DOCUMENTATIE

TEMA 3

Flutur Florina

GRUPA 30222

# CUPRINS

[1. Obiectivul temei 3](file:///C:\Users\ghica\Downloads\PT2023_Template_Documentatie_RO%20(2).doc#_Toc95297885)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](file:///C:\Users\ghica\Downloads\PT2023_Template_Documentatie_RO%20(2).doc#_Toc95297886)

[3. Proiectare 4](file:///C:\Users\ghica\Downloads\PT2023_Template_Documentatie_RO%20(2).doc#_Toc95297887)

[4. Implementare 5](file:///C:\Users\ghica\Downloads\PT2023_Template_Documentatie_RO%20(2).doc#_Toc95297888)

[5. Rezultate 12](file:///C:\Users\ghica\Downloads\PT2023_Template_Documentatie_RO%20(2).doc#_Toc95297889)

[6. Concluzii 13](file:///C:\Users\ghica\Downloads\PT2023_Template_Documentatie_RO%20(2).doc#_Toc95297890)

[7. Bibliografie 13](file:///C:\Users\ghica\Downloads\PT2023_Template_Documentatie_RO%20(2).doc#_Toc95297891)

1. **Obiectivul temei**

Scopul principal al acestui proiect a fost să dezvoltăm o aplicație grafică care să faciliteze administrarea unui depozit de produse, prin conectarea la o bază de date. Prin intermediul aplicației, utilizatorii pot realiza operațiuni precum editarea informațiilor despre clienți și produse, precum și generarea de comenzi.

Aplicația oferă funcționalități avansate care permit adăugarea de noi clienți și produse în baza de date, facilitând astfel gestionarea și actualizarea informațiilor relevante. Utilizatorii au posibilitatea să introducă în sistem detalii despre clienți.

De asemenea, aplicația permite utilizatorilor să editeze și să actualizeze informațiile existente despre clienți și produse. Aceasta include modificarea detaliilor personale ale clienților, cum ar fi email-ul sau varsta, precum și actualizarea datelor despre produse, cum ar fi denumirea, stocul disponibil sau prețul.

Un alt aspect important al aplicației este generarea de comenzi. Utilizatorii pot crea și gestiona comenzile clienților, selectând produsele dorite și specificând cantitățile necesare.

1. **Analiza problemei, modelare, scenarii, cazuri de utilizare**

* **Cerinte functionale:**

Aplicația de management trebuie să permită:

* Adăugarea unui nou client în sistem.
* Editarea informațiilor unui client existent.
* Ștergerea unui client existent din sistem.
* Afișarea listei tuturor clienților existenți în sistem.
* Crearea de comenzi de produse prin selectarea unui client existent și a unui produs existent, specificând cantitatea dorită.
* Afișarea unui mesaj de eroare în cazul în care nu există suficiente produse pentru a completa comanda.

Aplicația de management trebuie să ofere:

* Posibilitatea adăugării unui nou produs în inventar.
* Editarea informațiilor unui produs existent.
* Ștergerea unui produs existent din inventar.
* Afișarea listei tuturor produselor existente în inventar.
* **Cazuri de utilizare**
* Adăugarea unui client nou:

Utilizatorul deschide aplicația și navighează către secțiunea de adăugare client nou.

Introducerea informațiilor despre noul client, cum ar fi nume, adresa și detalii de contact.

Salvarea datelor și actualizarea bazei de date cu noul client.

* Editarea informațiilor unui client existent:

Utilizatorul caută clientul dorit în lista disponibilă în aplicație.

Accesarea opțiunii de editare pentru clientul respectiv.

Modificarea sau actualizarea informațiilor existente, cum ar fi adresa sau numărul de contact.

Salvarea modificărilor și actualizarea bazei de date.

* Ștergerea unui client existent:

Utilizatorul găsește clientul pe care dorește să-l șteargă în lista de clienți.

Selectarea opțiunii de ștergere pentru clientul respectiv.

Confirmarea acțiunii de ștergere și actualizarea bazei de date prin eliminarea clientului.

* Vizualizarea listei tuturor clienților:

Utilizatorul accesează secțiunea de afișare a clienților în aplicație.

Listarea tuturor clienților existenți într-un tabel, afișând informații precum nume, adresă și detalii de contact.

* Adăugarea unui produs nou:

Utilizatorul navighează către secțiunea de adăugare produs nou în aplicație.

Introducerea informațiilor despre noul produs, cum ar fi denumire, preț și stoc disponibil.

Salvarea datelor și actualizarea bazei de date cu noul produs.

* Editarea informațiilor unui produs existent:

Utilizatorul identifică produsul pe care dorește să-l editeze în lista disponibilă în aplicație.

Accesarea opțiunii de editare pentru produsul respectiv.

Modificarea sau actualizarea informațiilor existente, cum ar fi prețul sau stocul disponibil.

Salvarea modificărilor și actualizarea bazei de date.

* Ștergerea unui produs existent:

Utilizatorul găsește produsul pe care dorește să-l șteargă în lista de produse.

Selectarea opțiunii de ștergere pentru produsul respectiv.

Confirmarea acțiunii de ștergere și actualizarea bazei de date prin eliminarea produsului.

* Vizualizarea listei tuturor produselor:

Utilizatorul accesează secțiunea de afișare a produselor în aplicație.

Listarea tuturor produselor existente într-un tabel, afișând informații precum denumire, preț și stoc disponibil.

* Crearea unei comenzi de produse:

Utilizatorul selectează clientul pentru care dorește să creeze o comandă din lista de clienți disponibilă în aplicație.

Adăugarea produselor dorite la comandă prin selectarea din lista de produse și specificarea cantității.

Salvarea comenzii și actualizarea bazei de date cu informațiile respective.

* Mesaj de eroare pentru stoc insuficient:

În cazul în care utilizatorul specifică o cantitate pentru un produs care depășește stocul disponibil, aplicația afișează un mesaj de eroare corespunzător.

Mesajul poate indica că stocul este insuficient pentru a completa comanda sau poate sugera o cantitate disponibilă mai mică.

* Actualizarea stocului de produse:

După finalizarea unei comenzi, aplicația actualizează automat stocul de produse, scăzând cantitățile corespunzătoare din inventar.

3.**Proiectare**

Pentru realizarea proiectului a fost nevoie de introducerea a mai multor clase pentru interfata deoarece s-a adaugat un administrator cu parola deci avem o interfata de log in, dupa avem o interfata de meniu cu log out si la selectare avem diferite operatii.Cazurile de utilizare sunt simple.

S-a folosit tehnica de reflectie pentru a crea o clase generalizate pentru realizarea tabelului si pentru realizarea instructiunilor.

Inca nu sunt tratate toate exceptiile deoarece proiectul este un prototip. Tehnica de reflectie aduce eficienta codului.

Structuri de date folosite au fost structuri de date precum List arraylist defaultlistmodel, defaulttablemodel.

Algoritmii utilizati sunt algoritmi de parcurgere a listelor si eventual a tabelelor,sunt algoritmi care aduc o solutie specifica proiectului deoarece sunt realizati sa evalueze operatii specifice proiectului cum ar fi comenzi sql si parcurgerea setului de date pentru manipularea informatiilor.Informatiile prelucrate fiind inregistrate in tabele corespunzatoare.

Diagrama UML:

A picture containing diagram, text, plan, map

Description automatically generated

**4. Implementare**

* Pachetul BussinessLayer contine la randul lui un alt pachet numit:
  + - Pachetul Validators ce contine
      * Clasa ValidareEmail contine codul de validare a adresei de email ce verifica daca un string respecta formatul specificat pentru o adresa de email.
      * Enumeratia Validare
* Clasa ClientBLL conține logica de afaceri pentru manipularea clienților, inclusiv validarea adresei de email și operațiuni precum inserarea, actualizarea și ștergerea clienților în baza de date.
* Clasa OrdersBLL se ocupă de logica de afaceri legată de comenzile (orders), inclusiv inserarea, ștergerea comenzilor pentru un anumit client și ștergerea unei comenzi după numele produsului
* Clasa ProductBLL se ocupă de logica de afaceri legată de produse, inclusiv inserarea, actualizarea, ștergerea și găsirea unui produs după nume
* Pachetul dao
* Clasa AbstractDAO este o clasă abstractă generică utilizată pentru a gestiona operațiuni comune de acces la date pentru entități. Aceasta implementează metode generice precum inserare , actualizare , ștergere și găsire după ID , utilizând reflexia și JDBC pentru a interacționa cu baza de date. Clasa este parametrizată cu tipul entității (**T**) și oferă metode pentru a crea interogări SQL specifice pentru entitatea respectivă.
* Clasa ClientDAO extinde clasa AbstractDAO<Clients> și este responsabilă de gestionarea operațiunilor de acces la date specifice entității "Clients". Aceasta moștenește metodele generice din clasa de bază
* Clasa OrdersDAO ClientDAO extinde clasa AbstractDAO<Orders> și este responsabilă de gestionarea operațiunilor de acces la date specifice entității "Orders". Aceasta moștenește metodele generice din clasa de bază
* Clasa ProductDAO ClientDAO extinde clasa AbstractDAO<Products> și este responsabilă de gestionarea operațiunilor de acces la date specifice entității "Products". Aceasta moștenește metodele generice din clasa de bază
* Pachetul dataAccesLayers
* Clasa ConnectionFactory este responsabilă de gestionarea conexiunii la baza de date MySQL. Aceasta utilizează JDBC pentru a crea și închide conexiuni, declanșând excepții în cazul unor erori de conectare sau închidere. Clasa definește metode statice pentru a obține o conexiune (getConnection()) și pentru a închide conexiuni, declarații (Statement), instrucțiuni pregătite (PreparedStatement) și seturi de rezultate (ResultSet).
* Pachetul model
* Clasa Clients reprezintă o entitate a unui client. Aceasta are câmpurile private id, nume, email și varsta, împreună cu constructori și metode de acces și modificare pentru aceste câmpuri.
* Clasa Orders reprezintă o entitate a unei comenzi. Aceasta are câmpurile private id, nume\_client, nume\_produs, pret și cantitate, împreună cu constructori și metode de acces și modificare pentru aceste câmpuri
* Clasa Products reprezintă o entitate a unui produs. Aceasta are câmpurile private id, nume, cantitate și pret, împreună cu constructori și metode de acces și modificare pentru aceste câmpuri.
* Clasa Tabel extinde clasa AbstractTableModel și oferă un model de tabel personalizat utilizat pentru afișarea și gestionarea datelor într-un JTable
* Pachetul presentation
* Clasa Controller gestionează interacțiunea între interfața utilizator și logica de afaceri a aplicației. Aceasta primește evenimentele generate de utilizator și le procesează în funcție de acțiunile corespunzătoare. De asemenea, realizează schimbările de vizibilitate între diferitele ferestre ale aplicației în funcție de acțiunile utilizatorului.
* Clasa ViewClienti reprezintă interfața grafică pentru gestionarea operațiilor legate de clienți. Aceasta include un JFrame care conține câmpuri de introducere pentru nume, email și vârstă ale clienților, precum și un JComboBox pentru selectarea unei operații (adaugare nouă, editare sau ștergere) și un JButton pentru a declanșa acțiunea.
* Clasa ViewComenzi reprezintă interfața grafică pentru gestionarea operațiilor legate de comenzi. Aceasta include un JFrame care conține un JComboBox pentru selectarea unui client, un JComboBox pentru selectarea unui produs, câmpuri de introducere pentru cantitate și preț, precum și un JButton pentru a declanșa acțiunea.
* Clasa ViewListe reprezintă interfața grafică pentru afișarea listelor de clienți, produse și comenzi. Aceasta include un JFrame care conține trei JTable-uri pentru afișarea datelor într-un format tabular. Fiecare tabel este plasat într-un JScrollPane pentru a permite derularea verticală a conținutului în cazul în care lista de date depășește dimensiunea afișată pe ecran.
* Clasa ViewLogare reprezintă interfața grafică pentru ecranul de autentificare. Aceasta conține un JFrame care conține un JTextField pentru introducerea numelui utilizatorului și un JPasswordField pentru introducerea parolei. De asemenea, include și un JButton "Enter" pentru a iniția acțiunea de autentificare.
* Clasa ViewMeniu reprezintă interfața grafică pentru meniul principal al aplicației. Aceasta conține un JFrame care conține mai multe butoane pentru diferite opțiuni, inclusiv "Clienti", "Produse", "Comenzi", "Liste Clienti/Produse" și "Iesire".
* Clasa ViewProduse reprezintă interfața grafică pentru gestionarea produselor în aplicație. Aceasta conține un JFrame care conține câmpuri de text și butoane pentru introducerea și modificarea informațiilor despre produse.
* Clasa ViewUtile conține metode pentru autentificare, încărcarea datelor în combobox-uri și generarea de liste și tabele pentru diferite clase.
* Pachetul start
* Clasa Start punctul de pornire al aplicației. În metoda main, se creează o instanță a clasei Controller și se pasează instanțe ale diferitelor clase de prezentare.

**5.Rezultate**

La rularea aplicației, prima fereastră afișată este fereastra de logare, reprezentată de clasa ViewLogare din pachetul presentation. Această fereastră servește drept punct de autentificare pentru utilizatori, permițându-le să se conecteze înainte de a accesa întregul program.

Fereastra de logare conține câmpuri de introducere pentru utilizator și parolă. Utilizatorul trebuie să completeze aceste câmpuri cu informațiile lor de autentificare și apoi să apese butonul "Enter" pentru a trimite cererea de autentificare.

A screenshot of a computer

Description automatically generated

Dacă autentificarea este reușită, fereastra de logare se închide și este afișată fereastra principală a aplicației, denumită "Meniul principal" și reprezentată de clasa **ViewMeniu**.

Fereastra "Meniul principal" de exemplu, butonul "Clienti" deschide fereastra de gestionare a clienților , butonul "Produse" deschide fereastra de gestionare a produselor , iar butonul "Comenzi" deschide fereastra de gestionare a comenzilor . Butonul "Liste Clienti/Produse" permite utilizatorului să acceseze o fereastră specială care afișează liste cu clienți, produse și comenzi.

Fereastra principală oferă, de asemenea, un buton "Iesire" care permite utilizatorului să închidă complet aplicația.

A screenshot of a computer

Description automatically generated with medium confidence

Când utilizatorul apasă butonul "Clienti" din fereastra principală a aplicației, se deschide fereastra de gestionare a clienților (ViewClienti), care permite utilizatorului să adauge, să editeze sau să șteargă clienți din tabelul afișat.

A screenshot of a computer

Description automatically generated

Când utilizatorul apasă butonul "Produse" din fereastra principală a aplicației, se deschide fereastra de gestionare a produselor (ViewProduse), care permite utilizatorului să adauge, să editeze sau să șteargă produse din tabelul afișat.

A screenshot of a computer

Description automatically generated

Când utilizatorul apasă butonul "Comezi" din fereastra principală a aplicației, se deschide fereastra de gestionare a comenzilor (ViewComenzi), care permite utilizatorului să faca o comanda selectand persoana ce doreste sa cumpere un anumit produs din lista de produse din tabelul afișat.

A screenshot of a computer

Description automatically generated

Când utilizatorul apasă butonul "Lista " din fereastra principală a aplicației, se deschide fereastra de de vizualizare a clientilor, a produselor si a comenzilor (ViewListe).

A screenshot of a computer

Description automatically generated with medium confidence

**6.Concluzii**

Pe parcursul realizării proiectului simulator de gestionare a unui depozit, am dobândit cunoștințe și abilități valoroase, precum:

* Utilizarea bazelor de date relaționale pentru stocarea și gestionarea datelor într-un mod structurat și eficient.
* Înțelegerea și aplicarea arhitecturii stratificate (layered architecture pattern) pentru o dezvoltare mai ușoară, întreținere și extensibilitate a proiectului.
* Utilizarea tehnicilor de reflexie pentru a examina și manipula structura internă a codului în timpul execuției, oferind flexibilitate și capacitatea de a face operații dinamice.

In viitor as trata mai multe cazuri, precum resetarea indecsilor atunci cand stergem un client, produs sau comanda, precum si realizarea unei facturi electronice care sa tipareasca bonul pentru comanda facute.

**7.Bibliografie**

* <https://ro.ephesossoftware.com/articles/programming/how-to-connect-to-a-mysql-database-with-java.html>
* <https://uncoded.ro/lucrul-cu-baze-de-date-in-java/>
* [*https://app.diagrams.net/*](https://app.diagrams.net/)
* <https://www.rasfoiesc.com/educatie/informatica/baze-de-date/Interfatarea-cu-bazele-de-date16.php>