**Технически Университет – Варна**

Факултет по изчислителна техника и автоматизация

Катедра „Софтуерни и интернет технологии“

Специалност „Софтуерни и интернет технологии“

Курсов проект по Обектно-ориентирано програмиране – част II на тема

„Производство на вина“

Йордан Йорданов Добромир Начев

СИТ 2б СИТ 2б

18621692 18621700

**Условие**

**Производство на вина**

Да се разработи информационна система, предоставяща възможност за организиране на производството на различни типове вино.

Системата поддържа три вида потребители: администратор, оператори (винопроизводители) и домакин на склада с различни роли за достъп до функционалностите в системата.

Операции за работа с потребители:

* Създаване на оператори от администратор
* Създаване на домакини от администратор

Системата поддържа операции за винопроизводство:

* Регистриране на сорт грозде и количеството му в склада (сортовете се делят на две категории – бели и черни)
* Дефиниране на количеството вино, което може да се получи от килограм грозде от даден сорт
* Регистриране на видове и количество бутилки за бутилиране на вино (750, 375, 200 и 187ml)
* Всеки тип вино се състои от един или няколко сорта и за направата му е необходимо определено количество от тях
* Дефиниране на необходимите количества грозде за производството на даден вид вино
* Бутилиране на виното в бутилки (бонус – системата автоматично да изчислява оптимално запълване на бутилки)

Системата поддържа Справки по произволен период за:

* Наличности на сортове грозде
* Наличности на видове бутилки
* Наличности на видовете бутилирани вина

Системата поддържа Известия за:

* Критичен минимум и липса на даден сорт грозде
* Критичен минимум и липса на даден вид бутилки

**Анализ на проблема**

**Функционални изисквания**

Трябва да се създаде информационна система, която да предоставя възможност за производство на вино. Системата трябва да поддържа потребители с различни роли и да съхранява данни за грозде, бутилки, вина, както и да позволява бутилиране на вино.

Потребител от тип Администратор трябва да има възможност да:

* Създава потребители от тип оператор
* Създава потребители от тип домакин

Потребител от тип Оператор трябва да може да:

* Регистрира вид грозде и вид бутилки
* Дефинира кг. грозде за количество вино
* Определя необходимите сортове грозде за направата на вид вино
* Бутилира виното
* Прави справки за наличност на грозде, бутилки и бутилирано вино

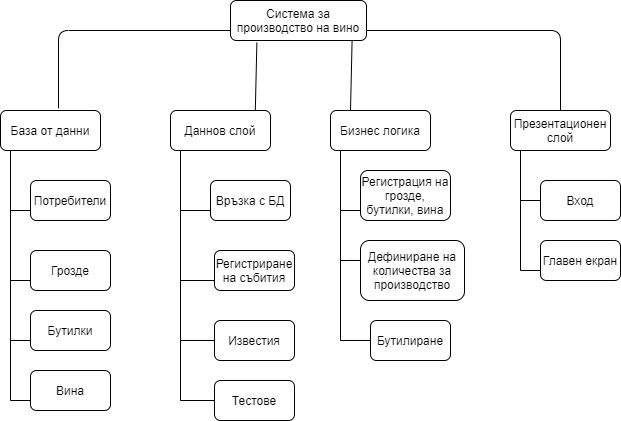
Потребител от тип Домакин трябва да може да:

* Регистрира наличност в склад на грозде, бутилки, бутилирано вино
* Прави справки за наличност на грозде, бутилки и бутилирано вино

Системата трябва да предоставя:

* Справки за наличност на грозде, бутилки, бутилирано вино
* Известия за критичен минимум и липса на сорт грозде и видове бутилки
* Графичен интерфейс за работа на потребители с функциите й

**Структура на проекта**

****

**Дефиниция на модулите в системата**

База от данни

* Oracle Database
* Таблици за съхранение на данни
* Процедури и тригери за промени в базата

Даннов слой

* DAO
* Hibernate
* Log4j
* JUnit
* Връзка с базата
* Известия
* Тестове

Бизнес логика

* Java
* Изчисления
* Програмен код

Презентационен слой

* JavaFX
* Графичен интерфейс за лесна работа на потребители

**Проектиране на системата**

**Проектиране на отделните модули**

База данни

Базата от данни трябва да:

* Съхранява данни за потребители
* Съхранява данни за грозде, бутилки, вино
* Прави нужни промени в данните при определени процеси

GUI

Графичният интерфейс трябва да:

* Има екран за влизане в системата
* Предоставя ясни и удобни за работа контроли
* Показва информацията от таблиците

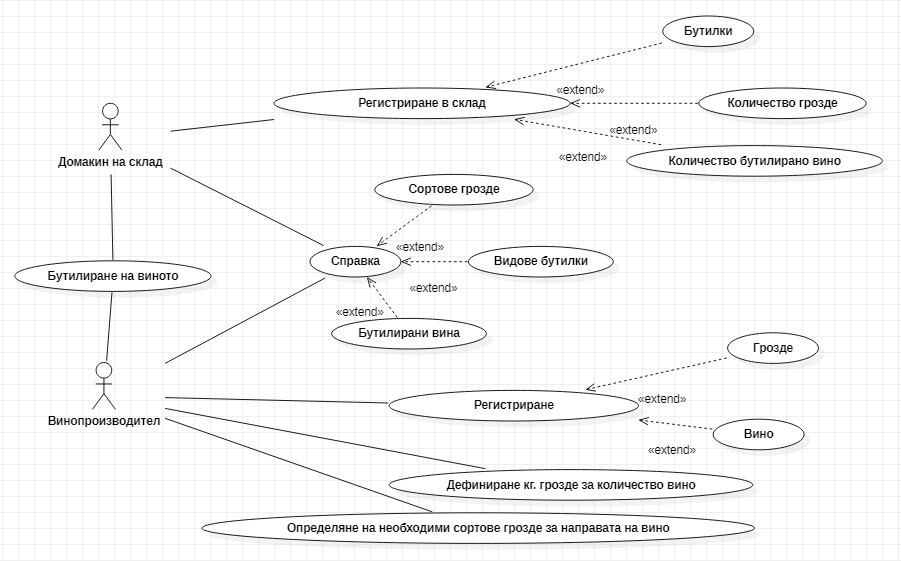
Бизнес логика

Системата трябва да предоставя възможност за:

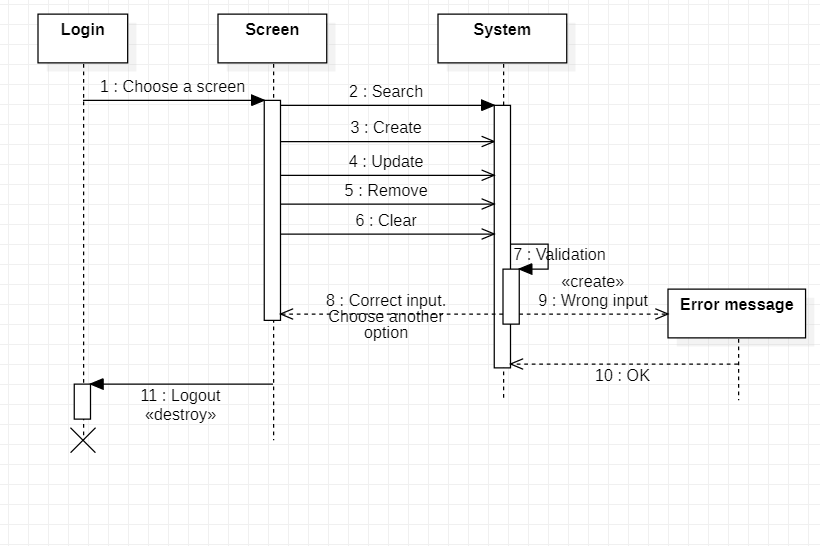
* регистрация на грозде, бутилки, вино
* бутилиране
* дефиниране на количество грозде, необходими за вино
* дефиниране на количество вино, което да се получи от кг. грозде

