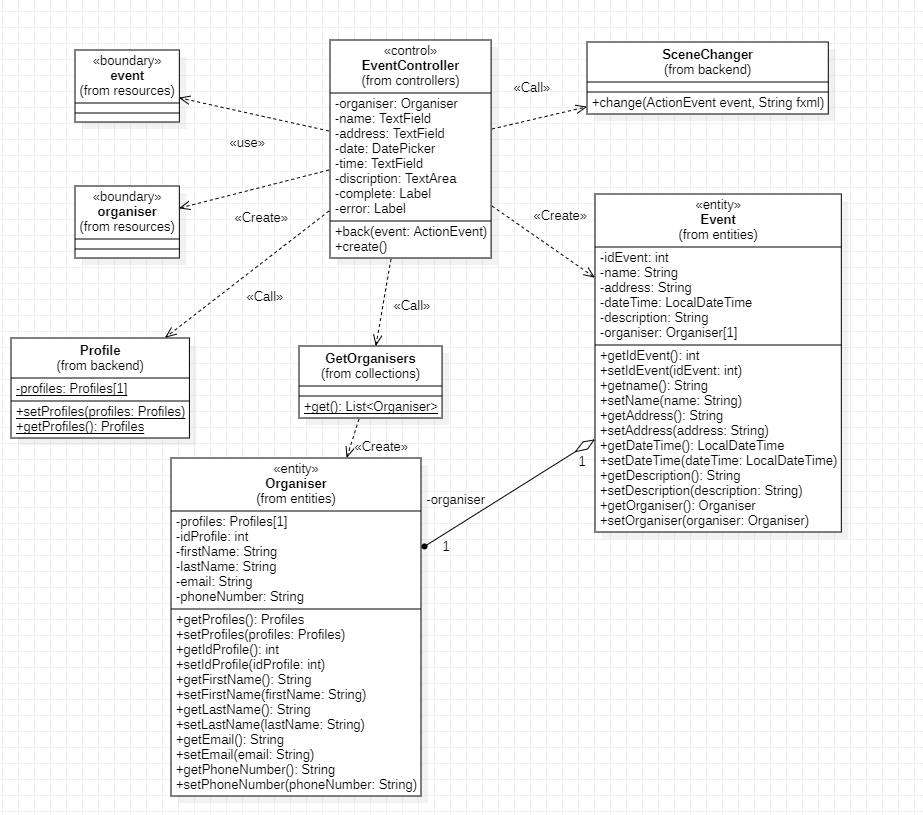
– OrganiserController клас, който реализира интерфейса на организатора



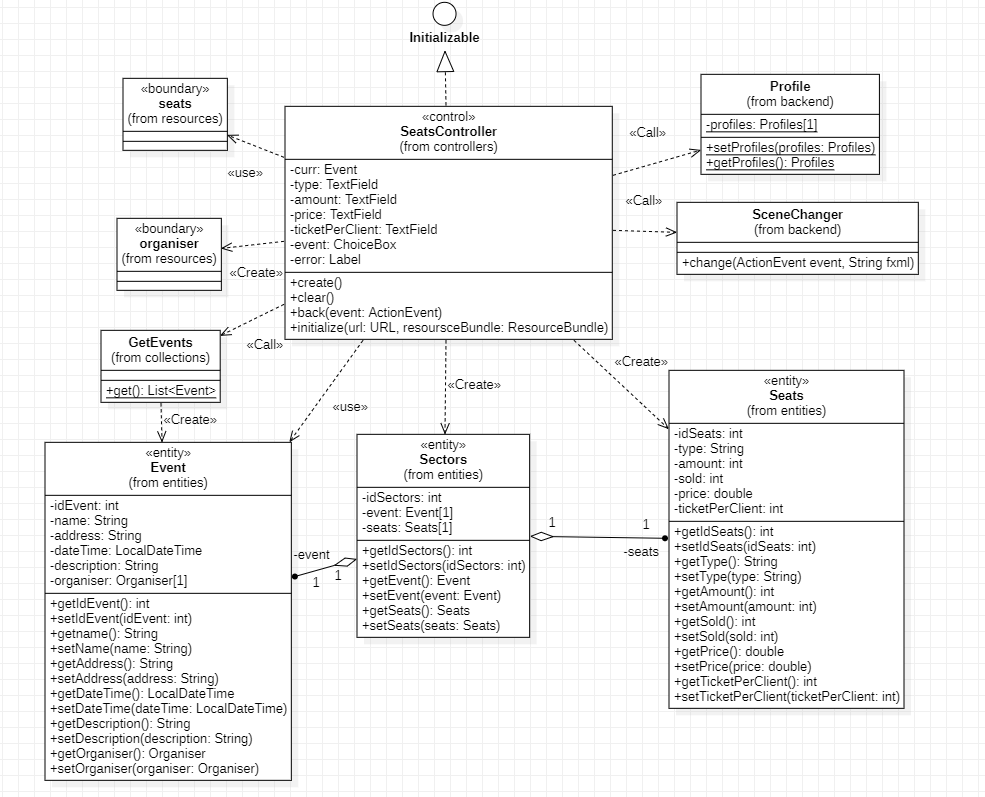
– ProfilesOrganiserController клас, който реализира интерфейса на профила на организатора

