**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА ФАКУЛТЕТ ПО ИЗЧИСЛИТЕЛНА ТЕХНИКА И АВТОМАТИЗАЦИЯ**

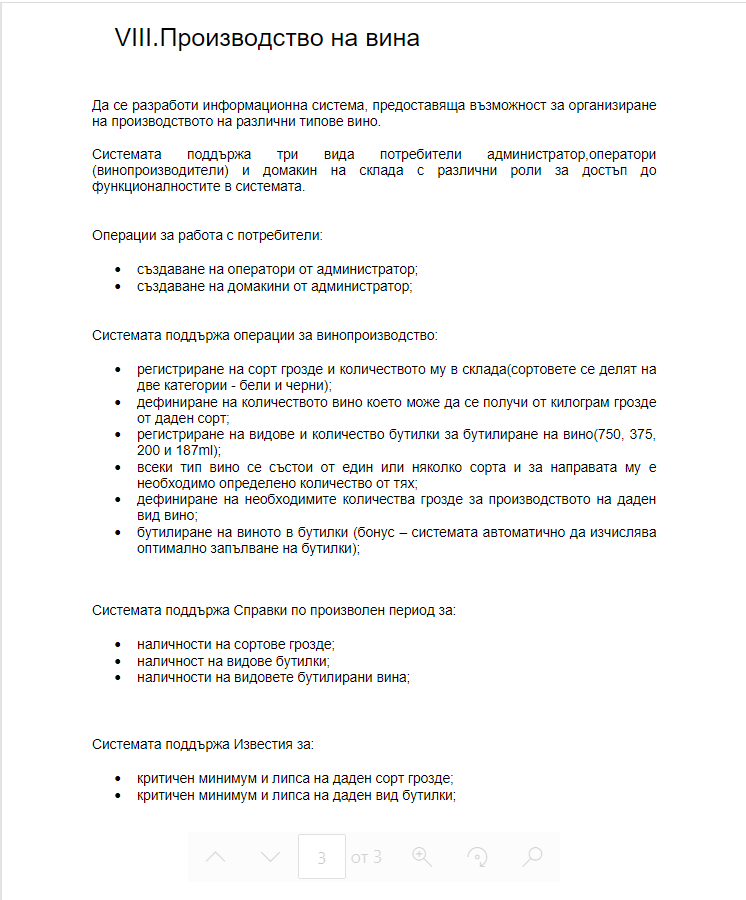
КУРСОВ ПРОЕКТ

на

Виктор Диянов Семков , фак.№ 19621834

спец.:СИТ, 3 курс, 5 група и Никола Свиленов Славчев, фак.№ 19621780, спец.:СИТ, 3 курс, 5 група

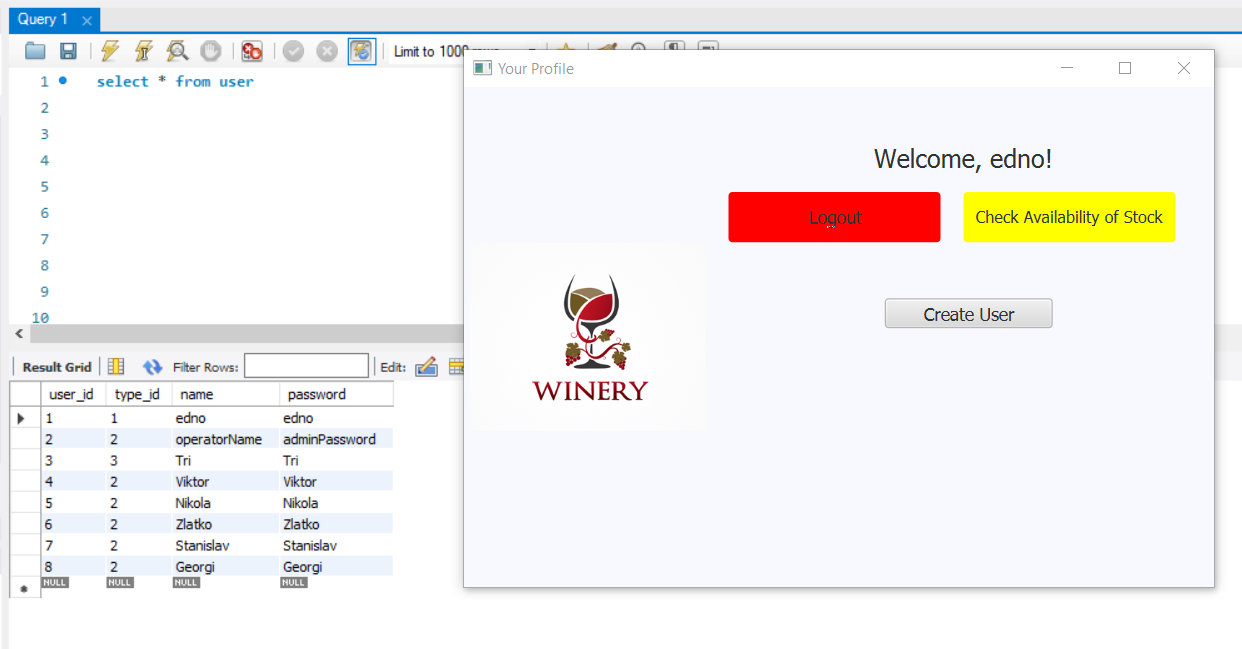
Тема № 28



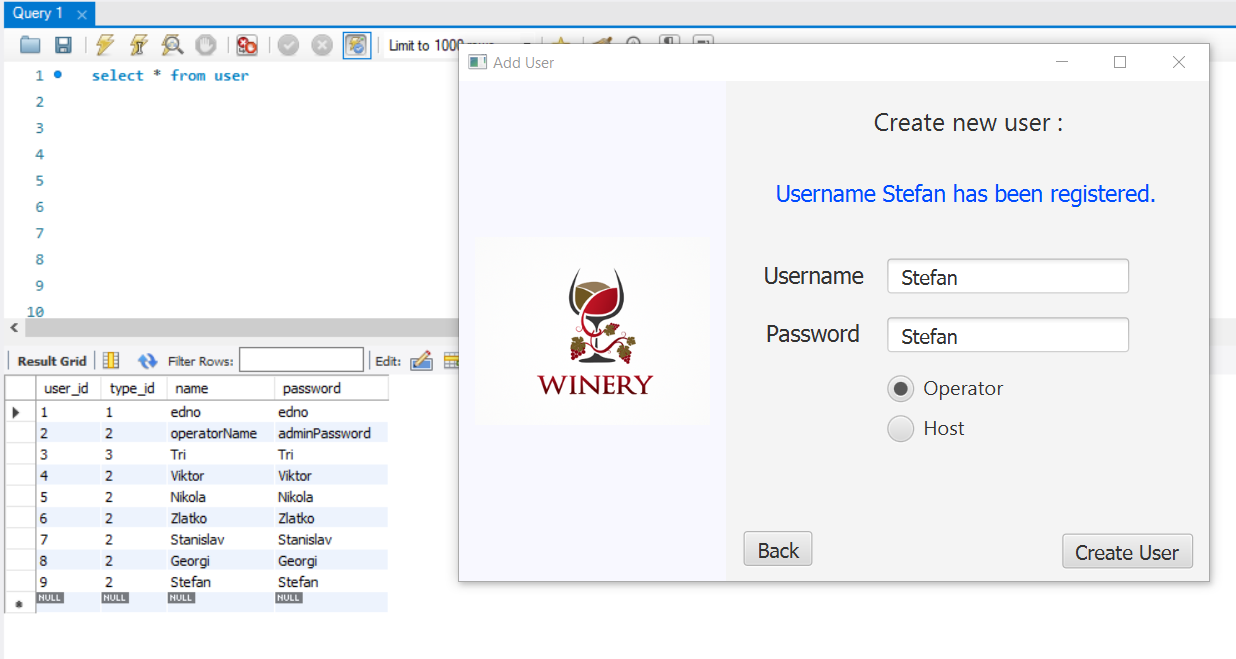
c. Анализ на проблема:

