

## ***Cumparare bilete online CTP - Cluj.***

Ca și client CTP îmi doresc să am la dispoziție o aplicație prin care să pot achiziționa bilete de călătorie pentru diferite zone teritoriale (preț diferit per zonă). Aceste bilete trebuie să le pot scana ulterior la aparatele dedicate din bus.

Pentru a putea face o achiziție înțeleaptă îmi doresc ca la cumpărare să îmi fie furnizate mai multe informații despre orarul și rutele autobuselor. Îmi doresc să am posibilitatea să pot selecta o stație de pornire, una de destinație și introduce o dată, iar aplicația să-mi furnizeze cele mai bune sugestii de autobuse (preferabil cât mai puține) ordonate după ora de plecare, pentru a ajunge din punctul A în punctul B. Dacă nu selectez nici o dată pentru această filtrare, implicit se consideră că am optat pentru data curentă. De asemenea mi-ar plăcea să pot face filtrări și în funcție de numărul autobusului pentru a vedea orarul și traseul acestuia.

După ce am adăugat biletele în coșul de cumpărături, voi putea vizualiza suma totală de plată și voi fi întrebat dacă doresc să finalizez plata. În caz afirmativ voi fi direcționat către o pagină de plată unde îmi voi introduce informațiile personale și bancare necesare finalizării plății biletelor.

Este foarte important ca în fiecare moment să-mi pot verifica situația contului și anume câte bilete am și pentru ce zone. Pentru verificarea contului și pentru inițierea plății sunt necesare date de login (username/parolă). Un utilizator nelogat poate doar vizualiza prețul biletelor și face filtrări.

Bonus: crearea interfeței pentru validarea biletelor. La fiecare scanare de bilet, aparatul mă va informa dacă biletul/contul meu a fost scanat cu succes (moment în care nr de bilete din cont vor descresce) sau nu (biletul nu mai e valid pentru că a depășit perioada de valabilitate de 30 minute sau nu există în cont bilete pentru zona curentă).

## **Mediateca**

Se dau fișiere (de diferite tipuri – filme, muzica, imagini) într-un număr mare (~100, oricum mult mai multe decât 10 pentru fiecare tip de fișier-filme, muzica, imagini). O parte din acestea sunt deja etichetate și putem reedita etichetele lor, altele sunt neetichetate și le putem atașa etichete. Etichetele posibile sunt de forma:

- Gen: filmografic/muzical/fotografic,
- Context: (locatie, moment de timp) în care au fost create sau modificate (ex. petrecerea X, evenimentul Y, situația Z).

Utilizatorul mediatecii poate solicita anumite fișiere printr-un text cu o structură predefinită (de ex. "Imaginile de la majoratul Felicie"). De asemenea, utilizatorul dorește să poată rafina căutarea fișierelor în funcție de anumite etichete (autor, data, dimensiuni, tip de fișier, cuvinte cheie, etc.) – de ex. "Fișierele create de Mihai până la ora 22:00".

Cerința suplimentară (bonus): Să se extragă anumite fișiere neetichetate și aplicația să le propună în mod automat etichete precum: data și ora, dimensiunea în memorie a fișierului, tipul de fișier în funcție de extensie, cuvinte cheie din numele fișierului.

## **Product documentation**

Company X, leading tech company in both hardware and software solutions is seeking help in developing a platform for its clients. The platform is supposed to be used by both

Company X employees and its associates (like Company A, Company B, Company C, etc.) to create documentation for company products. Such company products include electric engines, mobile phones, AC systems, trains, windmills and so on. Products and articles are to be stored in files for easy access.

On this idea, you are tasked with creating 4 different components.

- Component 1 will be called PMDImport (PMD standing for Product Management Database) Windows Task. Its purpose will be to insert randomly generated products into a products file. The purpose of this project is to simulate an application capable of synchronizing your database with an external database. Such synchronization includes products addition, update and deletion. Since you do not have an external database, the application will just add new products every 10 minutes.
- Component 2 will be called Authoring. The Authoring is a platform where Siemens solution Managers and Siemens associates can create articles. These articles represent the documentation associated to a product. Such articles can be of 2 types: TextBased articles and PdfBased articles. As the name implies PdfBased articles have at their core a pdf representing a usage manual, a certificate, a brochure or a product characteristic. TextBased articles will require a user a write text. There can be a total of 8 different TextBased Articles: FAQ,ExampleOfUse, Compatibility, Download, TopicPage, GeneralEntry, productNote, VersionHint. All articles created in Authoring will be stored in a file. The general architecture of the platform is N-tier where N represents 5 layers: UI, Controller, Repository, Testing and BusinessLayer.
- Component 3 will be called Delivery. This platform targets to be used end clients that have bought a product and seek usage documentation. This way, the platform displays published articles from Authoring. For displaying one you need filters. The demand team from Company X requires you for easier access to articles that you a combination of multiple filters.
- Component 4 will be called PublishingService. PublishingService will be a Windows task. In order for an article created in Authoring to appear in Delivery, it needs to be first published. Upon publishing an article, an entry will be created in a file called "PublishingQueue" that will tell the service which articles needs publishing. The publishing consists of copying an article from the Authoring file (called tblObjectAuthoring) to the Delivery file (bearing the same name). The publishing service will run at intervals of 1 minute.

It is mandatory to note that the Authoring and Delivery platforms are user-based platforms that require special user rights to access. A user with access to delivery will not be able to access Authoring while users with Authoring rights can only access the Authoring platform. You are free to create admin users for testing purposes.

Etape de dezvoltare:

Etapa	Activitatea
1	stabilirea echipei (3 personae) si a rolurilor, clarificarea si definirea functionalitatilor
2	modelarea aplicatiei (UML diagram -> diagrama de clase/actiuni)
3	Proof of concept -> GUI/Consola
4	live demo