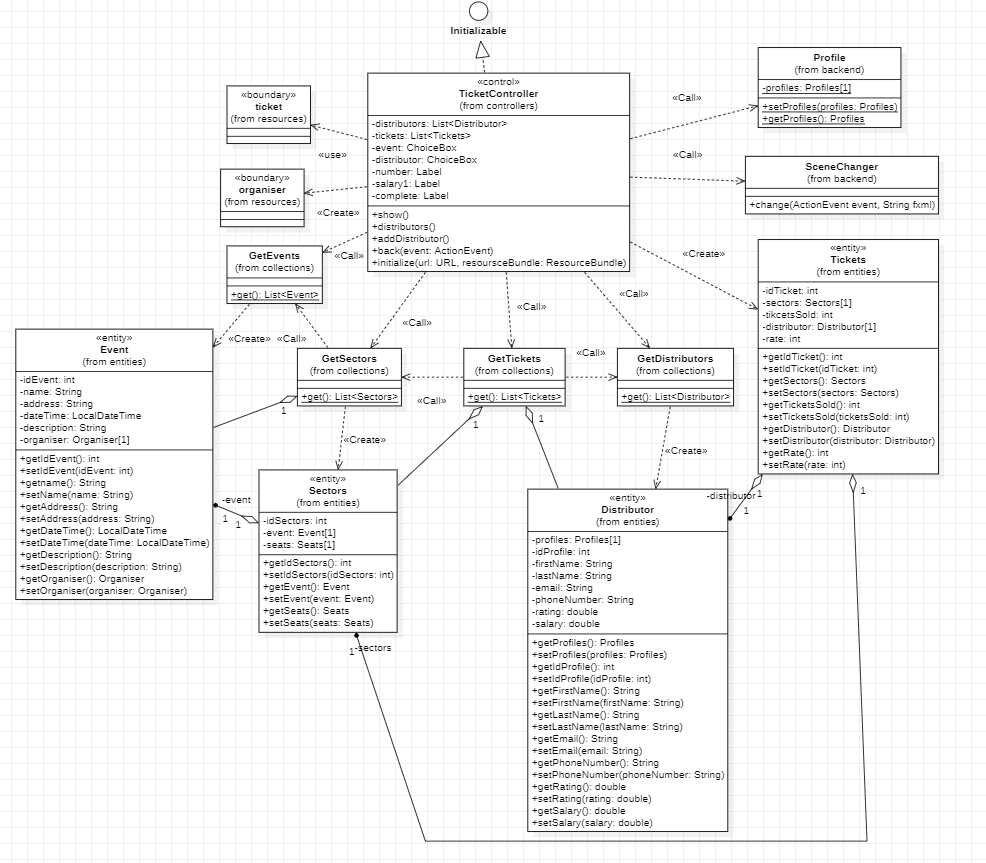


– EventController клас, който реализира интерфейса за създаване на събития

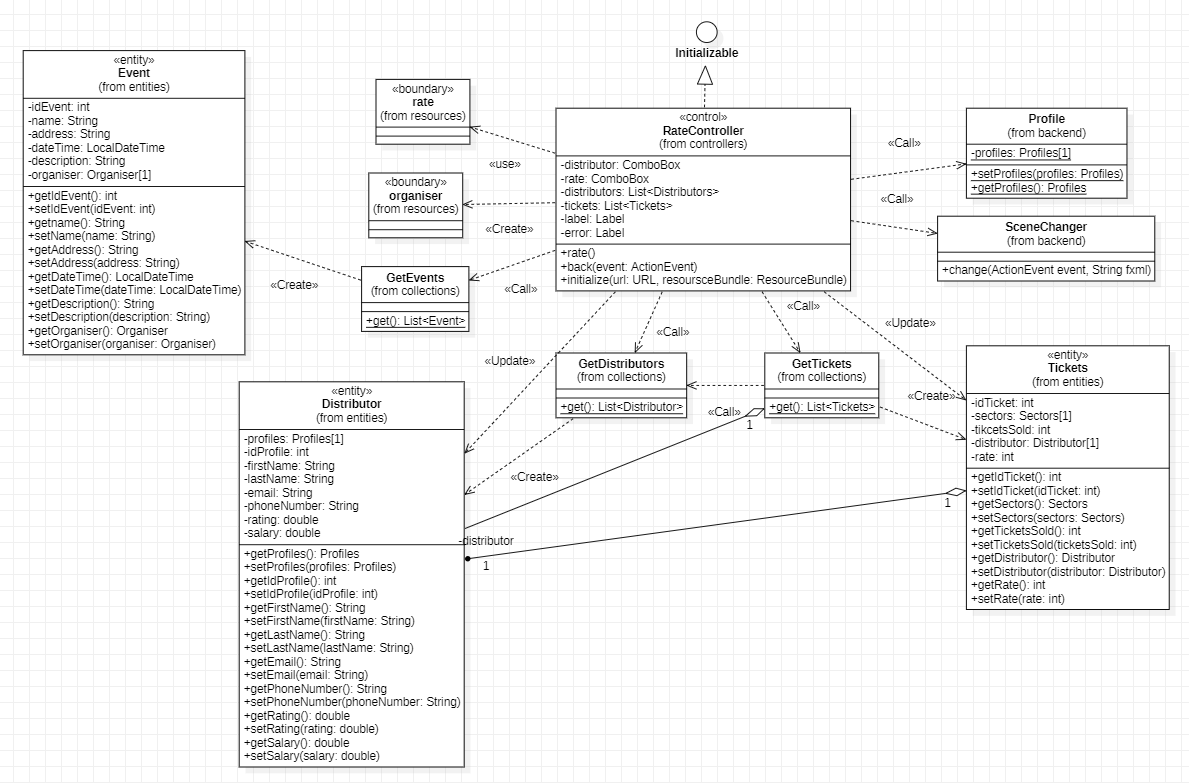


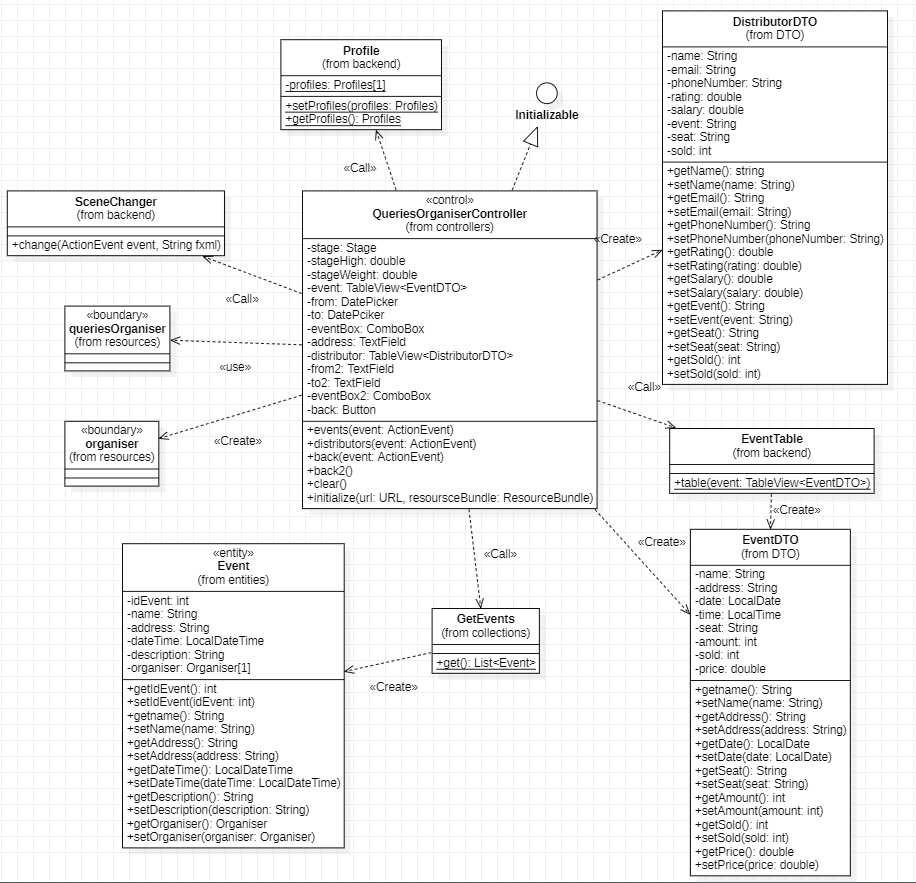
– SeatsController клас, който реализира интерфейса за добавяне на места за събитие



– TicketController клас, който реализира интерфейса за назначаване на разпространители

– RateController клас, който реализира интерфейса за даване на оценка на разпространител от организатор



– QueriesOrganiserController клас, който реализира интерфейса за справки на организатора

– DistributorController клас, който реализира интерфейса на разпространителя

Diagram

Description automatically generated

– ProfileDistributorController клас, който реализира интерфeйса профила на разпространителя