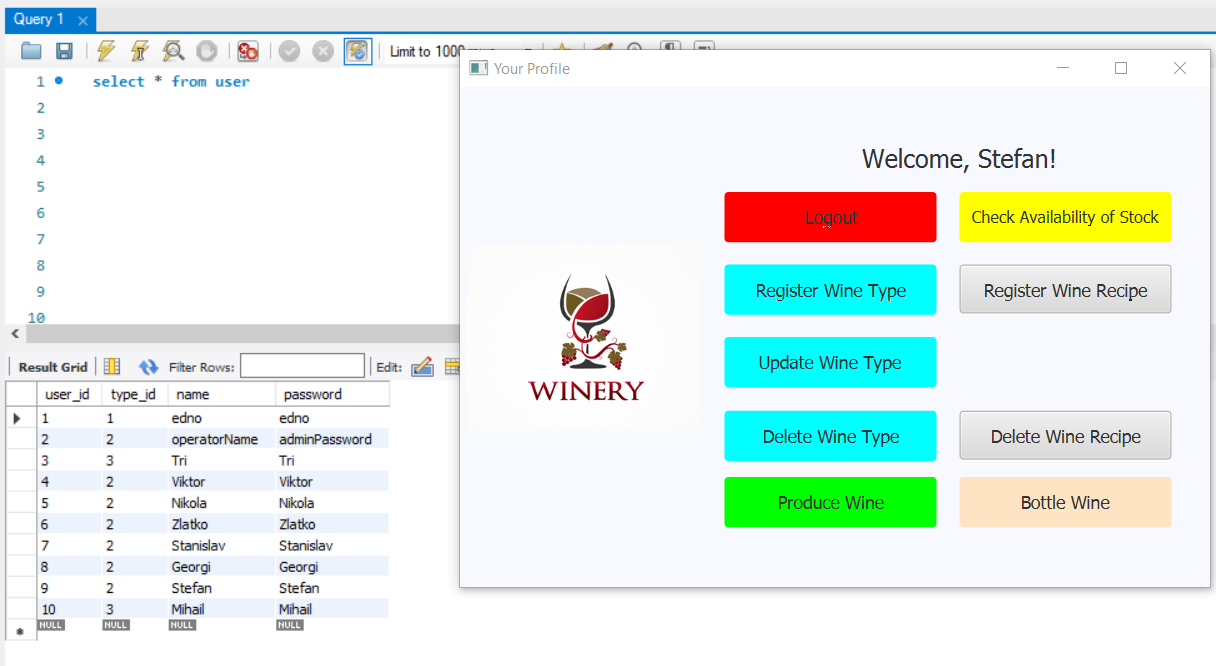
i. Функционални изисквания.

Операции за работа с потребители:

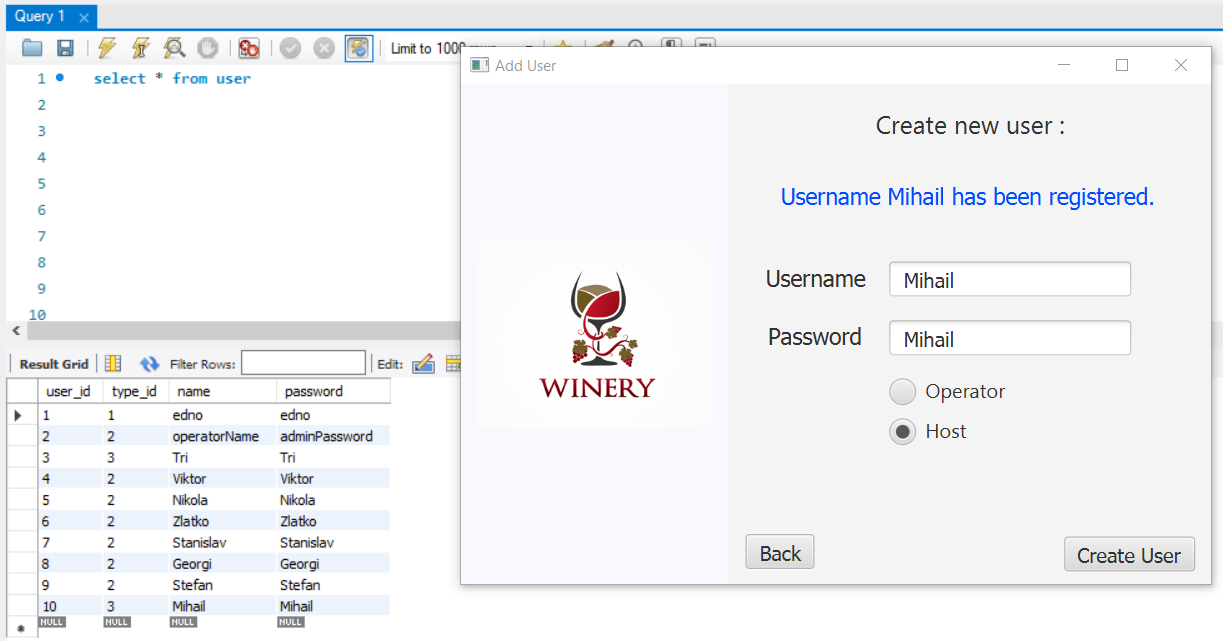


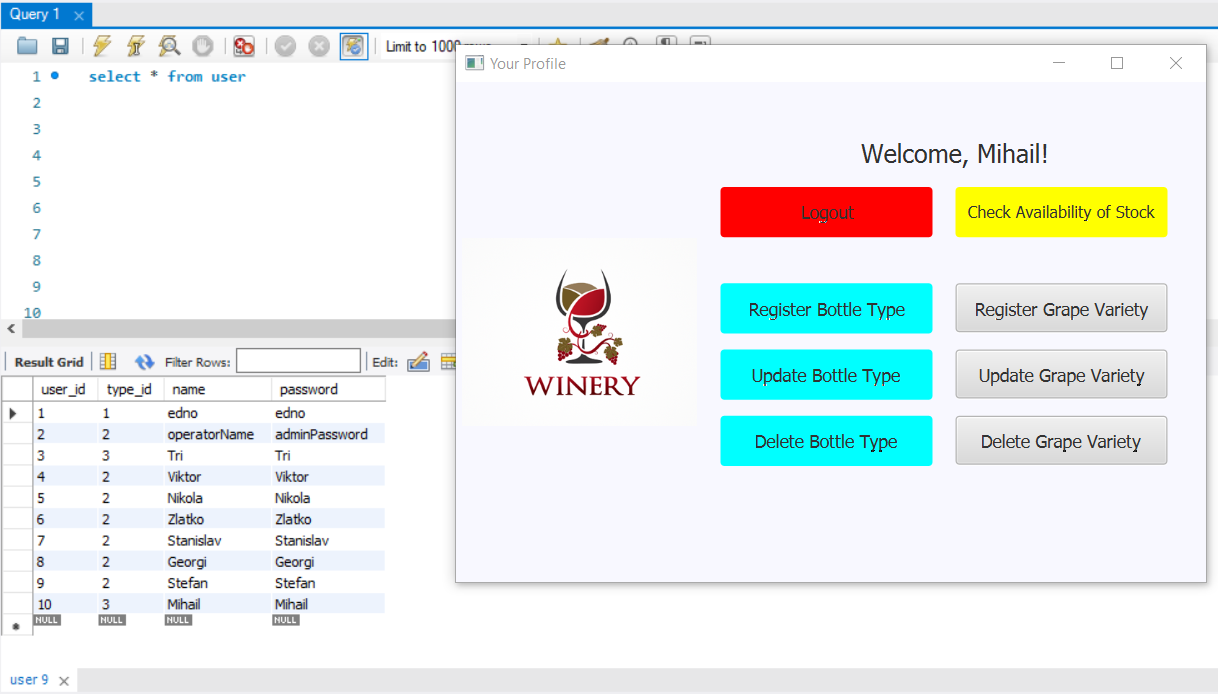
1.създаване на оператори от администратор;





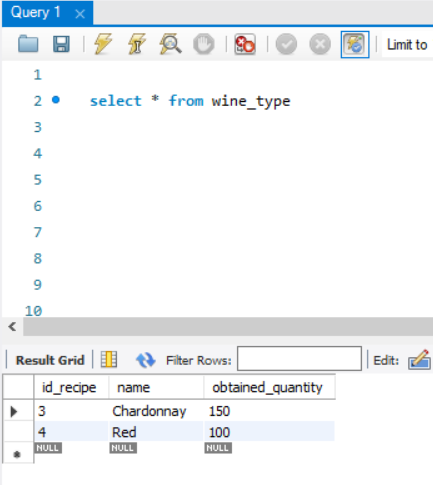
2.създаване на домакини от администратор;

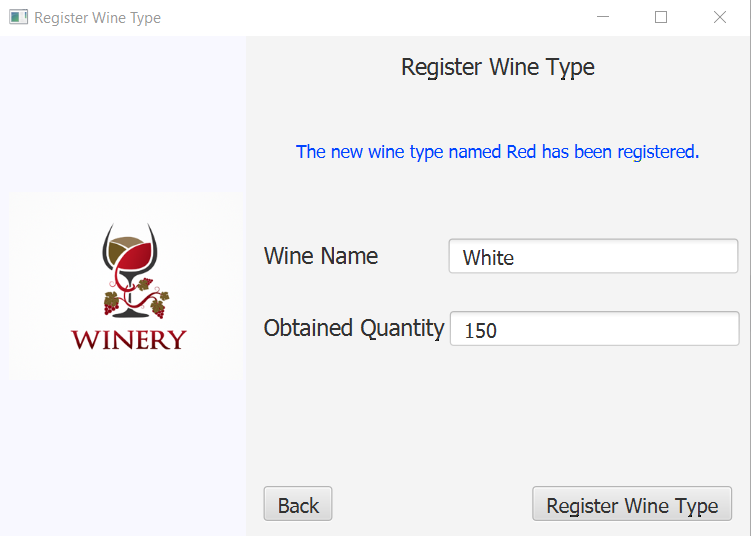


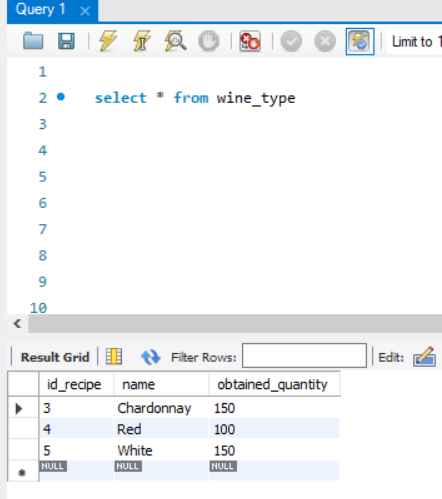


Системата поддържа операции за винопроизводство:

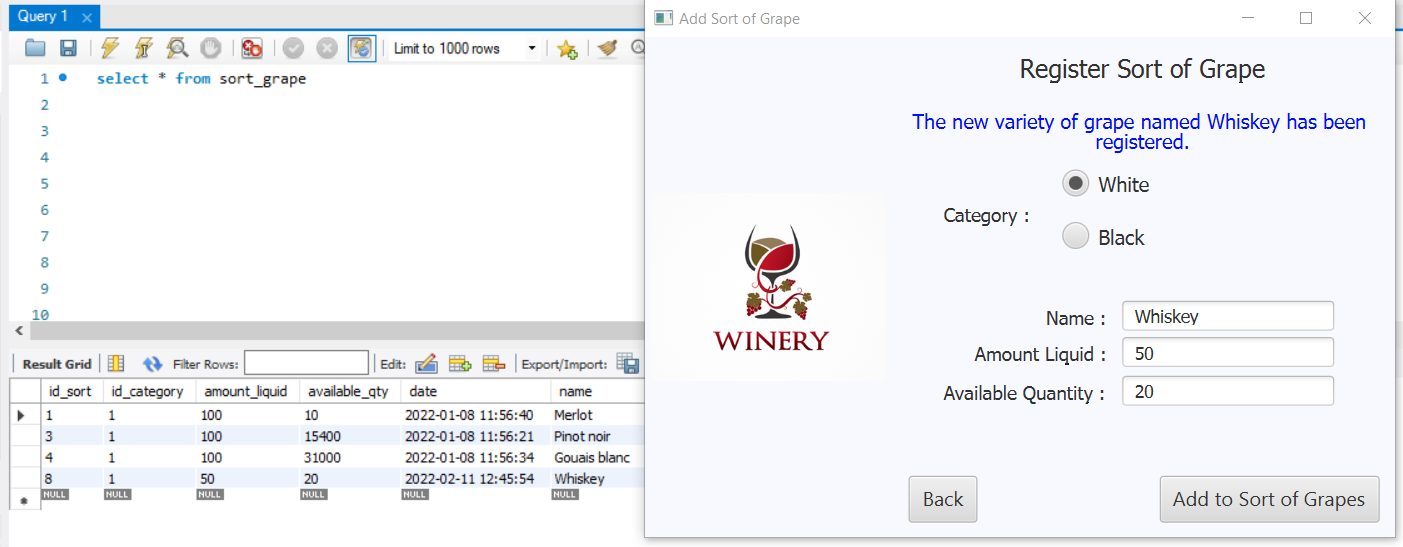
1.регистриране на сорт грозде и количеството му в склада(сортовете се делят на две категории - бели и черни);



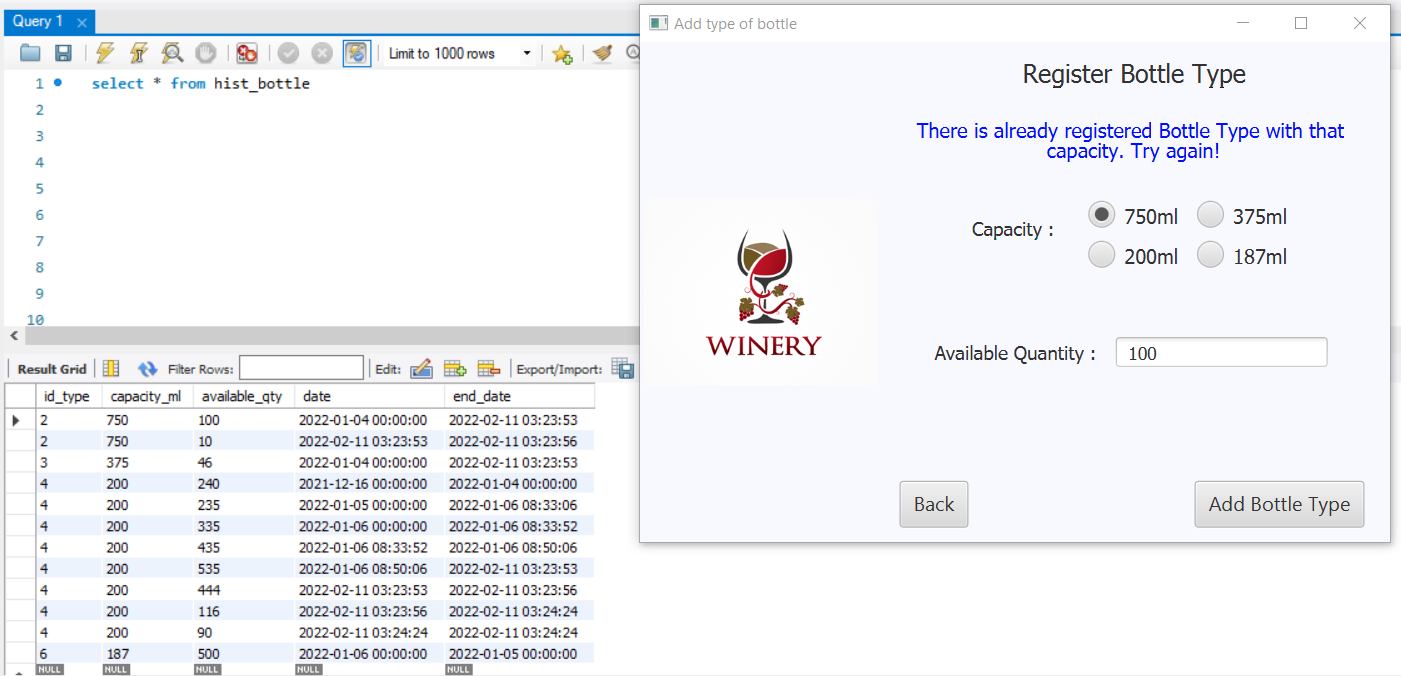




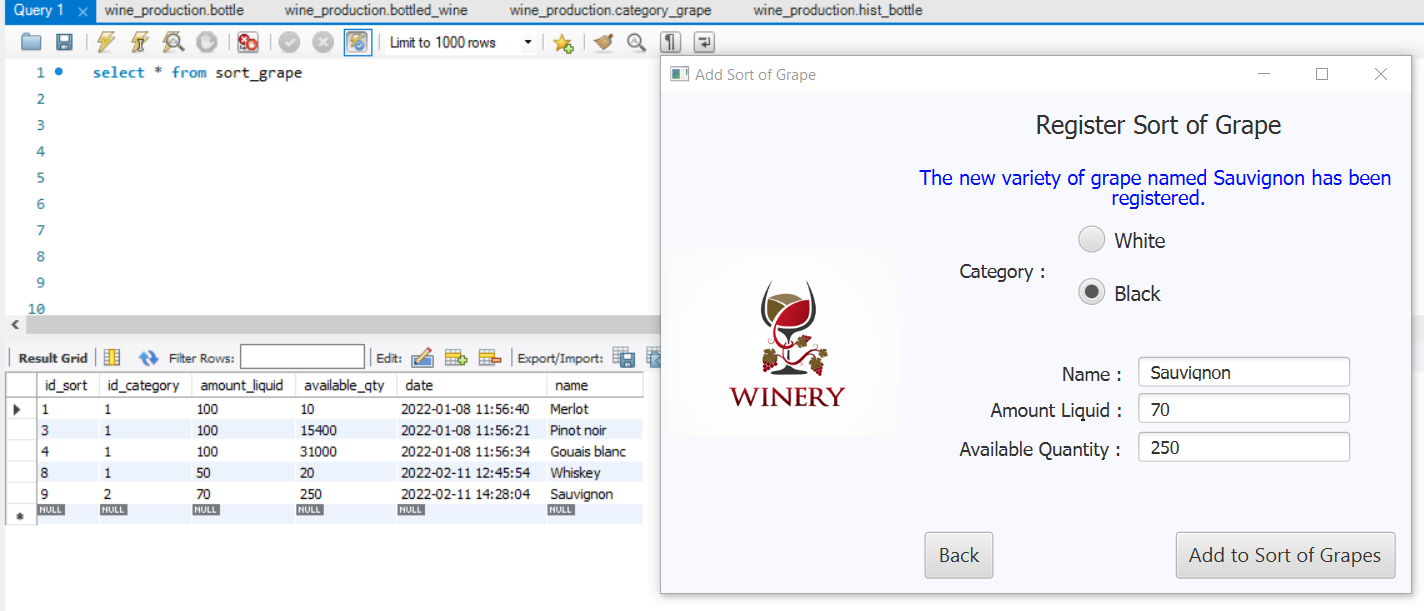
2.дефиниране на количеството вино което може да се получи от килограм грозде от даден сорт;



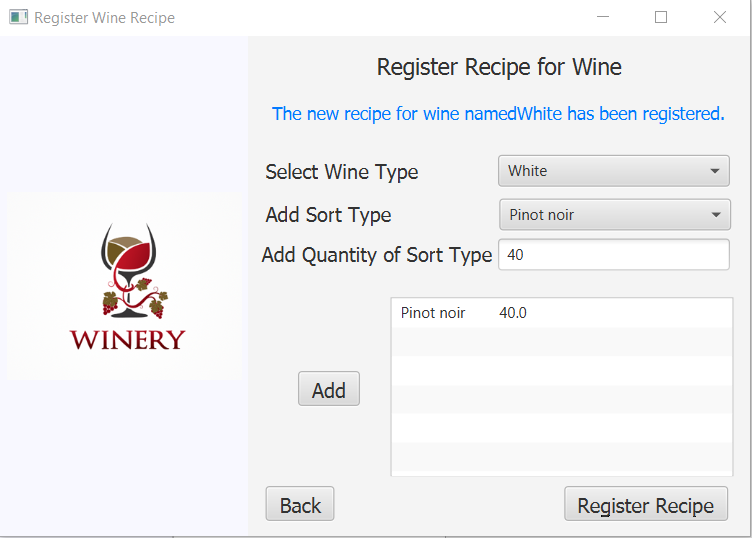
3.регистриране на видове и количество бутилки за бутилиране на вино(750, 375, 200 и 187ml);



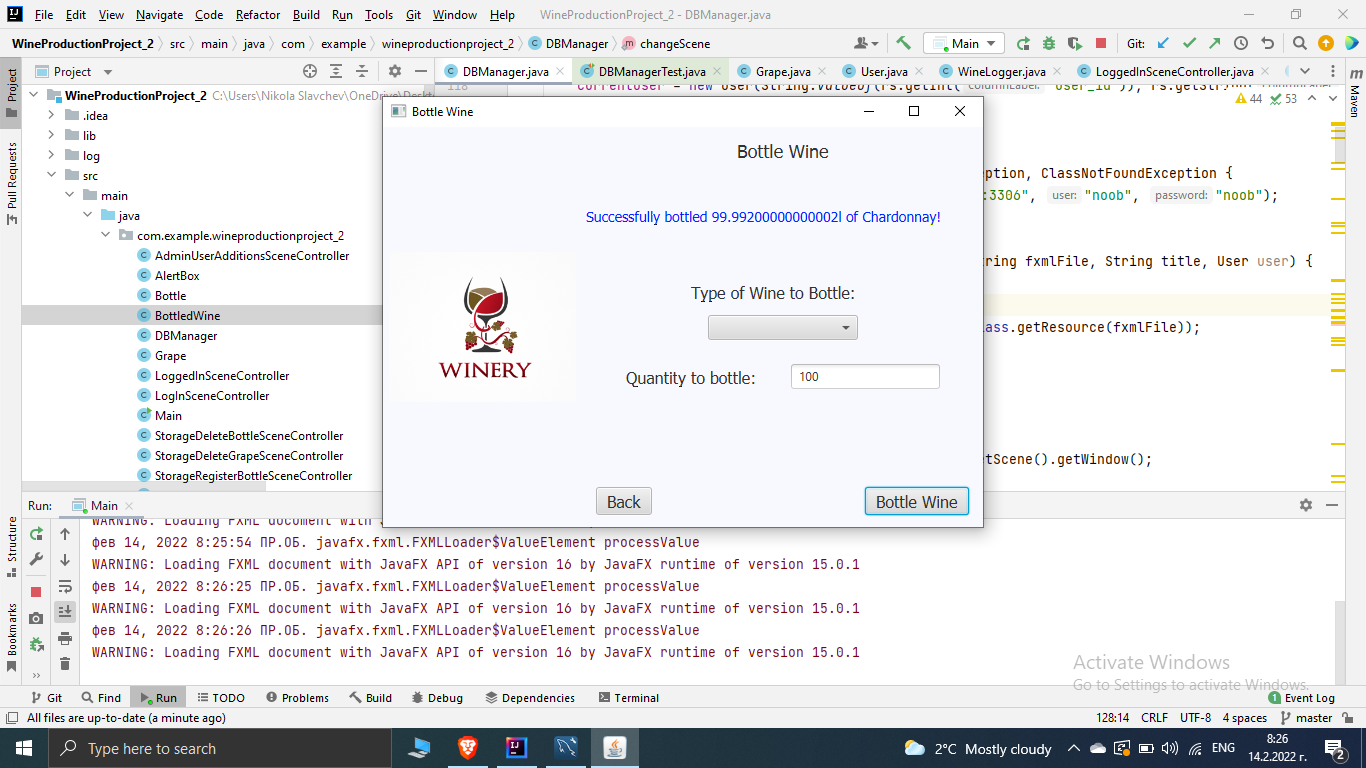
4.всеки тип вино се състои от един или няколко сорта и за направата му е необходимо определено количество от тях;



5.дефиниране на необходимите количества грозде за производството на даден вид вино;



6.бутилиране на виното в бутилки (бонус – системата автоматично да изчислява оптимално запълване на бутилки);

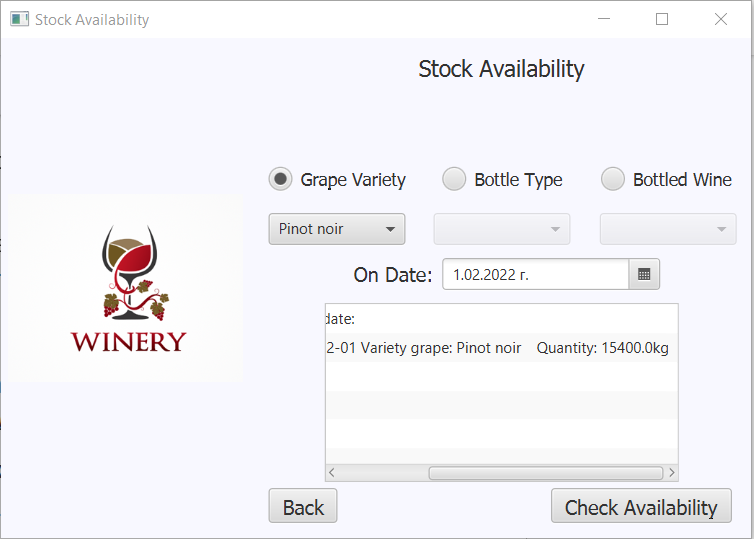




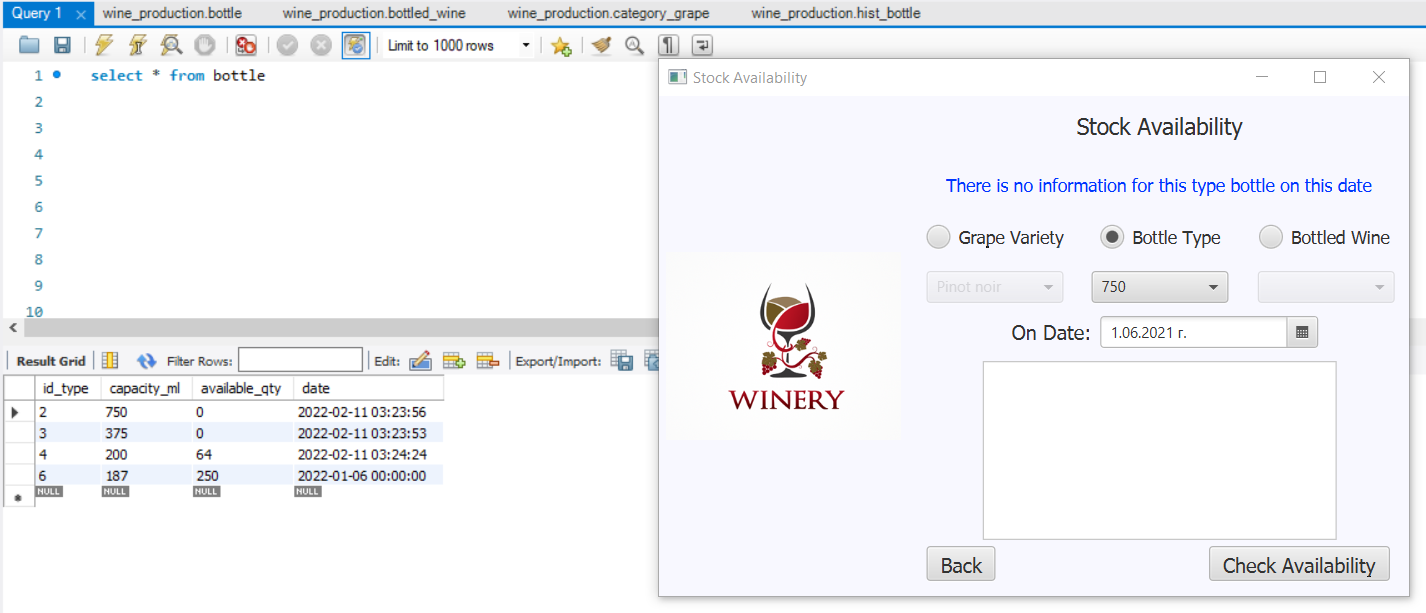


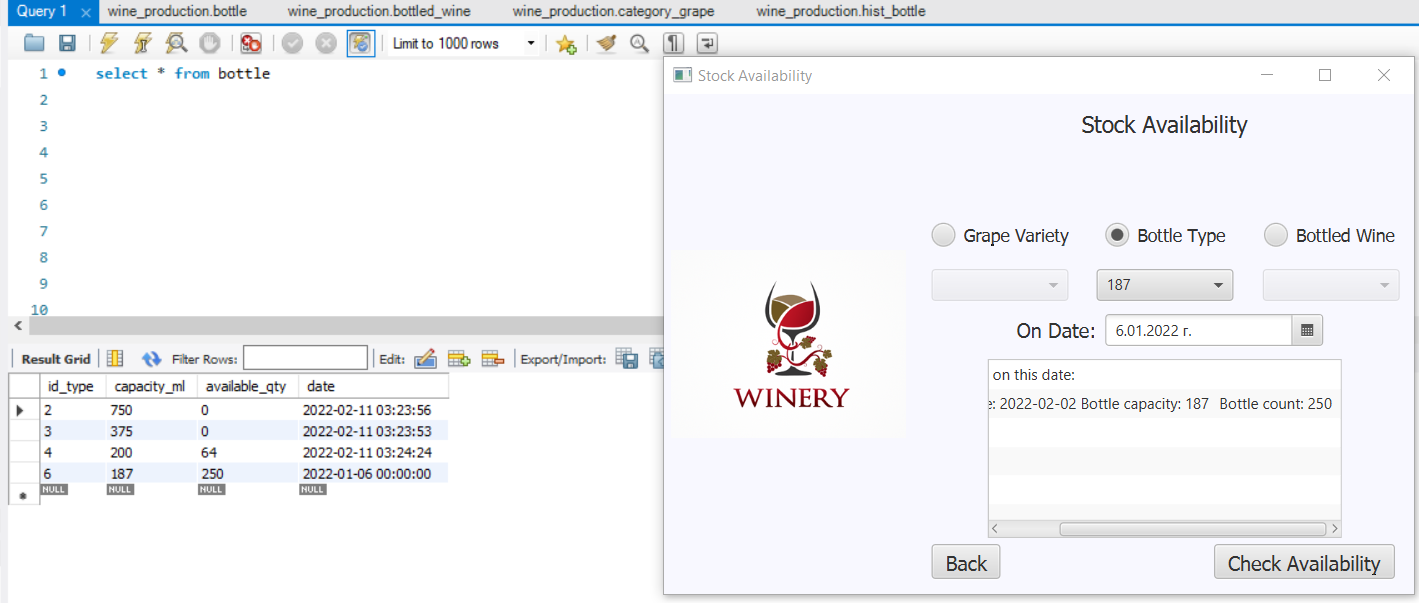
Системата поддържа Справки по произволен период за: 1. наличности на сортове грозде;



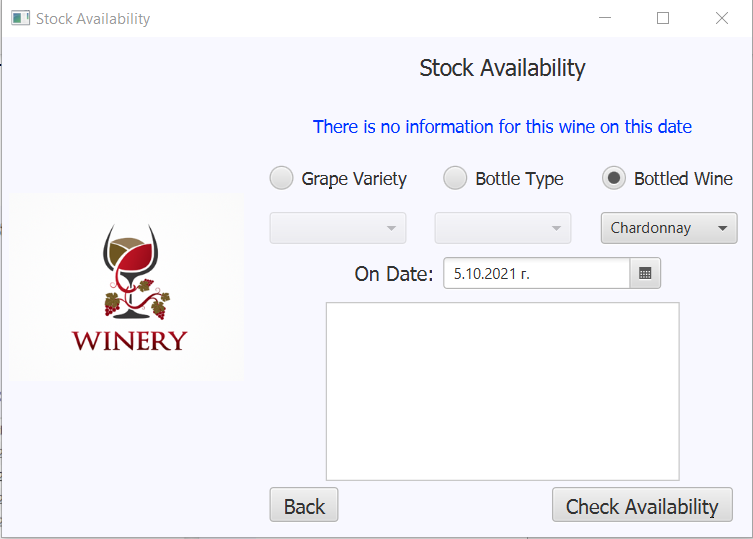


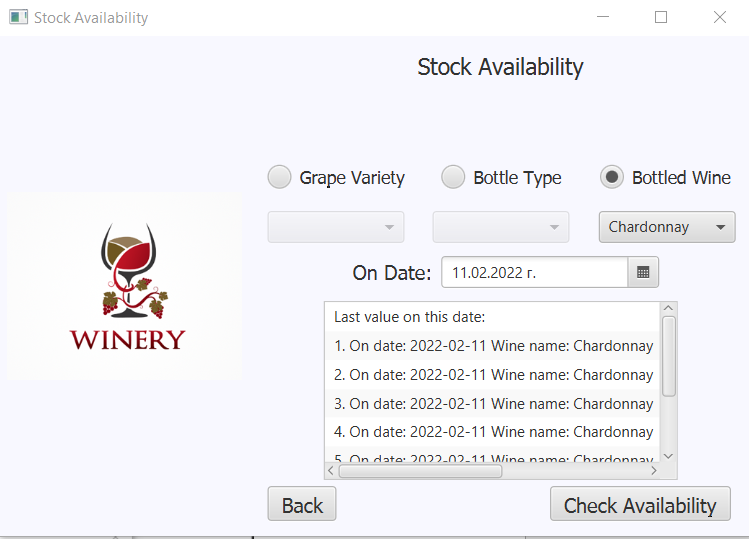
2. наличност на видове бутилки;





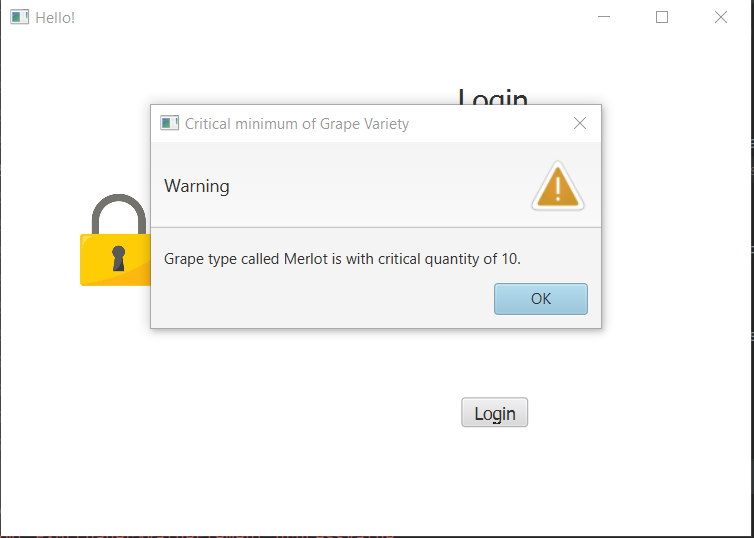
3. наличности на видовете бутилирани вина;



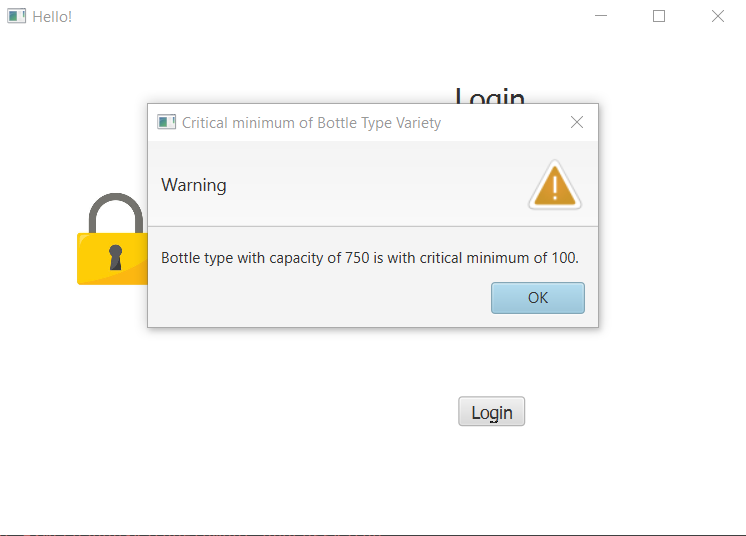


Системата поддържа Известия за:

1. критичен минимум и липса на даден сорт грозде;



1. критичен минимум и липса на даден вид бутилки;



ii. Структура на проекта.

iii. Дефиниция на модулите на системата.

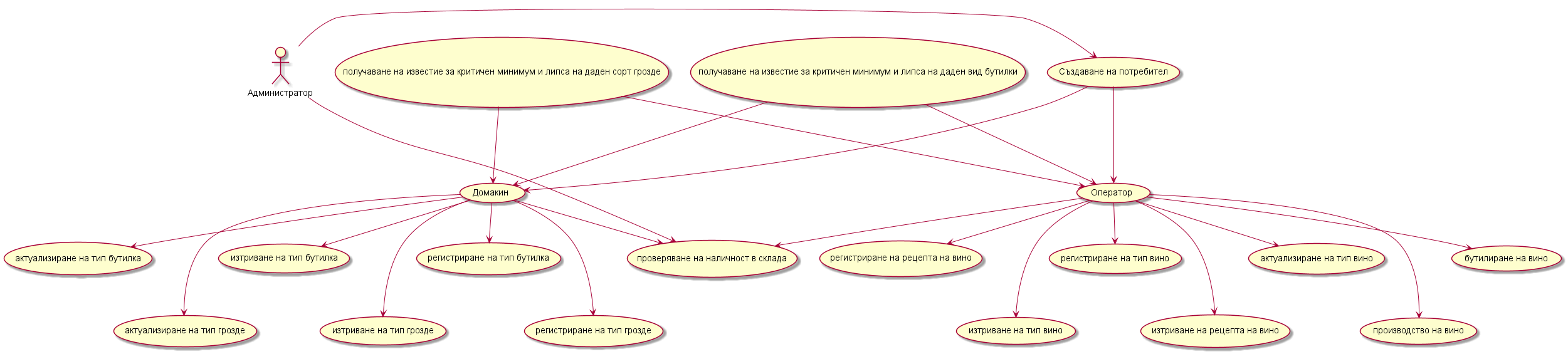
1. AdminUserAdditionsSceneController - Клас за създаване на нов user от Администратора.
2. AlertBox - Клас за предупреждение за критичен минимум.
3. BottledWine - Клас за въвеждане на бутилирано вино.
4. DBManager - Клас за връзка с базата данни и главната бизнес логика.
5. Grape - Клас за описване на характеристиките и количеството на вид грозде.
6. LoggedInSceneController - Клас за сцената на вече логнат user. Админ, оператор или домакин. Служи за навигация в приложението.
7. LogInSceneController - Клас за сцената на логин екрана, при съществуващ и несъществуващ user.
8. Main – main функцията на програмата
9. StorageDeleteBottleSceneController - Клас за изтриване на тип бутилка от домакин.
10. StorageDeleteGrapeSceneController - Клас за изтриване на вид грозде от домакин.
11. StorageRegisterBottleSceneController - Клас за регистриране на тип бутилка и количеството от този тип бутилка от домакин.
12. StorageRegisterGrapeSceneController - Клас за регистриране на категория грозде от домакин.
13. StorageUpdateBottleSceneController - Клас за актуализиране на количеството бутилки от даден тип от домакин.
14. StorageUpdateGrapeSceneController - Клас за актуализиране на наличното количество и налична течност грозде от домакин.
15. User - Клас за описване на характеристиките на user.
16. UserCheckStockAvailabilitySceneController - Клас за проверка на наличността в склада
17. WineLogger - Клас описващ процесите, случващи се с приложението.
18. WineOperatorBottleWineSceneController - Клас за сцена за бутилиране на тип вино и количество от оператор.
19. WineOperatorDeleteWineRecipeSceneController - Клас за изтриване на рецепта от тип вино от оператор.
20. WineOperatorDeleteWineTypeSceneController - Клас за изтриване на тип вино от оператор.
21. WineOperatorProduceWineSceneController - Клас за произвеждане на вино по рецепта и количеството от оператор.
22. WineOperatorRegisterWineRecipeSceneController - Клас за регистриране на рецепта по тип вино, тип сорт и наличност от сорта от оператор.
23. WineOperatorRegisterWineTypeSceneController - Клас за регистриране на тип вино по име и получено количество.
24. WineOperatorUpdateWineTypeSceneController - Клас за актуализиране на тип вино и полученото количество.

d. Проектиране на системата:

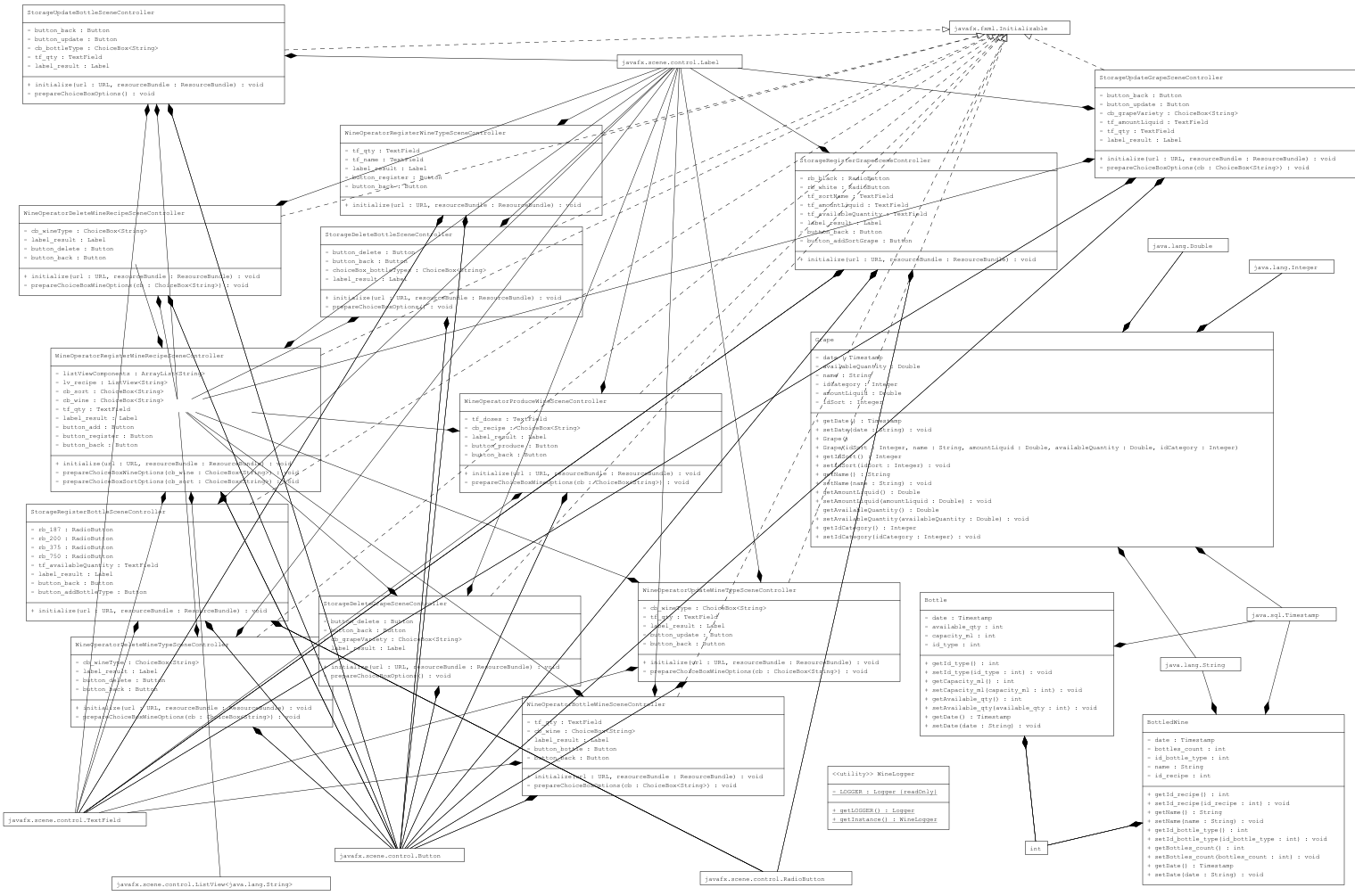
i. Проектиране на отделните модули (база данни, GUI, бизнес логика);

ii. UML –

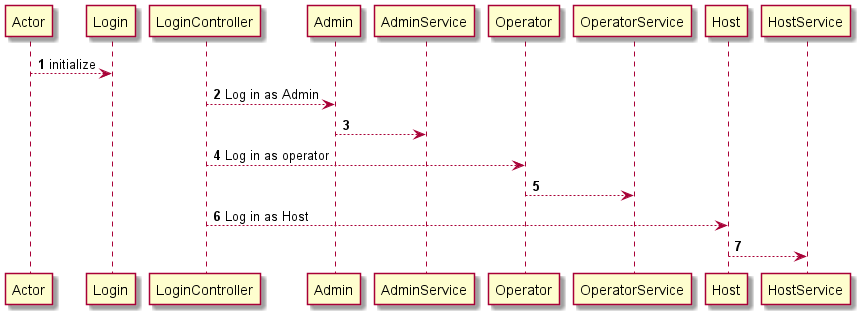
Use Case



Class Diagram



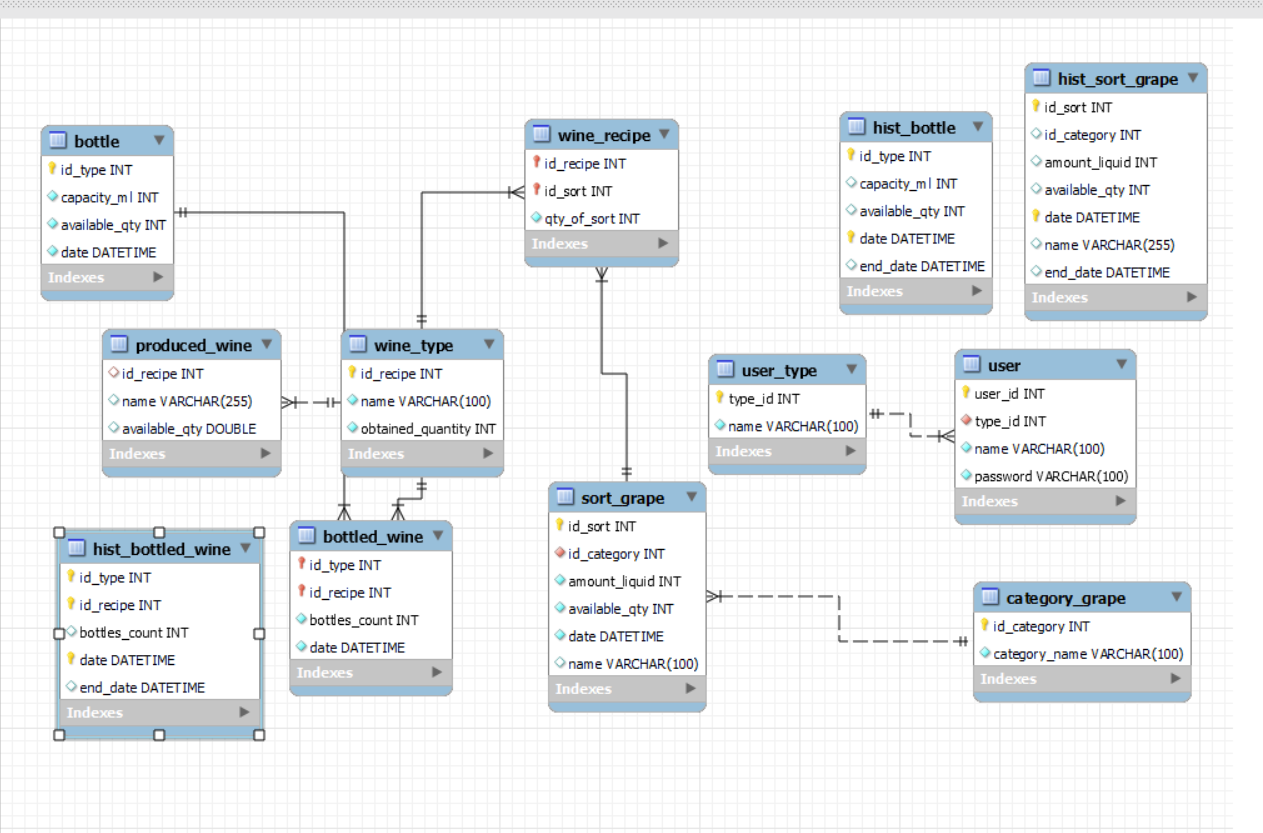
Sequence diagram за Login



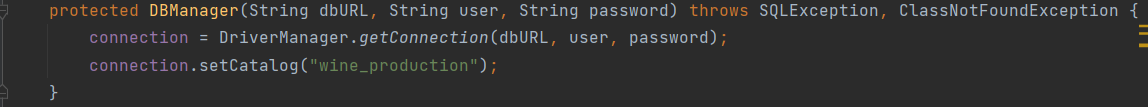
iii. Концептуален модел на базата от данни (ER диаграма (модел на Чен))

e. Реализация на системата:

i. Реализация на базата от данни (PostGres, Oracle, MS SQL, My SQL) – Релационна схема описание на таблиците

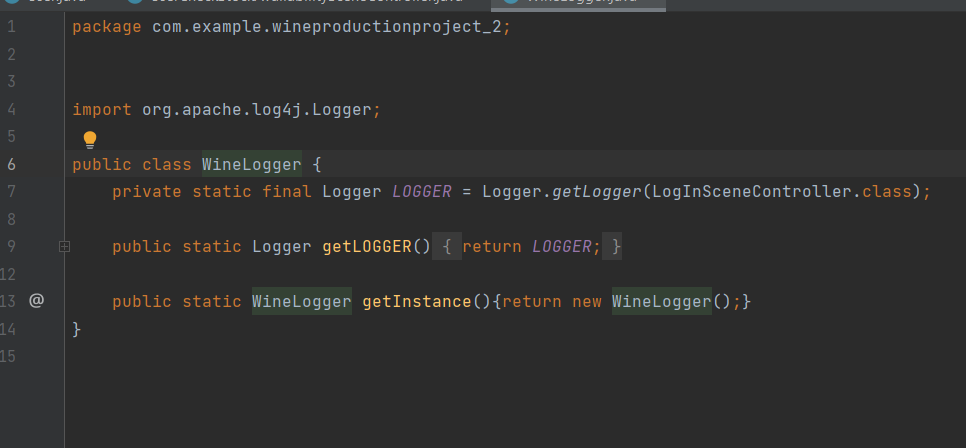


ii. Реализация на слоя за работа с базата данни (DAO, JDBC, JPA (Hibernate))



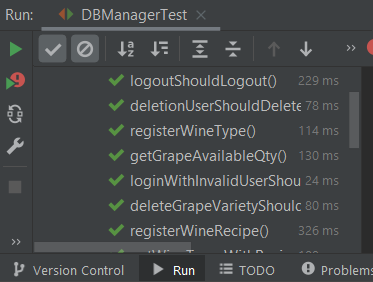
iii. Реализация на бизнеслогика и графичен интерфейс – Java, JavaFX

iv. Реализация на модул за регистриране на събития в системата – log4J



f. Тестови резултати:

i. JUnit tests



ii. Функционални тестове

iii. Интеграционни тестове