**Enuntul problemei:**

**Problema 15**

**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.

**Limbaj de programare:** Java

**Mediu de dezvoltare:** Eclipse 2019-09, Java 8, JDK 12.0.2

**Librarii folosite:** Json Simple pentru salvarea fisierului JSON, JAXB pentru salvarea fisierului XML, JFreeChart pentru a vizualiza statisticile, Swing pentru interfata grafica, MySQL Connector pentru a face conexiunea cu baza de date MySQL, Java RMI pentru transferul de date(obiecte) intre aplicatia client si server, Java SocketServer pentru server si Java Socket pentru client

**Baza de date:** MySQL 8.0

**Credentiale pentru un angajat:**

Username: third

Password: pass3

**Credentiale pentru un administrator:**

Username: first

Password: pass1