Diagram

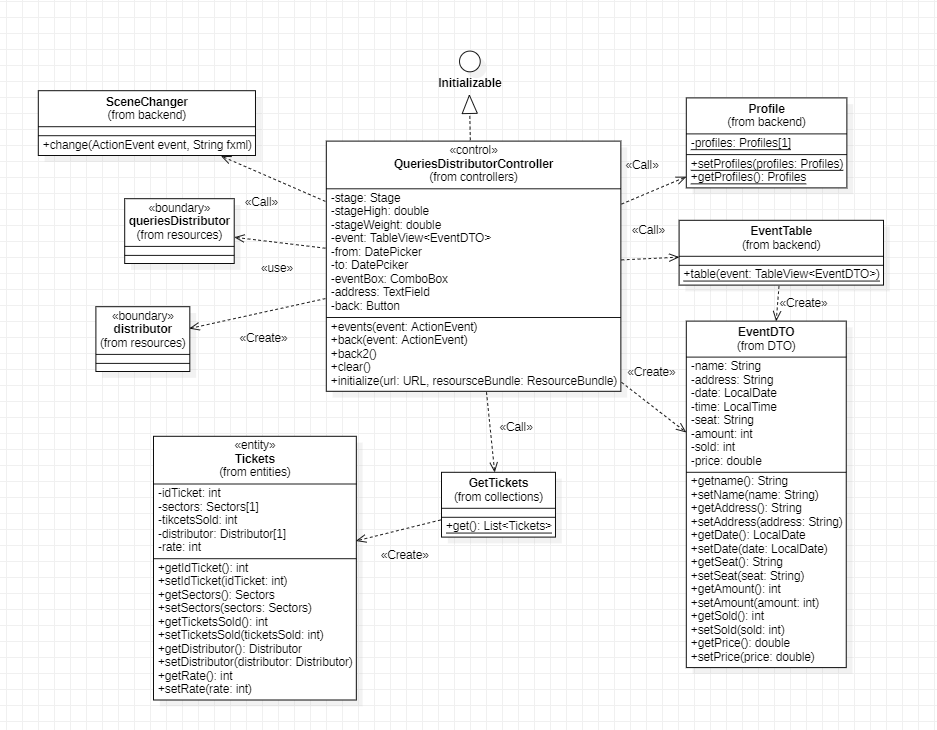
Description automatically generated

– PurchaseController клас, който реализира интерфейса на формуляра за продаване на билети

Diagram

Description automatically generated

– QueriesDistributorController клас, който реализира интерфeйса на справките на разпространителя



1. Реализация на системата
   1. Реализация на базата данни

Базата данни е реализирана чрез Oracle data base server and SQL. Базата данни е съставена от 9 таблици. Заявки за създаване на някой таблици.

Table

Description automatically generated

* 1. Реализация на слоя с работа с базата данни

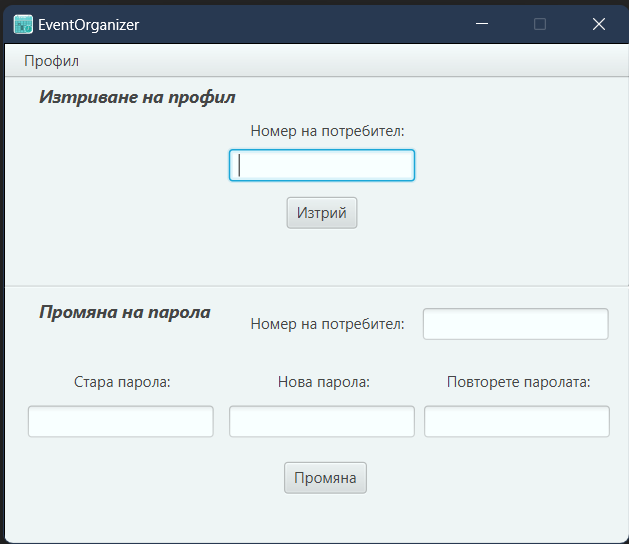
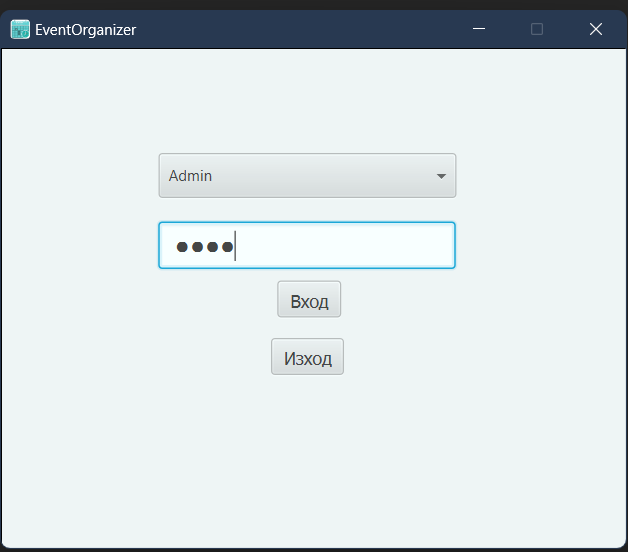
Връзката между базата данни и системата е осъществена чрез чисто JDBC.

