**Lant de farmacii**

Samoila Alexandra Marina

Grupa 30238

1.Textul problemei

Cerinta temei 1:

Descrierea aplicației:

Dezvoltați (analiză, proiectare, implementare) o aplicație desktop care poate fi utilizată într-un lanț de farmacii / plafare.

Aplicația va avea două tipuri de utilizatori: angajat al lanțului de farmacii / plafare și administrator.

Utilizatorii de tip angajat al lanțului de farmacii/plafare pot efectua următoarele operații după autentificare:

* Vizualizarea tuturor produselor dintr-o farmacie selectată / dintr-un plafar selectat;
* Filtrarea produselor după următoarele criterii: disponibilitate, valabilitate, preț, producător;
* Căutarea unui produs după denumire;
* Vizualizarea unor statistici legate de produse: procente după disponibilitate, termen de valabilitate, categorii, pret utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).
* Operații CRUD în ceea ce privește persistența produselor;
* Salvare rapoarte / liste cu produse în mai multe formate: csv, xml, json.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare:

* Operații CRUD pentru informațiile legate de utilizatorii de tip angajat.

Constrângeri ale aplicației:

* Datele utilizate în aplicație vor fi socate într-un fișier binar.
* Utilizați șabloanele de proiectare creaționale Abstract Factory și Builder.

Cerinta temei 2 :

Obiectiv: Obiectivul acestei teme este familiarizarea cu șablonul architectural Client/Server.

Cerințe:

Transformați aplicația desktop implementată la tema 1 într-o aplicație client-server. Mare parte din modelul logic (inclusiv persistența) va face parte din server, mai puțin partea care poate fi determinată din datele primite de la server (generare rapoarte, statistică, etc.).

* În faza de analiză realizați diagrama cazurilor de utilizare.
* În faza de proiectare realizați diagrama de clase corespunzătoare aplicației server și diagrama de clase corespunzătoare aplicației client.
* În faza de implementare scrieți cod pentru îndeplinirea tuturor funcționalităților precizate de diagrama cazurilor de utilizare utilizând unul dintre următoarele limbaje de programare: C#, C++, Java, Python.
* Finalizarea temei va consta în predarea unui director ce va cuprinde:
* Un fișier word care cuprinde numele studentului, grupa, enunțul problemei și instrumentele utilizate.
* Un fișier cu diagramele UML realizate;
* Directorul cu aplicația implementată.

Constrângeri ale aplicației:

* Pentru persistență se va utiliza o bază de date relațională (MySQL, SQL Server, Oracle, etc.).
* Se cere utilizarea arhitecturii client/server, NU web client/server.

Observație: Se va utiliza baza de date doar în aplicațiile unde este necesară persistența.

Cerinta temei 3 :

Obiectiv: Obiectivul acestei teme este familiarizarea cu șablonul de proiectare comportamental Observer.

Cerințe: Optimizați aplicația client/server implementată la tema 2 astfel încât să utilizați șablonul de proiectare comportamental Observer în dezvoltarea aplicației client. În plus, interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională (implicit limba română).

* În faza de analiză realizați diagrama cazurilor de utilizare și diagramele de activități corespunzătoare tuturor cazurilor de utilizare.
* În faza de proiectare realizați:
* diagrama de clase corespunzătoare aplicației server;
* diagrama de clase corespunzătoare aplicației client;
* diagrame de secvență corespunzătoare tuturor cazurilor de utilizare.
* În faza de implementare scrieți cod pentru îndeplinirea tuturor funcționalităților precizate de diagrama cazurilor de utilizare utilizând unul dintre următoarele limbaje de programare: C#, C++, Java, Python.
* Finalizarea temei va consta în predarea unui director ce va cuprinde:
* Un fișier word care cuprinde numele studentului, grupa, enunțul problemei, instrumentele utilizate și detalii despre modul în care a fost utilizat șablonul de proiectare Observer.
* Un fișier cu diagramele UML realizate;
* Directorul cu aplicația implementată.

Observație: Pentru persistență se va utiliza o bază de date relațională (MySQL, SQL Server, Oracle, etc.). Se va utiliza baza de date doar în aplicațiile unde este necesară persistența.

2.Limbajul de programare ales

Am ales sa lucrez in java deoarece acest limbaj imi place cel mai mult si il stiu cel mai bine, iar la vara as vrea sa fac un internship pe java si atunci acest proiect ma ajuta sa fiu mai bine pregatita.

***Java:*** 11

***Mediul de dezvoltare****:* Eclipse

***Versiunea mediului:*** 2019-12 (4.14.0)

Ca si baza de data am lucrat cu MySQL. Pentru asta am folosit phpMyAdmin si XAMPP.

***Credențiale:***

Administrator:

User: admin

Pass: admin

Angajat Plafar

User: p1

Pass: p1

Angajat Farmacie

User : f1

Pass : f1