Курсов проект

Факултет: Факултет по изчислителна техника и автоматизация

Специалност: СИТ

Тема на проекта: Производство на вина

Изготвили:

Петко Венелинов Борисов Ф.н:18621828

Никола Веселинов Колев Ф.н:18621828

Задание

**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 и домакин на склада с различни роли за достъп до функционалностите в системата.*

**Операции за работа с потребител:**

* *създаване на оператори от администратор;*
* *създаване на домакин от администратор;*

**Системата поддържа операции за винопроизводство:**

* *регистриране на сорт грозде и количеството му в склада (сортовете се делят на две категории - бели, червени;*
* *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(750,375,200,187);*
* *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 производство на даден вид вино;*
* *бутилиране на вино в бутилки (бонус - система автоматично да изчислява оптимално запълване на бутилки)*

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

* *наличности на сортове грозде;*
* *наличности на видове бутилки;*
* *наличности на видове бутилирани вина;*

**Системата поддържа Известия за:**

* *критични минимум и липса на даден сорт грозде;*
* *критичен минимум и липса на даден вид бутилки;*

Функционални изисквания

**Администратор:**

1.Създаване, редактиране, премахване на оператор.

2.Създаване, редактиране, премахване на винопроизводител- всеки винопроизводител произвежда различен тип вино , за който е нужно определено количество бяло и червено грозде(В случай че виното се произвежда само от 1сорт се въвежда ‚0‘ за количеството на другия).

3.Може да извършва операциите на оператор.

**Оператор:**

1.Информация за текущото количество в склада на всяка една стока.

2.Зареждане на определено количество стока.

3.История на поръчките.

4.Известия за критичен минимум на определена стока.

**Винопроизводител:**

1.Информация за стоката в склада, и какво количество от даден сорт е нужно за направата на 1л вино.

2.Пресмятане на максимално количество вино което може да се направи с текущата стока.

3.Произвеждане на вино – показва се отделен панел на екрана чрез който се избира броят на бутилките от определен вид които ще бъдат произведени.(Виното се съхранява само в бутилиран вид)+Функция за автоматично разпределяне между бутилките.

4.История на произведеното вино от текущия винопроизводител за период.

Структура на проекта

**1.**Проекта използва инструмента предоставен от средата “IntelliJ” чрез който могат да се добавят библиотеки от “maven”.

**2.JavaFX**-графичен интерфейс.

**3.Hibernate**-работа с база данни(записване, редактиране, извличане на данни)

**4.log4j**-система за регистриране на събитията в системата

**5.MySQL Connector**-връзка с „MySQL“ сървър.

**5.Junit** – отново предоставен от “IntelliJ” за тестове.

Дефиниция на модулите на системата

Файловете в проекта са разделени на 4 пакета Scenes, Dao ,Controllers и tables.

**1.Scenes**-това са всички „.fxml“ файлове който представляват графичния интерфейс на проекта.

-Тук също така се намира и класа WinNotifications , който реализира изскачащите графични нотификации.

-В този пакет се намират и иконите на приложението.

**2.Dao**-това са класове със функции за работа със съответна таблица(Entity) ,като записване, премахване или редактиране на данни в таблицата , както и различни справки.

-Тук също така се намира и класа HibernateUtil чрез който се конфигурира системата за връзка с базата данни.

**3.Controllers**-това са класове в който се реализира функционалността на различните елементи в системата(за всеки “.fxml” файл в проекта има съответстващ контролер).

**4.tables**-тук са класовете който съответстват на таблиците в базата данни(Hibernate Entities).

Реализация на база данни

Базата данни е реализирана чрез **MySQL** и има следните таблици:

**Warehouse**- съдържа информация за склад(местоположение, име на склада, уникален номер на склада).Към момента приложението не поддържа повече от един склад който се генерира автоматично при първоначалното изпълнение на програмата.

**Users**-Тук се съдържа информация за потребител(потребителско име, парола, уникален номер на привилегия, уникален номер на потребител, уникален номер на склад).

**PrivilegeType**-Тук се съдържа информация за видовете привилегии(име на привилегия, уникален номер на привилегия).

**Wine**-Тук се съдържа информация за готовото бутилирано вино(брой бутилки, уникален номер на вид вино, уникален номер на записа, уникален номер на склад, дата на производство, размер на бутилка, сериен номер).Когато се записва готовото вино за отделните партиди(които са с различна големина на бутилката) се генерира един общ сериен номер.

**WineType**-Тук се съдържа информация за видовете вино(име на вид вино, уникален номер на вид вино, нужно количество червено грозде за 1л вино, нужно количество бяло грозде за 1л вино, уникален номер на потребител(винопроизводител)).

**Grape**- Тук се съдържа информация за зарежданията на грозде(количество, уникален номер на вид грозде, уникален номер на записа, дата на зареждането, уникален номер на склад).

**GrapeType**-Тук се съдържа информация за вида грозде(име на вид, уникален номер на вид).

**Bottles**-Тук се съдържа информация за зарежданията на бутилки(уникален номер на зареждане, количество бутилки, уникален номер на вид бутилки, дата на зареждане, уникален номер на склад).

**BottleType**-Тук се съдържа информация за видовете бутилки(размер на бутилка,уникален номер на вид бутилка).

**CurrentStock**-Тук се съдържа информация за наличната стока като при промяна се актуализира таблицата(добавя се актуален запис)-(отделно поле за количеството на всички стоки, дата, уникален номер на склад, уникален номер на запис).

**Notifications**-Тук се съдържа информация за известията(уникален номер на известието, дата, съобщение, статус(ново или изтрито)).

Работата с базата данни се осъществява чрез Hibernate като операциите за всяка таблица се извършват посредством DAO за съответната таблица. В DAO класовете също така има различни справки нужни за работата на програмата като: максимален уникален номер, текущ потребител…

**Пример за въвеждане:**

1.Създава се обект от класа на съответната таблица в която искаме да въведем данни.

2.Попълват се всички член данни на класа.

3.Създава се обект DAO за съответната таблица.

4.Извиква се функция от обекта създаден в горната точка. Като параметър функцията приема обект от съответния клас.

5.След изпълнението на функцията записа ще бъде прехвърлен към базата данни ако всичко е попълнено вярно.

Реализация на бизнес логика и графичен интерфейс

Реализация на бизнес логика за администратор

1.*Създаване на оператор*- показва се нов панел върху главния който има 2 текстови полета съответно за име и парола , след натискане на бутон запис ако текстовите полета не са празни се създава оператора.

2.**Изтриване на оператор**- показва се нов панел върху главния със текстово поле за име на оператор. При натискане на бутон изтриване се проверява дали има оператор с такова име и се изтрива при валидна проверка.

3.**Създаване на винопроизводител**- Показва се нов панел върху главния с полета за име, парола, нужно бяло вино, нужно червено вино, име на вид вино. Със създаването на нов винопроизводител се създава и нов тип вино.

4.**Редактиране на винопроизводител**-избира се винопроизводител от списъка. При натискане на бутон редактиране се отваря нов панел върху главния с полета попълнени с информацията за избрания производител. При натискане на бутон редактиране

се редактира информацията ако някое поле е променено , ако не информацията остава същата.

5.**Преглеждане на стока** – при натискане на бутон преглеждане на стока се отваря панела на оператора и могат да се извършват всички негови операции.

6.Преглеждане на ‚логове‘ - при натискане на бутон ‘View Logs’ администраторът може да прегледа всички логове.

Реализация на бизнес логика на оператор

**1.**Оператора разполага с панел на който има информация за текущата наличност в склада на всяка една стока. За да се направи поръчка първо се отбелязват “checkbox”-овете до всяка стока която искаме да заредим . След това се въвежда количество на стоката в съответното поле. При натискане на бутон „Направи поръчка“ се обновява база данни с новите данни.

**2.**Оператора разполага с история на поръчките-при натискане на бутон „Виж история“ се показва нова сцена на която има голям списък, две полета за избор на дата ‚от‘ и ‚до‘ и бутон за търсене. При избиране на период и натискане на бутона се показват детайли за всички поръчки в този период като(дата,количество…).

**3.**Бутон за отваряне на панел с известия- до бутона има червена точка с номер вътре който показва какъв е броят на известията които не са изтрити. При натискане на бутона се отваря панел със списък от информация за известията (дата, съобщение) ,бутон за изтриване на избрано известие и бутон за изчистване на списъка. Брояча на известията се нулира при изчистване на списъка.

Реализация на бизнес логика за винопроизводител

**1.Бутон за калкулация**- при натискане на бутона се пресмята колко вино може да бъде направено от текущата стока и количеството се изобразява в полето под бутона в литри.

**2.Бутон за произвеждане на вино**- при натискането на този бутон се отваря нов панел със различните видове бутилки и знаци + и – съответно за бутилката под която стоят. Чрез контролите се избират ръчно броя на всеки вид бутилки който искаме да произведем или се използва бутона за автоматично разпределяна който ,чрез алгоритъм прави това за нас. След разпределението при натискане на бутон ‚Пълнене‘ се записват новите данни в базата.

**3.История за произведено вино**- избира се период чрез полетата за тази цел и се натиска бутон виж история. След това списъка от ляво на контролите се запълва с данни за произведеното вино за периода.

Реализация на модул за регистриране на събитията в системата

За регистрирането на събития се използва log4j. При първоначално стартиране на приложението автоматична се създава файл log, в който се записва всички логове по дата и час. Достъп до тези логове има само администратор.