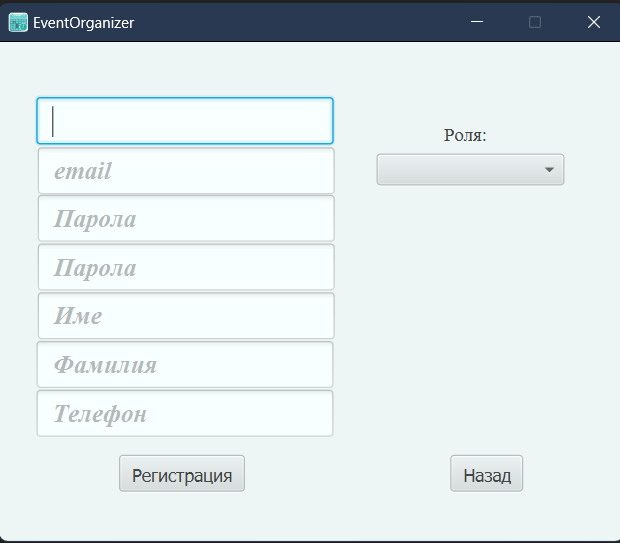
Graphical user interface, text, application

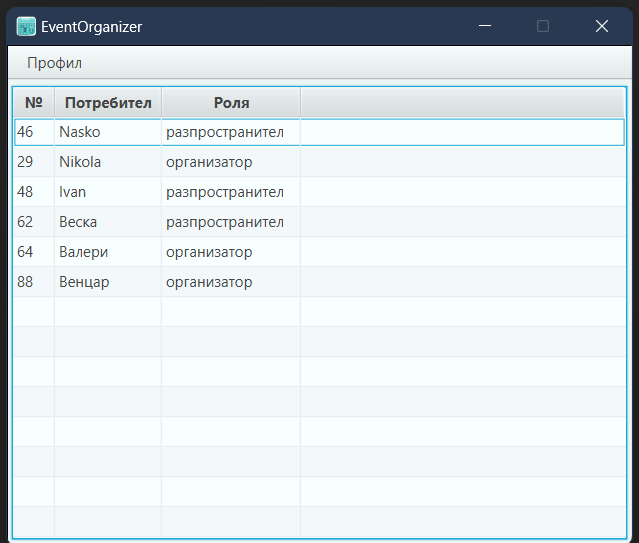
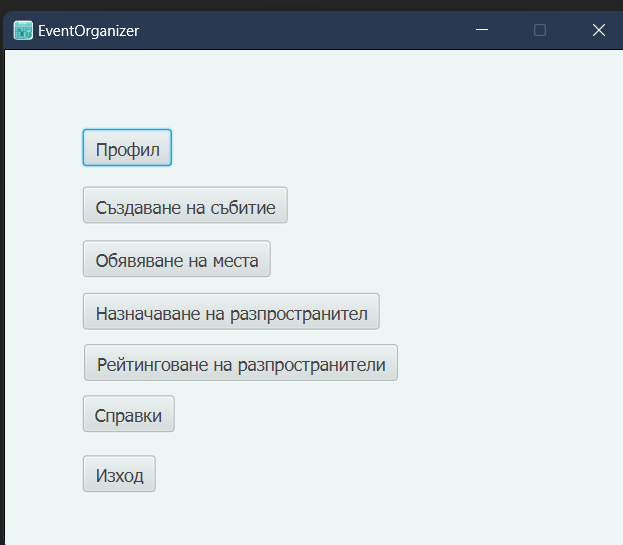
Description automatically generated Text