**UML**

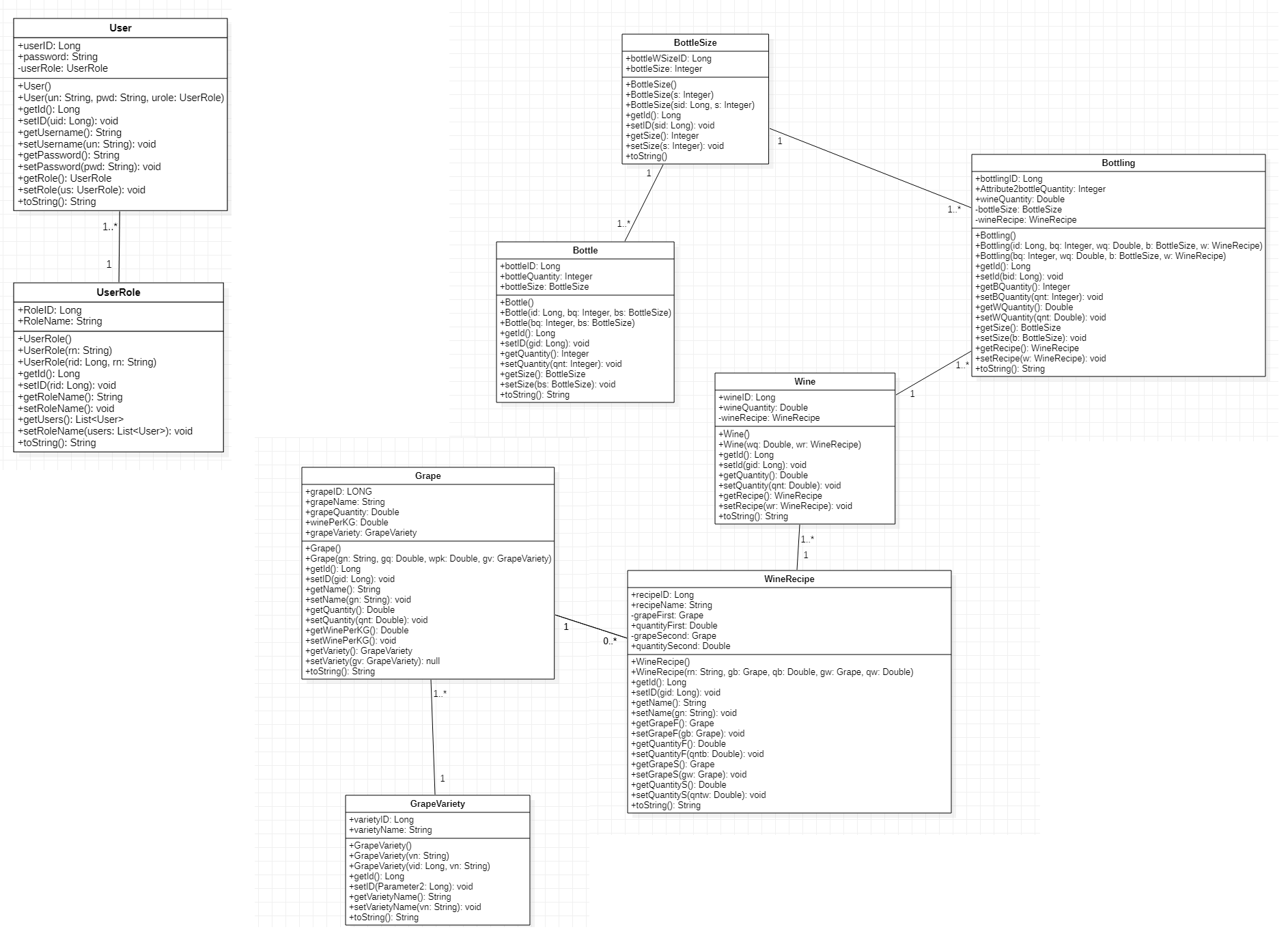
Use Case Diagram



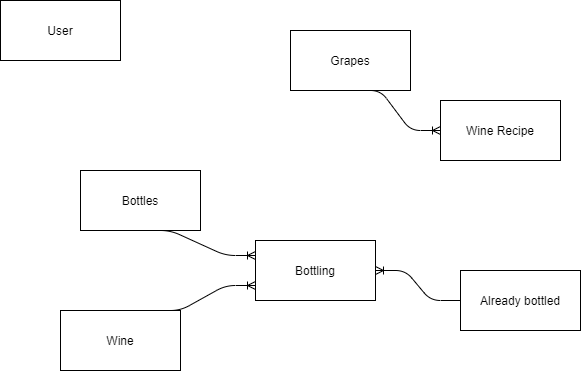
Activity Diagram



Class Diagram

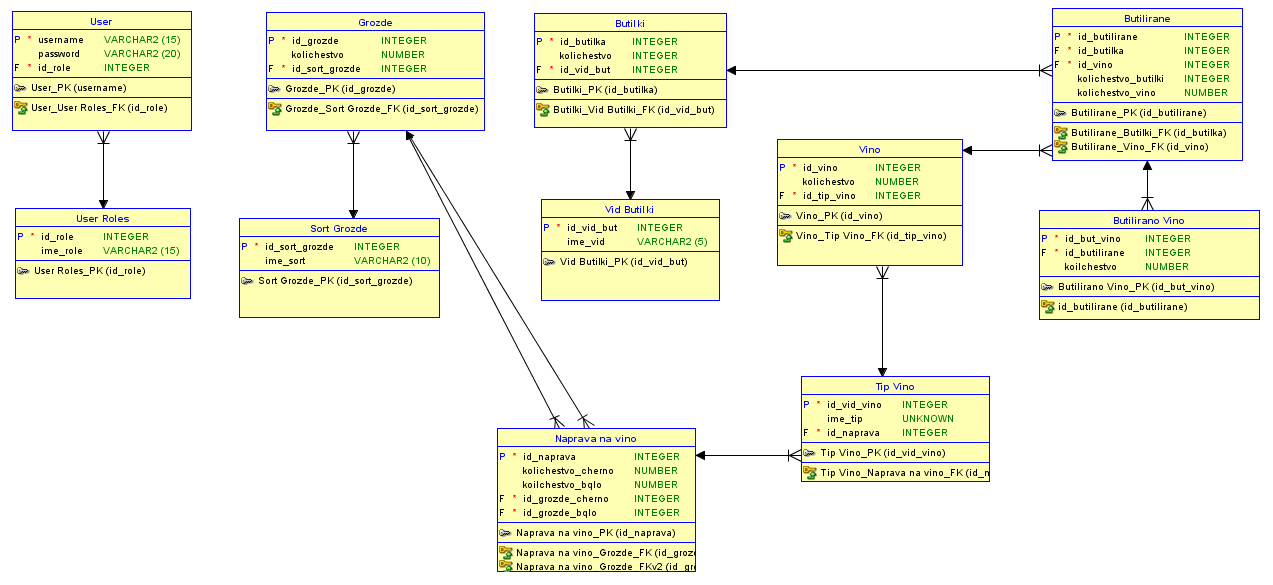
****

**Концептуален модел на базата от данни**

****

**Реализация на системата**

**Релационна схема**



**Слой за работа с БД**

Класовете:

* BottleDao
* BottleSizeDao
* BottlingDao
* GrapeDao
* GrapeVarietyDao
* UserDao
* UserRoleDao
* WineDao
* WineRecipeDao

Съдържат методи за опериране с базата от данни – създаване, обновяване и изтриване на данни.

**Бизнес логика и графичен интерфейс**

Бизнес логиката се състои от класовете:

* BottleService
* BottleSizeService
* BottlingService
* GrapeService
* GrapeVarietyService
* UserRoleService
* UserService
* WineRecipeService
* WineService

Които съдържат различни методи за осъществяването й.

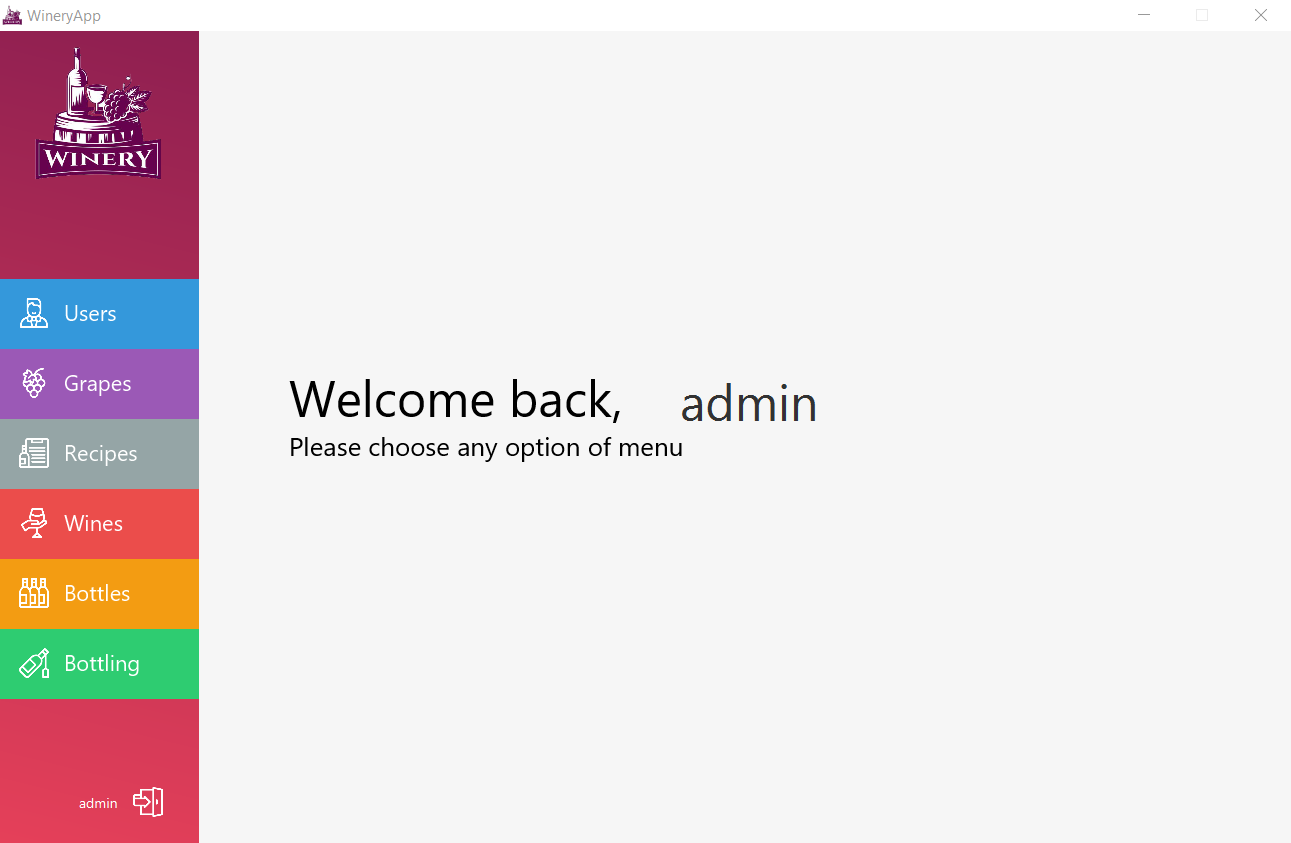
Графичният интерфейс се реализира от класовете:

* BottleController
* BottlingController
* GlobalController
* GrapeController
* LogController
* Main
* RecipeController
* UserController
* WineController
* Winemakercontroller
* wmBottleController
* wmBottlingController
* wmGrapeController

В които има методи за работа с JavaFX и реализация на различните екрани.

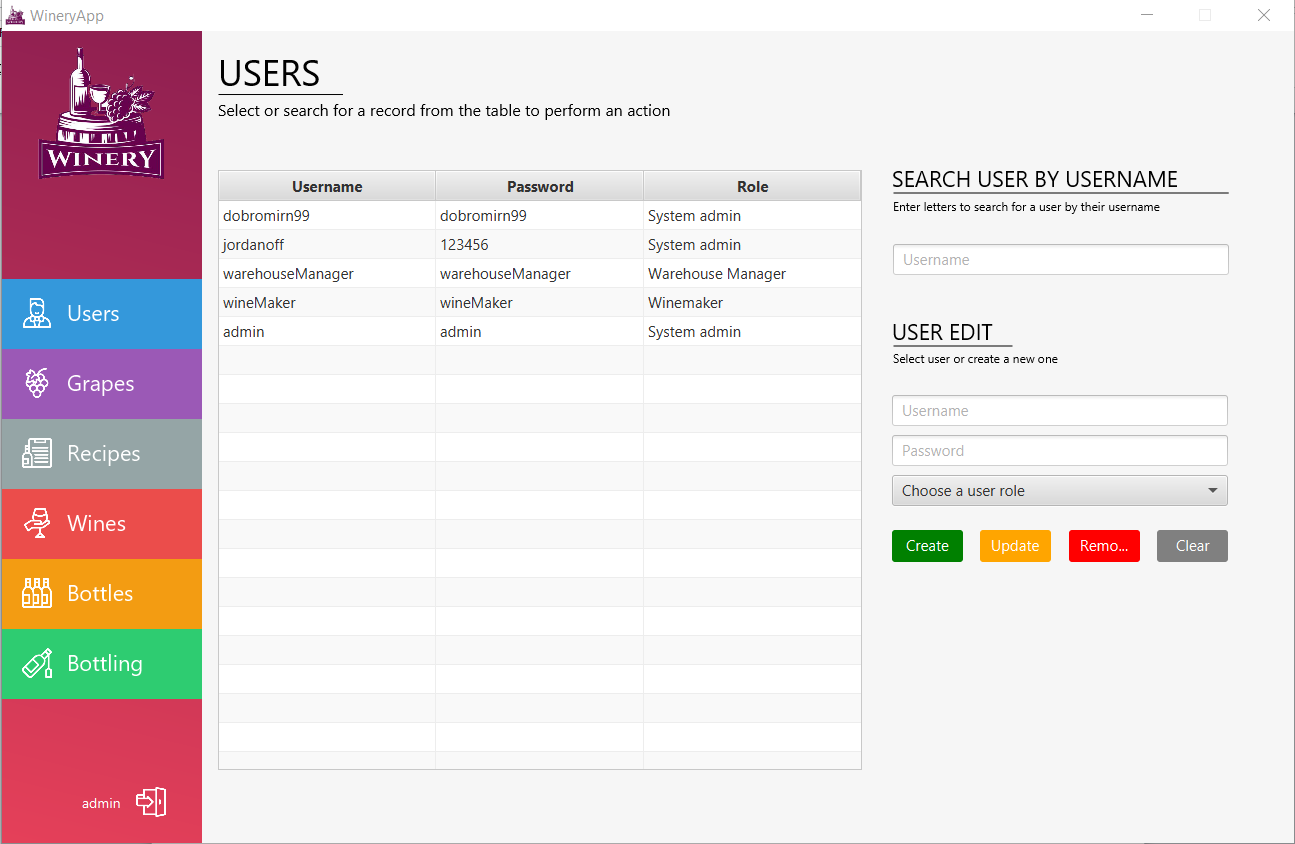
Екран за вход



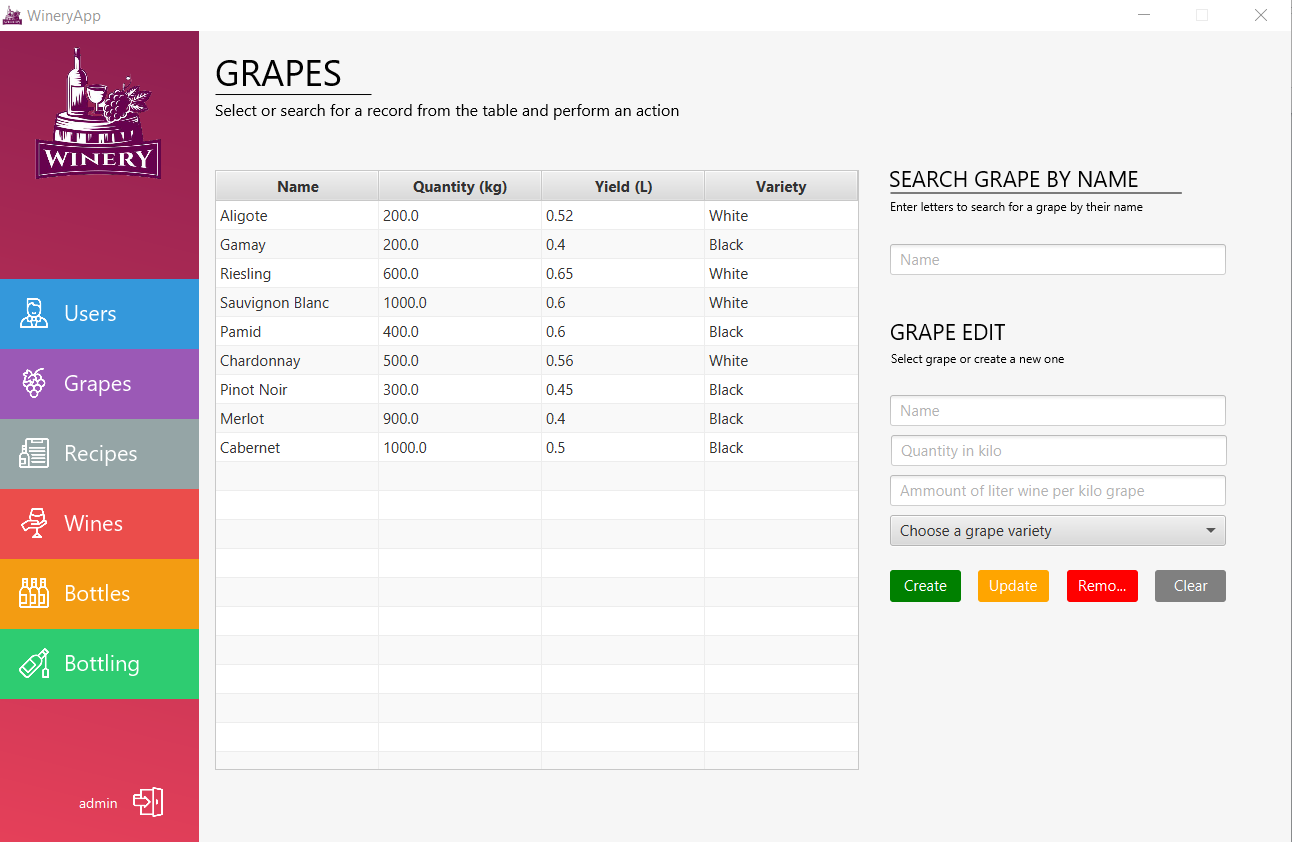
Начален екран

**Модул за регистриране на събития в системата**

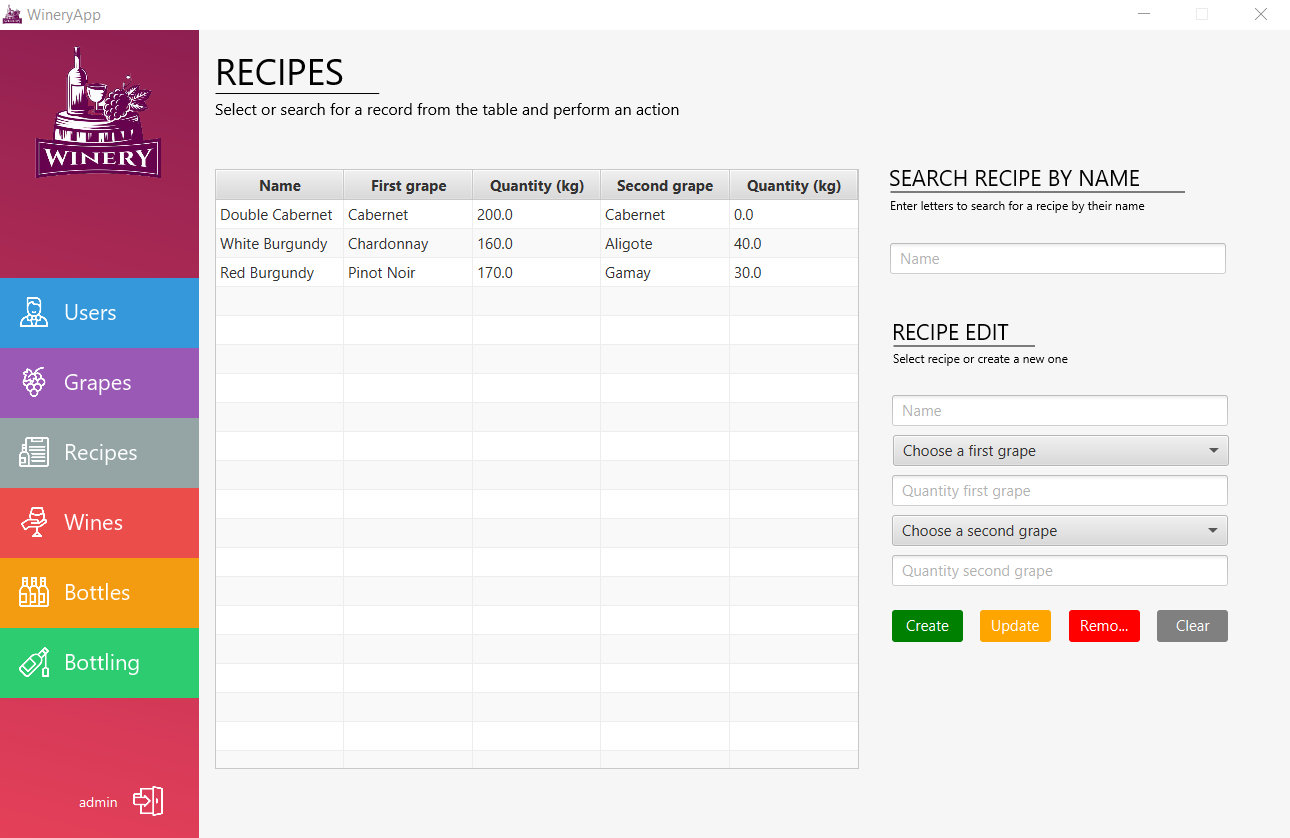
Екран с потребители, достъпен само до администратори



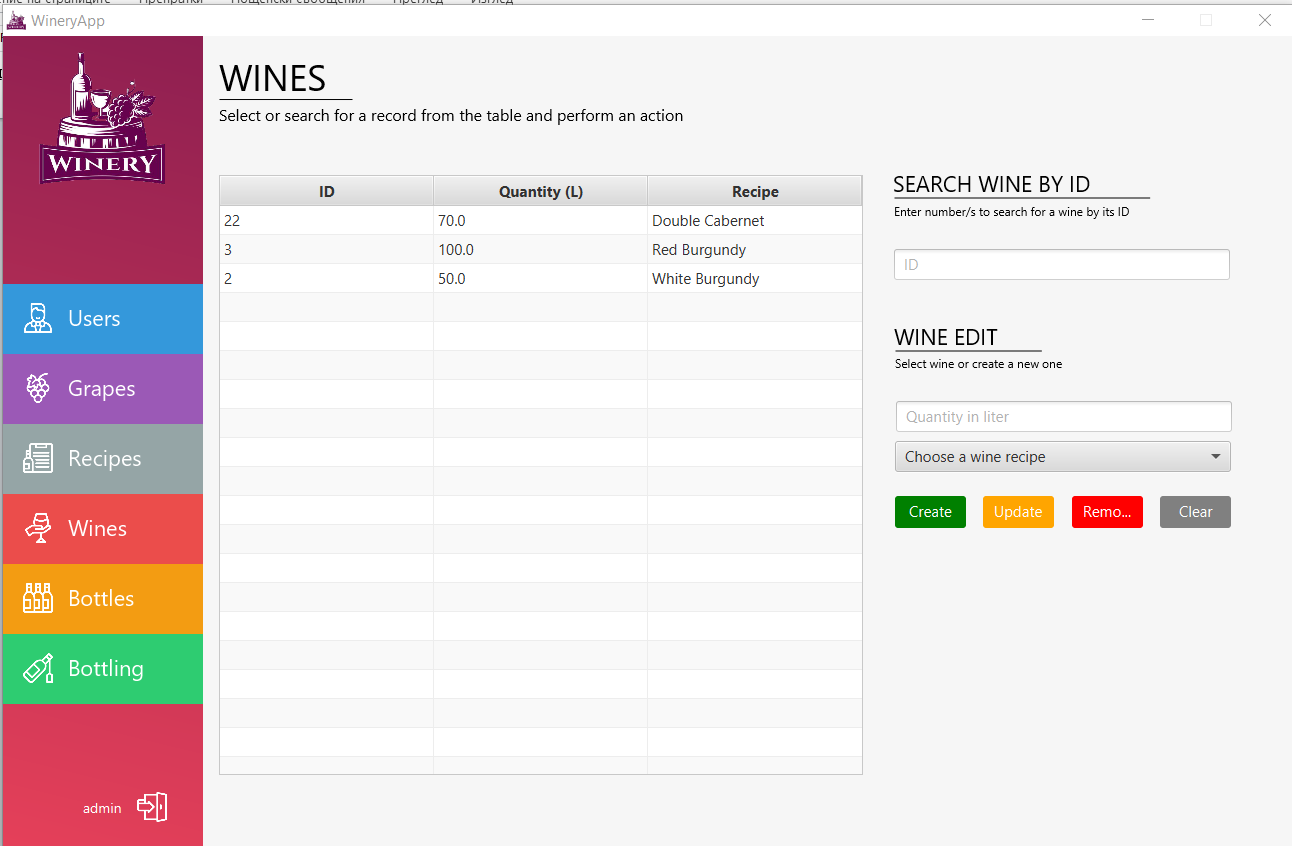
Екран с грозде, достъпен за всички роли



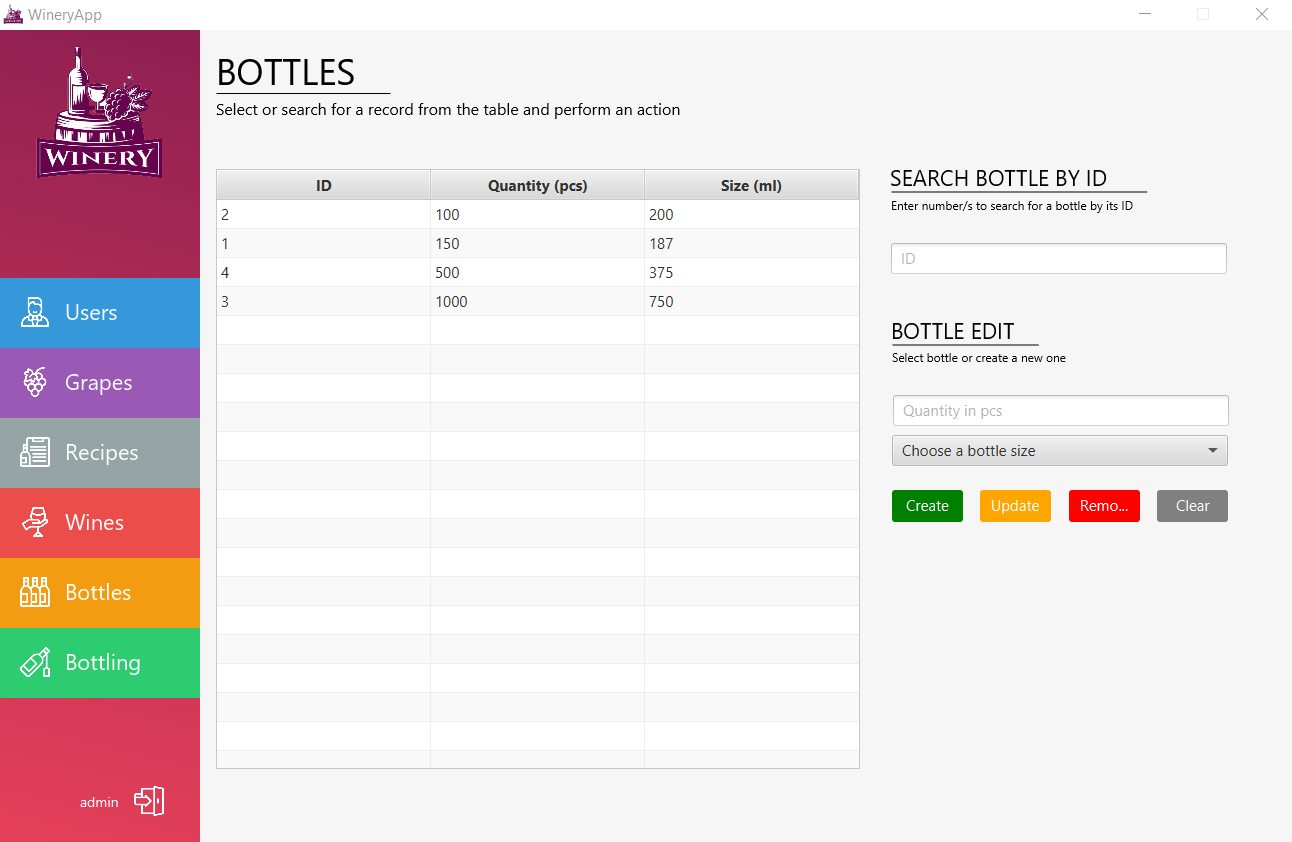
Екран с рецепти, недостъпен до домакини на склад



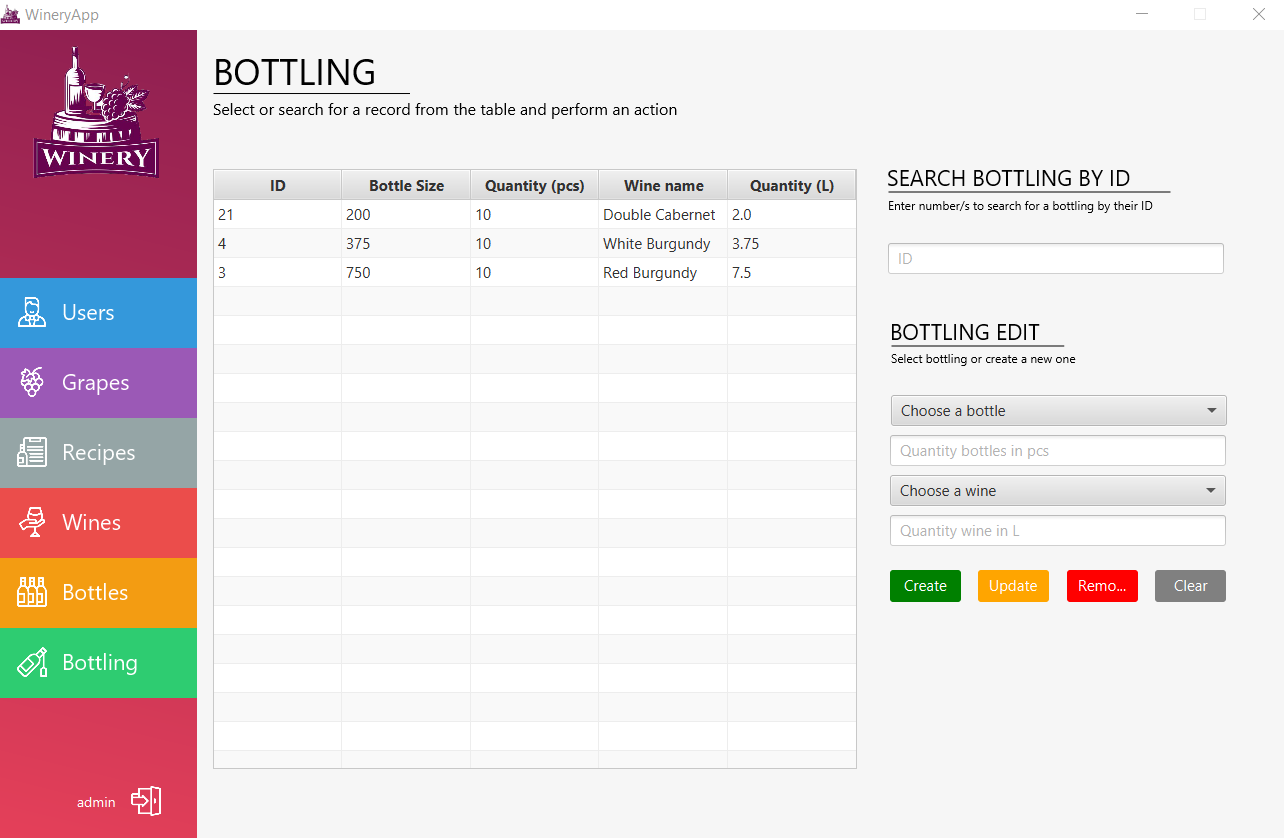
Екран с вина, недостъпен до домакини на склад



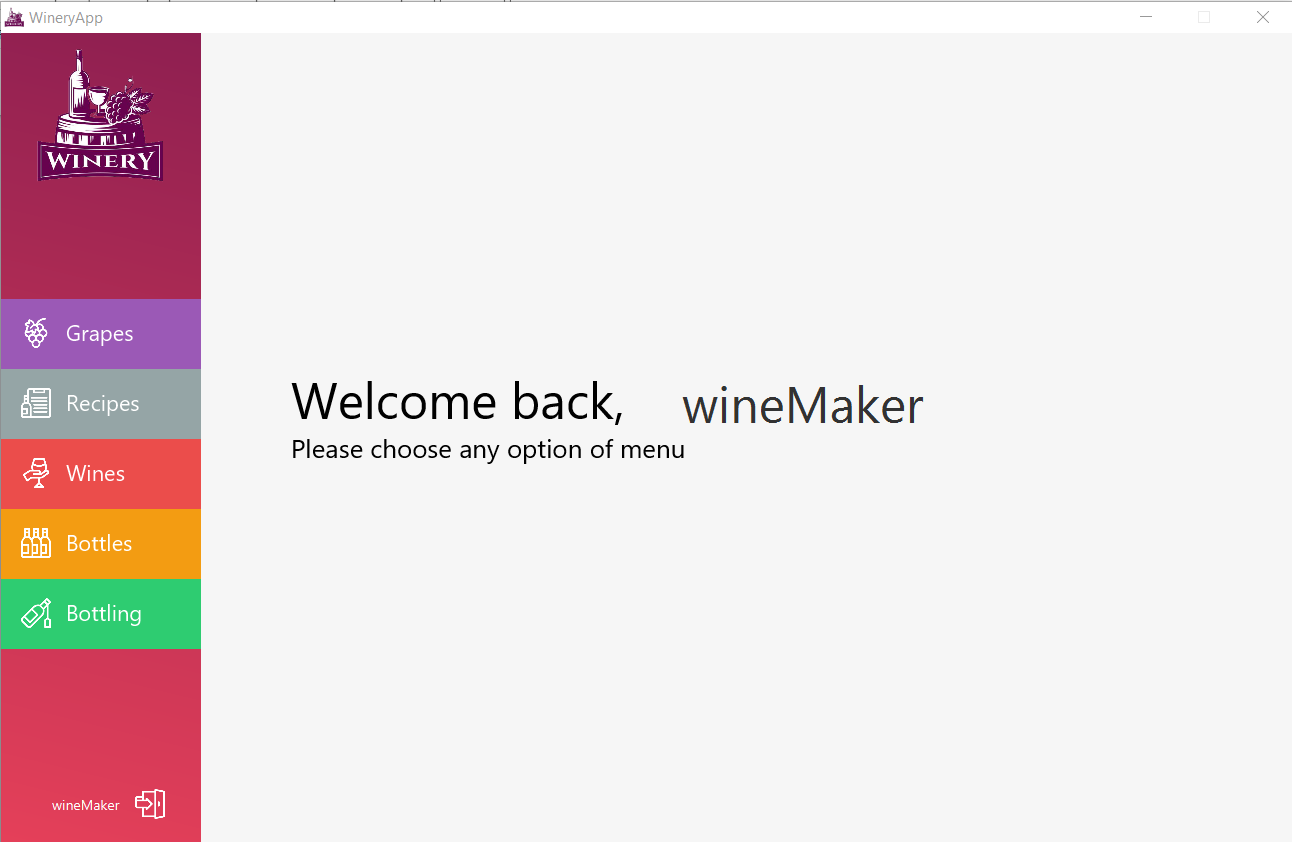
Екран с бутилки, достъпен за всички



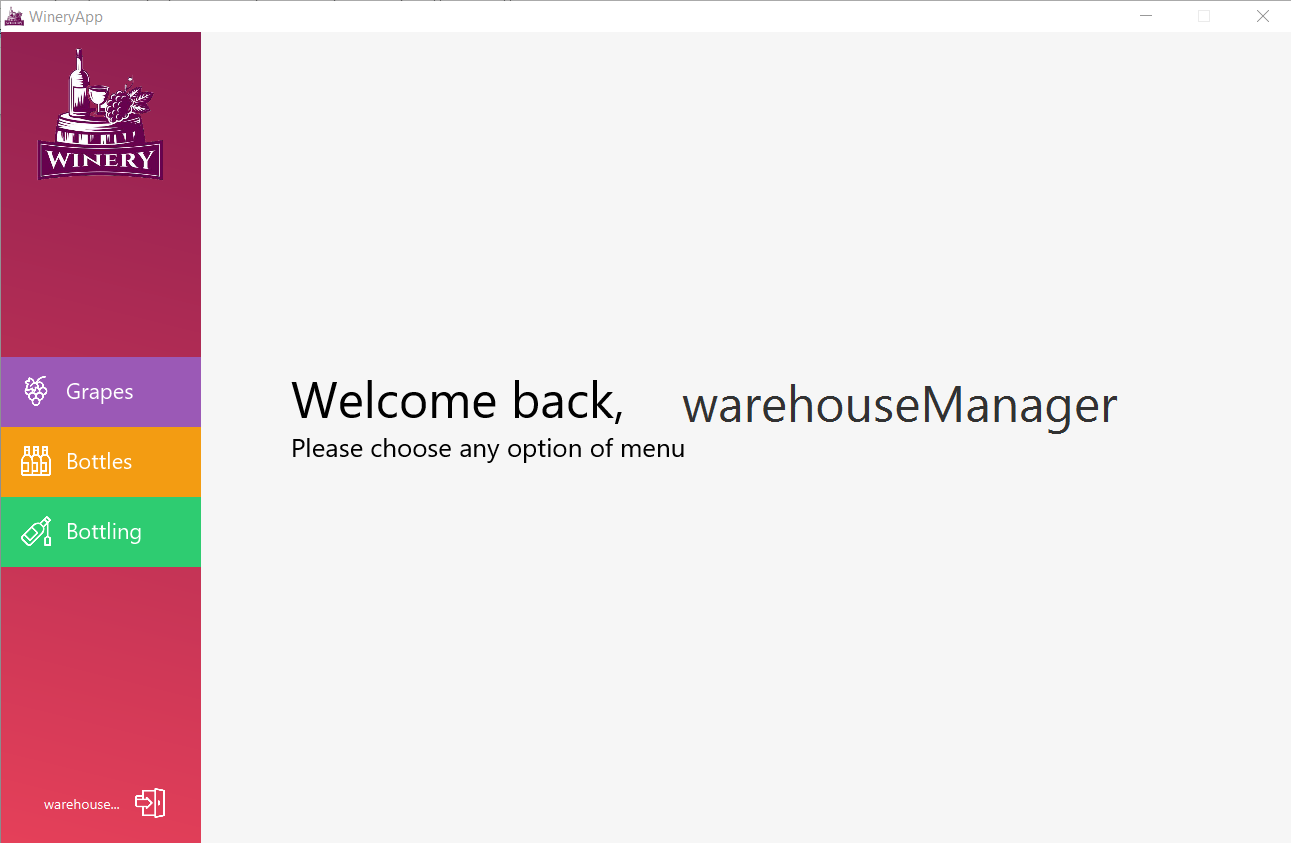
Екран за бутилиране, достъпен за всички



Перспектива на винопроизводител

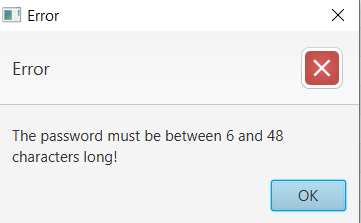
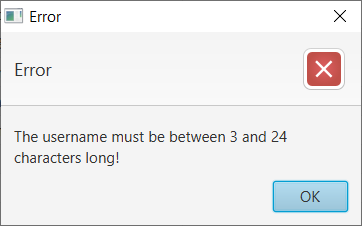


Перспектива на домакин на склад

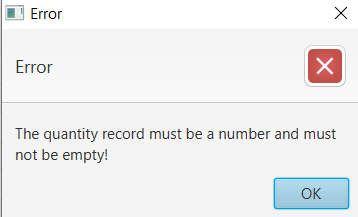
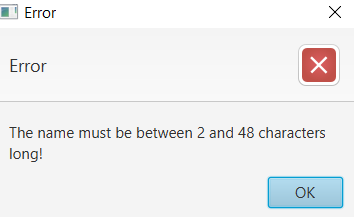


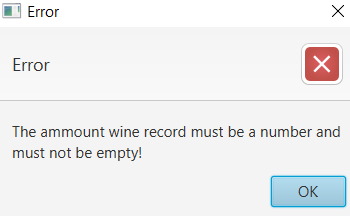
Съобщения за грешка при въвеждане на некоректни данни

За потребители

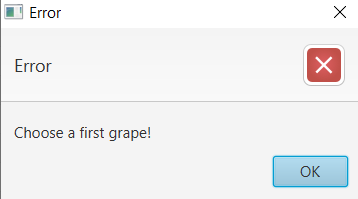
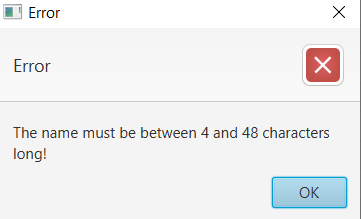


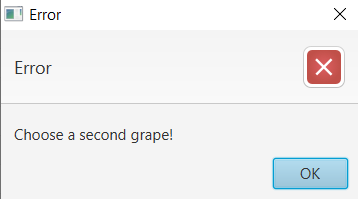
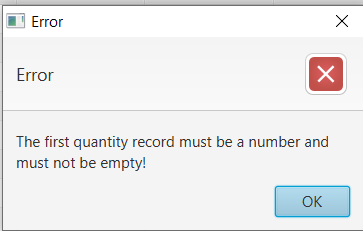
За грозде



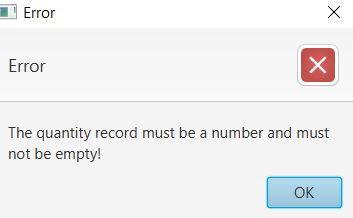


За рецепти

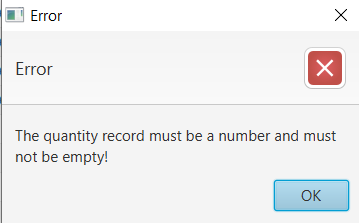




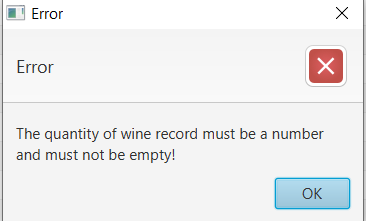
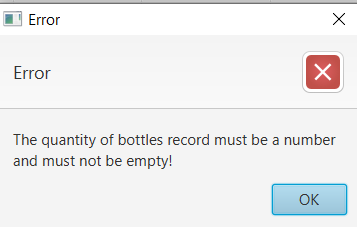
За вино

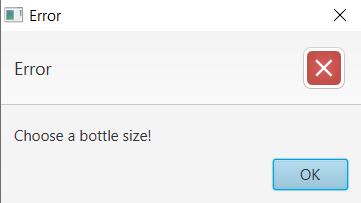


За бутилки



За бутилиране





Пример за въвеждане на нови данни

