

**FACULTATEA: Automatica si Calculatoare**

**SPECIALIZAREA: Calculatoare si tehnologia informatiei**

**DISCIPLINA: Tehnici de Programare**

**AN UNIVERSITAR: 2019/2020**

**PROIECT: Order Management**

**Realizat de :**

Ghiura Darius

**Grupa:** 30227

**CUPRINS**

1. **Obiectivul temei**
2. **Analiza problemei, modelare, scenarii, cazuri de utilizare**
3. **Proiectare**
4. **Implementare**
5. **Rezultate**
6. **Concluzii**
7. **Bibliografie**
8. Obiectivul temei

Obiectivul temei este proiectarea si implementarea unei aplicatii care proceseaza comenzile unui depozit. Aplicatia trebuie sa poata stoca produse, clienti si comenzi in tabelele unei baze de date relationale.

Aplicatia citeste dintr-un fisier operatiile de efectuat si creeaza fisiere PDF care pot fi : o factura cand este plasata o comanda, un mesaj de under-stock daca stockul unui produs nu este suficient pentru a realiza comanda sau un tabel care afiseaza tabelul din baza de date.

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

Pentru a implementa un sistem de gestionare a unui depozit, aplicatia trebuie sa poata insera, sterge si modifica clienti, produse si comenzi din baza de date.

Aplicatia citeste instructiunile dintr-un fisier, deci are nevoie de un Parser care sa preia acele instructiuni si sa le execute.

Instructiunile din fisier pot fi :

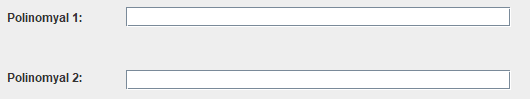
* Insert client: Ion Popescu, Bucuresti : Insereaza clientul cu numele Ion Popescu si adresa Bucuresti in baza de date
* Delete client: Ion Popescu : Sterge clientul cu numele Ion Popescu din baza de date
* Insert product: apple, 20, 1 : Insereaza produsul cu numele „apple”, cantitatea 20 si pretul 1 in baza de date
* Delete product: apple : Sterge produsul cu numele „apple” din baza de date
* Order: Ion Popescu, apple, 5 : Plase
* Report client
* Report order
* Report product

Pentru fiecare comanda plasata un fisier PDF cu numele „order + id-ul comenzii + „\_bill.pdf” va fi creat. Daca comanda nu se poate efectua din cauza stocului insuficient al unui anumit produs se va crea un fisier PDF cu numele : „order + id\_ul clientului + \_understock.pdf”.

Polinoamele introduse in interfata grafica trebuie sa respecte urmatorul format : „ax^b” , unde a reprezinta coeficientul monomului si este un numar intreg (daca a = 1, acesta poate sa nu fie scris , iar daca a = -1 se poate scrie doar „-”), iar b este exponentul monomului care este un numar intreg pozitiv (daca b = 1 se poate scrie doar x, fara „^”, iar daca b = 0 se scrie doar coeficientul).

Scenariul aplicatiei este reprezentat de introducerea polinoamelor si selectarea operatiei dorite.

**Cazuri de utilizare :**

* **Adunare, scadere, inmultire, impartire**
* Se introduc polinoamele in zonele respective, respectand formatul specificat anterior;
* In caz de nerespectare a formatului pe ecran va aparea o fereastra de eroare cu mesajul „Incorrect input”. O eroare va aparea si pentru introducerea polinomului „0” in cazul impartitorului la operatia de impartire, cu mesajul „Cannot divide by 0”.
* Se apasa butonul corespunzator operatiei dorite;



* Se apasa pe butonul „=” pentru afisarea rezultatului;



* Rezultatul operatiei va fi afisat in zona din stanga, iar pentru impartire in zona din dreapta va fi afisat restul;
* Daca nu se introduce un polinom acesta va fi considerat ca fiind 0;
* Pentru stergerea polinoamelor de intrare, se apasa butonul de CE (Clear Entry).
* **Derivare, integrare**

In cazul derivarii si integrarii, procesul de introducerea datelor, selectarea operatiei si afisarea rezultatului este acelasi, doar ca va fi luat in considerare doar polinomul introdus in zona „Polynomial 1”.

Se pot efectua oricate operatii fara a mai fi nevoie de introducerea datelor de intrare.