Description automatically generated

* 1. Реализация на бизнес логика и графичен интерфейс

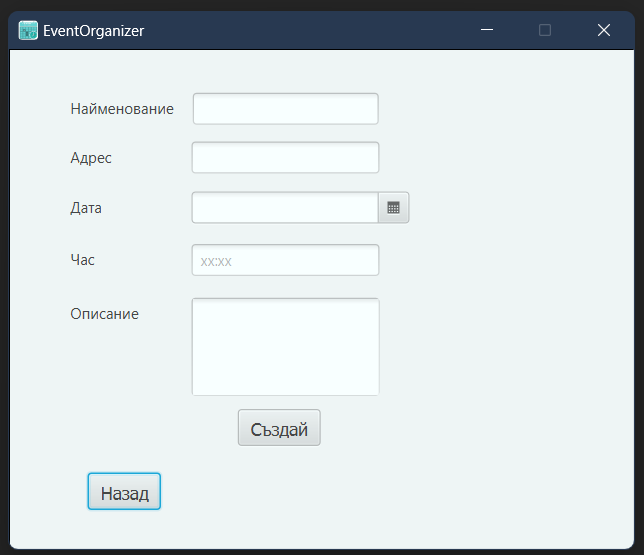
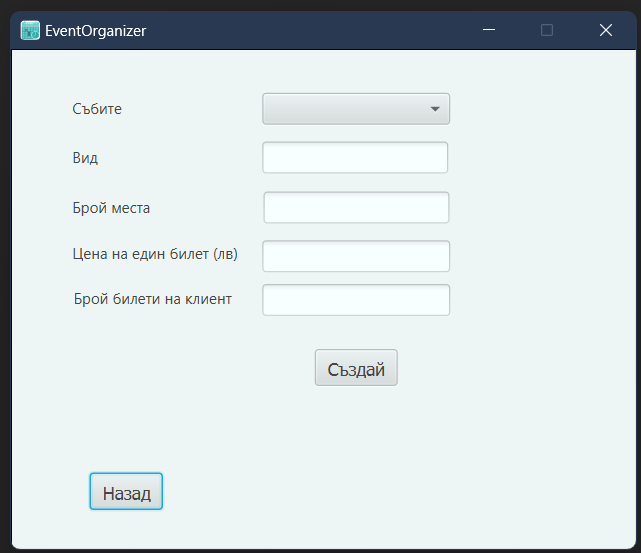
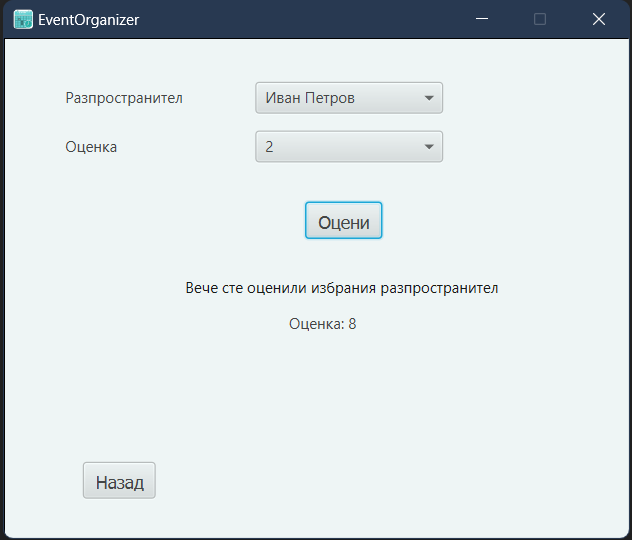
Реализацията на бизнес логиката е осъществена чрез Java, а интерфейса чрез JavaFX

