

### Cerinte generale

Se cere realizarea unei aplicatii (desktop sau web) cu arhitectura stratificata (presentation, business logic, data access), conform cerintelor problemei asignate/alese. Limbajul de implementare va fi un limbaj orientat obiect, la alegere dintre cele deja studiate. Interfata cu utilizatorul se recomanda a fi realizata cu un instrument vizual (ex. NetBeans, WindowBuilder, Visual Studio). Se cere aplicarea sablonului arhitectural MVC (Model-View-Controller). Aplicatia trebuie sa exploateze o baza de date relationala prin stratul de data access. Se impune ca interactiunea cu baza de date sa se faca prin intermediul unei biblioteci ORM (Object-Relational Mapping, ex. Hibernate/EclipseLink, NHibernate/Entity Framework). Proiectul trebuie sa ofere functii de tip CRUD (Create-Read-Update-Delete) pentru cel putin o entitate din domeniul problemei (clasa din Model).

Obligativu, toate modelele UML ale aplicației (functional, structurale, comportamentale) vor fi construite cu ajutorul unui instrument CASE (Computer Aided Software Engineering, ex. StarUML) și vor fi incluse în cadrul aceluiași proiect, structurat adecvat, conform etapelor ciclului de viață al unui sistem soft (pentru a putea urmări aspecte legate de sincronizare). Implementarea claselor din pachetul Model poate fi realizata partial folosind o operatie de tip forward/direct engineering, aplicata cu ajutorul instrumentului CASE folosit.

### Laboratoare: activități și artefacte

#### **FAZA I (Colectarea si analiza cerintelor)**

##### **Laboratorul 1**

- momente organizatorice, stabilirea temelor individuale

##### **Laboratorul 2**

- modelul functional (diagrama UML a cazurilor de utilizare + descrierea textuala/tabelara a cazurilor de utilizare, cu scenariul normal si scenariile alternative/de exceptie posibile)
- planificarea cazurilor de utilizare pe 3 iteratii

##### **Laboratorul 3**

- modelul conceptual (reprezentat folosind o diagrama UML de clase)
  - !!! modelul conceptual reda exclusiv entitatile *din domeniul problemei* si relatiile aferente
- prototipul interfetei grafice cu utilizatorul

#### **FAZA II (Proiectare si implementare iteratia 1)**

##### **Laboratorul 4**

- reprezentarea scenariilor aferente cazurilor de utilizare din iteratia 1, prin diagrame de interactiune (diagrame de secventa si de comunicare/colaborare)
- rafinarea diagramei de clase initiale, cu includerea entitatilor din domeniul solutiei aferente iteratiei 1

##### **Laboratorul 5**

- implementare iteratia 1: prima versiune functionala a aplicatiei

### **FAZA III (Proiectare si implementare iteratii 2, 3)**

#### **Laboratorul 6**

- reprezentarea scenariilor aferente cazurilor de utilizare din iteratia 2, prin diagrame de interactiune (diagrame de secventa si de comunicare/colaborare)
- rafinarea diagramei de clase initiale, cu includerea entitatilor din domeniul solutiei aferente iteratiei 2
- implementare iteratia 2: a doua versiune functionala a aplicatiei

#### **Laboratorul 7**

- reprezentarea scenariilor aferente cazurilor de utilizare din iteratia 3, prin diagrame de interactiune (diagrame de secventa si de comunicare/colaborare)
- rafinarea diagramei de clase initiale, cu includerea entitatilor din domeniul solutiei aferente iteratiei 3
- iteratia 3: versiunea finala a aplicatiei

### **Termene de predare si artefacte/activitati implicate**

- termen faza 1 (lab1 + lab2 + lab 3) - sapt. 5/6
- termen faza 2 (lab 4 + lab 5) - sapt. 9/10
- termen faza 3 (lab 6 + lab 7) - sapt. 13/14

Fiecare faza se termina cu o nota. Nota finala se calculeaza ca medie aritmetica a notelor fazelor. Predarea dupa termen se penalizeaza cu 1 punct / saptamana de intarziere. Termenul limita de predare a materialelor de proiect este ultima sedinta de laborator din semestru.

### **Probleme**

(domeniile si cerintele sunt orientative si pot fi schimbate/rafinare, in colaborare cu cadrul didactic coordonator)

#### **1. BIBLIOTECA**

O biblioteca ofera abonatilor sai o lista de carti ce pot fi imprumutate. Pentru un abonat, se retin în sistem (cel puțin) informații legate de cnp, nume, adresa, telefon și un cod unic de identificare a acestuia în cadrul bibliotecii. Fiecare carte poate exista în unul sau mai multe exemplare, identificate prin coduri unice. Biblioteca are mai multe terminale, de unde abonatii pot sa imprumute carti. Pentru a putea folosi un terminal, un abonat trebuie să se autentifice. Dupa autentificare, acesta vede lista exemplarelor disponibile în acel moment si poate imprumuta unul sau mai multe. Pentru restituirea cartilor, exista un singur punct de lucru, deservit de un bibliotecar. Dupa fiecare împrumut/restituire, toți utilizatorii terminalelor bibliotecii văd lista actualizata a cartilor disponibile.

#### **2. AGENTI DE VANZARI**

O firma are mai multi agenti de vanzari, care fac comenzi pentru clientii firmei si le distribuie acestora produsele. Pentru desfasurarea activitatilor, firma ofera o aplicatie prin care agentii pot comanda produse. Aplicatia pune la dispozitia fiecarui agent al firmei un terminal prin care:

- agentul vizualizeaza lista tuturor produselor vândute de firma, împreuna cu preturile aferente și cantitatile existente pe stoc.
- agentul poate comanda o cantitate dintr-un produs. După orice comanda valida, toti agentii logati în aplicație vor vedea lista actualizata a stocurilor (este posibil ca declansarea înregistrării unei comenzi din partea unui agent sa conduca la un mesaj informativ "cantitate insuficienta în stoc").

### 3. FARMACIE SPITAL

Un spital are o farmacie proprie si mai multe sectii. Necesarul de medicamente pe sectii este deservit de catre farmacia spitalului. Spitalul pune la dispozitia acestora o aplicatie pentru gestiunea comenzilor de medicamente ale sectiilor si onorarea acestor comenzi din partea farmaciei. Aplicatia este compusa din:

- *mai multe terminale localizate in fiecare sectie:* Personalul medical de pe sectii foloseste aceste terminale pentru a inregistra comenzile de medicamente. Un element de comanda refera medicamentul dorit si cantitatea solicitata. De pe o sectie se pot inregistra oricâte comenzi. Imediat dupa inregistrarea unei comenzi, aceasta poate fi consultata la farmacie.
- *un terminal pentru farmacie:* Farmacistii observa lista comenzilor facute de pe sectii, prioritizata după momentul sosirii acestora. Pentru onorarea unei comenzi, farmacistul o selecteaza din lista, (pregateste medicamentele si le trimite pe sectii) dupa care declanseaza un buton "comanda onorata". În urma acestei acțiuni, comanda este eliminata din lista afisata de farmacie si apare ca si onorata in fereastra sectiei care a inițiat-o.

### 4. MONITORIZARE ANGAJATI

O firma si-a creat o infrastructura prin care seful monitorizeaza angajatii prezenti la lucru si le traseaza sarcini individuale. Firma are o aplicatie care ofera:

- *o fereastră pentru sef,* cu ajutorul careia seful vede lista angajatilor prezenti în firma, un element din lista precizând numele angajatului si ora la care s-a logat în sistem. De asemenea, seful poate transmite o sarcina unui angajat prezent astfel: selecteaza angajatul din lista, introduce o descriere a sarcinii si declanseaza un buton "transmite sarcina". Imediat dupa transmiterea unei sarcini, aceasta poate fi consultata de catre angajatul respectiv.
- *câte o fereastră pentru fiecare angajat:* Atunci când angajatul vine la serviciu, introduce ora sosirii si declanseaza un buton "prezent". Imediat dupa declansarea butonului, seful vede în lista lui ca angajatul este prezent. În continuare, cât timp angajatul sta la serviciu, el primeste si, în consecinta, vede în fereastra lui, sarcinile transmise de sef. La plecare, angajatul închide fereastra, moment în care seful este notificat de delogarea acestuia din sistem.

### 5. URMARIRE BUG-URI

O firma producatoare de software pune la dispozitia programatorilor si verifcatorilor sai un sistem prin care acestia pot sa comunice electronic. Astfel, fiecare dintre angajatii mentionati are la dispozitie un terminal prin care:

- *verifcatorul* poate înregistra un bug, dându-i o denumire si o descriere; imediat dupa înregistrarea bug-ului, toti programatorii vad lista bug-urilor actualizata cu obiectul nou introdus;
- *programatorul* vizualizeaza lista bug-urilor; de asemenea, programatorul poate selecta un bug din lista si poate declansa un buton prin care declara ca bug-ul a fost eliminat, caz în care bug-ul este scos din lista tuturor programatorilor.

### 6. REZERVARI LOCURI

O institutie teatrala pune la dispozitia spectatorilor un sistem pentru rezervarea locurilor la spectacole. În fiecare zi, institutia are o singura reprezentatie, la care spectatorii pot rezerva locuri începând de dimineata. Pentru un loc în sala, sunt memorate urmatoarele informatii: pozitie (rândul x, loja y etc.), numar si pret. Terminalele puse la dispozitia spectatorilor afiseaza întreaga configuratie a salii, precizând pentru fiecare loc pozitia, numarul, pretul si starea (liber sau rezervat). Folosind un astfel de terminal, spectatorul își poate introduce datele personale, poate selecta unul sau mai multe locuri si poate declansa un buton pentru rezervarea lor. După fiecare rezervare, toate terminalele vor afisa situația actualizata referitor la ocuparea sălii.