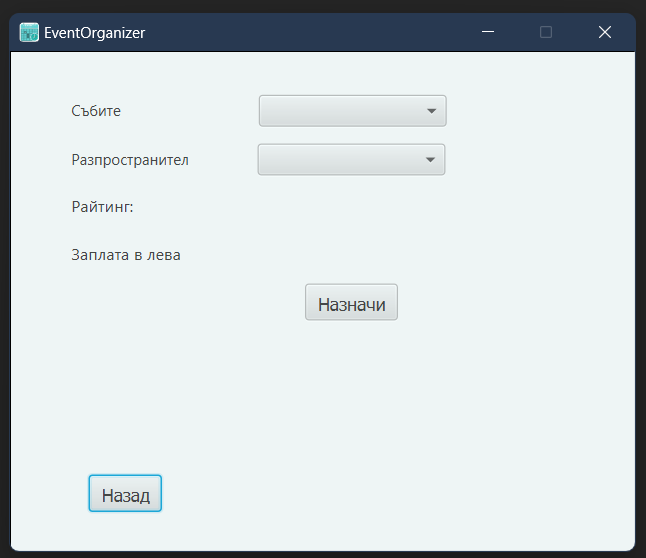




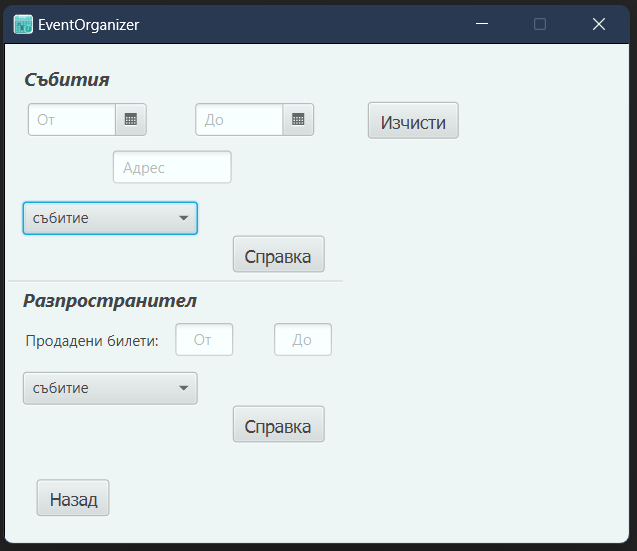
Graphical user interface, application

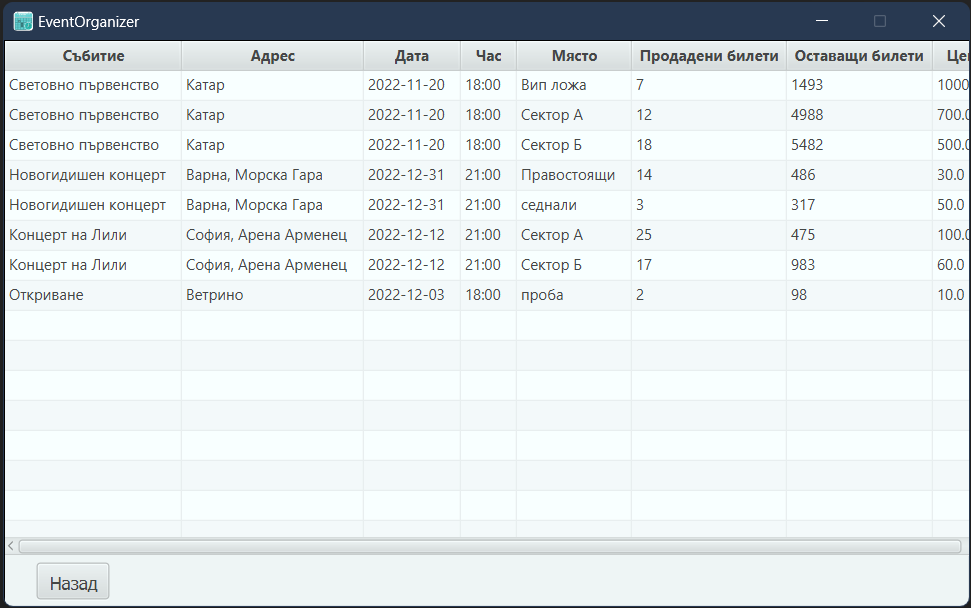
Description automatically generated





Graphical user interface, website

Description automatically generated



* 1. Реализация на модул регистриране на събития в системата

Реализацията на модула е осъщесвена чрез Log4J2. Файла за конфигуриране на log4j2 e:

Graphical user interface, text, application

Description automatically generated

В програмата логовете са изполвани в критични секции за улавяне на фатални грешки, като най-вече при работата с базата данни. Освен това са използвани за складиране на информация по време на работа на програмата. Логовете са в основата на работата с нотификации в системата, като за целта конкретна информация се запазва в определен файл с id-то на контретния организатор или разпространител.

1. Тестови резултати
   1. JUnit тестове

Направени са тестове на основните класове на бизнес логиката:

За валидация на телефония номер е използвана regex комбинация, която позволява попълването само на български телефонни номера.

Text, letter

Description automatically generatedText, letter

Description automatically generated

За валидация на имейла е използвана regex комбинация, която проверява за @.Text

Description automatically generated

Text, letter

Description automatically generated

Проверка за правено хеширана парола:

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated