

DATA SERVICE CENTER FOR STUDENTS' MOBILITY (EUROSTUDENTS)

MIHAI Alexandra-Monica, MUNTEANU Beniamin

Facultatea de Automatică și Calculatoare, Ingineria Sistemelor, Anul II, mail@monicamihai.ro

Conducător științific: prof. dr. ing. **Irina SEVERIN**, prof. dr. ing. **Mihai CARAMIHAI**

ABSTRACT: This paper presents the process of Erasmus students' management, using an application based on the object-oriented programming paradigm. Currently, students that come to the IMST faculty with Erasmus scholarships accomplish this process via email. Within the faculty, data management is done through the correspondence of several columns in a series of Excel tables. This project allows remote students to register by filling the online application form with all the required information and selecting the courses they intend to attend. The Administrator (the Erasmus Coordinator) can see the registered students and generate several reports.

KEY WORDS : automation, accessibility, security, portability

1. Introducere

În cadrul universităților din întreaga lume, există un număr mare de studenți ce se înscriu în programul Erasmus, pentru a studia în străinătate. Acest proces se desfășoară în două moduri principale:

1. Datele de contact ale unui profesor ce se ocupă de gestionarea studenților Erasmus sunt puse la dispoziție pe website-ul universității. Un potențial candidat folosește aceste date pentru a lua legătura directă cu profesorul. Schimbul de date este realizat via e-mail.

2. Website-ul universității conține o platformă prin intermediul căreia studentul își poate crea un cont și se poate înscrie. Cererea de înscriere, precum și informațiile sunt astfel preluate și gestionate de către o persoană ce se ocupă de managementul studenților.

Prima metodă aduce cu sine un deficit semnificativ, având în vedere că informațiile studentului sunt gestionate manual, apărând astfel riscuri de securitate. Posibilitatea erorii umane este de asemenea o vulnerabilitate majoră ce trebuie luată în considerare. Pe de altă parte, a doua metodă presupune un proces automat al înscrierii, studenții fiind înregistrați într-o bază de date în momentul în care își crează un cont. Ulterior, coordonatorii programului Erasmus din cadrul universității pot vizualiza și gestiona studenții cu ușurință.

Prezentul proiect a fost inițiat din necesitatea de a gestiona automat procesul de management al studenților în condițiile creșterii numărului de studenți incoming în cadrul Facultății de Ingineria și Managementul Sistemelor Tehnologice. Gestionarea datelor studenților străini se realizează prin intermediul unor tabele Excel, unde sunt adăugate și prelucrate manual toate informațiile. Acest lucru cere foarte mult timp și poate genera erori umane. Soluția oferită este automatizarea procesului. Proiectul are ca scop dezvoltarea și implementarea unei aplicații de management al studenților Erasmus incoming, oferind funcționalități pentru automatizarea procesului de înregistrare și prelucrare a datelor acestora.

Comparativ cu Facultatea IMST din cadrul Universității Politehnica București, peste 50 % dintre Universitățile Europene desfășoară în prezent activitatea de management al studenților incoming prin intermediul unor aplicații Web integrate în website-urile acestora.

2. Soluția aleasă ca implementare la IMST/UPB. Obiective, facilități și limitări

Varianta actuală a aplicației, este dezvoltată în C++, deoarece a început ca o temă de proiect semestrială, în cadrul obiectului de studiu Programare Orientată pe Obiecte. Cerința principală a acestuia a fost utilizarea limbajului de programare C++, întrucât acesta a fost limbajul suport din cadrul materiei. În continuare, după finalizarea cursului, proiectul a continuat prin efort propriu, cu o tehnologie mai performantă, și anume JAVA.

În cele ce urmează, se vor prezenta funcționalitățile principale ale aplicației actuale. Aceasta trebuie să acceseze și modifice o bază de date ce conține toate datele necesare pentru o gestionare optimă a studenților. Înregistrările din baza de date corespund unor câmpuri de tip nume, prenume, adresă de e-mail, țara de origine, cursuri selectate etc. Aplicația trebuie să poată să opeze cu trei tipuri de utilizatori : STUDENT, PROFESOR și ADMINISTRATOR. Toți utilizatorii pot accesa aplicația, autentificându-se prin intermediul unei matrice de drepturi.

Studentul poate crea un cont pentru ca apoi să poată accesa aplicația folosind datele introduse. După autentificare, studentul poate să își vizualizeze și modifice profilul, să selecteze cursuri la care ar dori să participe, dacă acestea sunt puse la dispoziție de către universitate în semestrul în care ar vrea să studieze în străinătate.

Un alt tip de utilizator este **profesorul**. Acesta poate să se înregistreze și autentifice în aceeași manieră ca un student. După ce se loghează, profesorul poate să își vizualizeze și modifice profilul, având totodată posibilitatea de a interoga baza de date, pentru a extrage informații, cum ar fi : numărul de studenți participanți la cursul asociat lui, precum și datele acestora.

Administratorul este rolul asociat coordonatorului Erasmus din cadrul universității, prin urmare, un “super-user” al aplicației. Poate vizualiza și edita toți profesorii, studenții și cursurile. De asemenea, administratorul poate genera rapoarte precum: câți și ce studenți participă la un curs, ce cursuri a selectat fiecare student, etc.

Materiile disponibile, numărul de ore alocate și cadrele didactice sunt de asemenea disponibile în aplicație.

Pe parcursul dezvoltării am sesizat următoarele probleme aduse de implementarea în C++ peste QT. Punctele slabe ale aplicației ar fi:

- securitatea;
- lipsa portabilității – este generat un fișier executabil care rulează doar pe sistemul de operare Windows, prin descărcare local;
- limitări de design care împiedică un mod intuitiv de utilizare al aplicației;

3. Descriere funcțională

ErasmusIO (primul nume al aplicației) a fost gândită inițial în QT Creator, un mediu de dezvoltare pentru C++, Javascript și QML. Acest IDE oferă posibilitatea de a crea o interfață grafică într-un mod accesibil, cerința necesară pentru aplicația de management al studenților. QT Creator conține un editor de cod și diverse unelte de design. Baza de date folosită este PostgreSQL, aleasă pentru reputația eficienței ridicate.

Aplicația constă într-un set de clase interconectate ce au elemente grafice asociate. Scenariile principale sunt: welcome page, new user, log-in teacher, log-in student, menu teacher, menu student, courses.

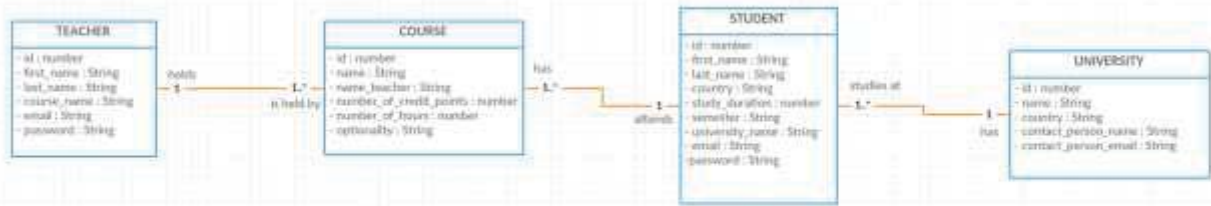


Fig. 1. Diagrama de clase

Ferestrele sunt construite într-un mod intuitiv, pentru a elimina posibile surse de ambiguitate în timpul utilizării. Fiecare câmp are asociate etichete ce specifică informația necesară, iar butoanele indică în mod clar acțiunile așteptate.

Aplicația este construită pentru a permite autentificarea mai multor categorii de utilizatori. Prima fereastră afișează cele două posibilități: profesor și student.

Structura are ca fundament o relație de tip client-server. Aplicația permite utilizatorilor să se conecteze la baza de date asociată, în mod evident prin intermediul interfeței grafice. În funcție de permisiunile individuale ale utilizatorilor, aceștia pot adăuga/modifica/șterge informații din baza de date.

Fluxul de lucru al aplicației este descris în următoarea diagramă UML (figura 2):

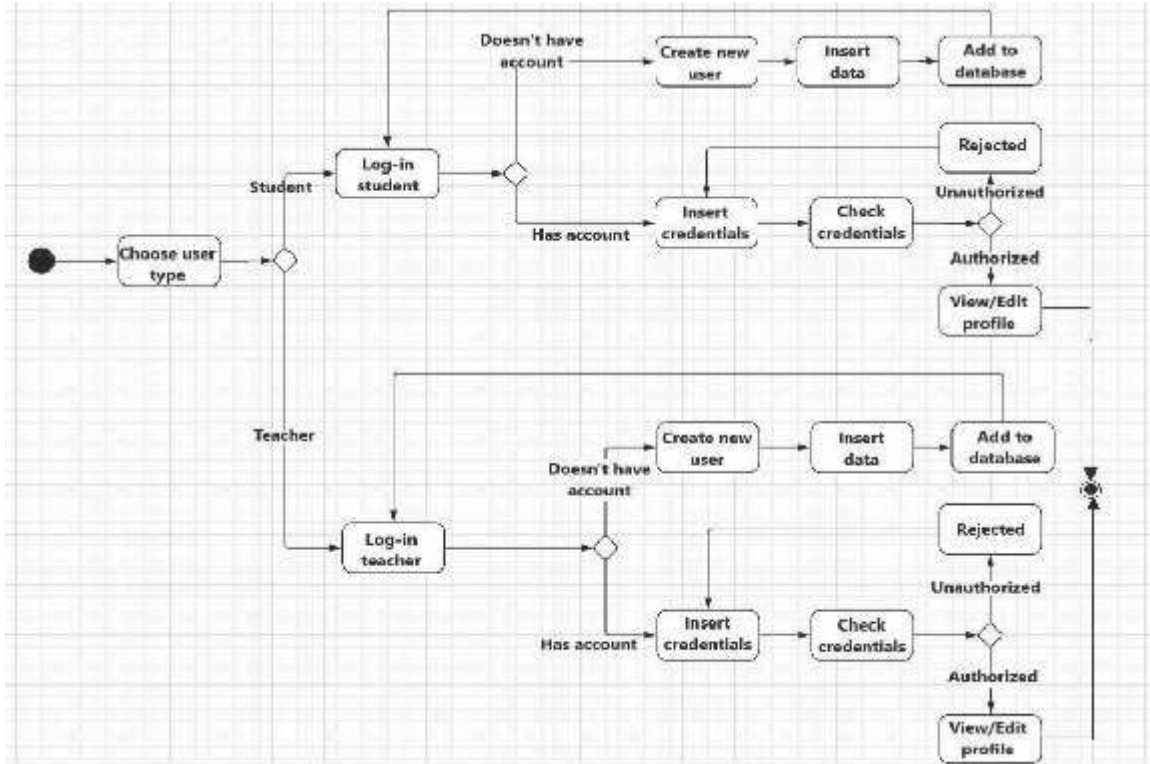


Fig. 2. Diagrama de activități, prezentând logica aplicației

La rularea programului, clasele sunt instanțiate, iar obiectele sunt create. Aceste instanțe sunt chiar ferestrele afișate. De asemenea, la rularea programului, este creat un obiect de tip QApplication.

Pagina de profil este afișată după ce un utilizator se autentifică în aplicație. Pagina conține toate informațiile introduse la momentul creării contului, împreună cu posibilitatea de a le modifica, schimbând câmpurile asociate și apăsând butonul “SAVE”. Din pagina de profil, un student poate naviga la paginile “Choose courses” și “My courses”, pentru a selecta noi cursuri la care dorește să participe, sau pentru a le vedea pe cele alese precedent.

În pagina “Courses” (figura 3), studentul poate să vadă toate cursurile disponibile, organizate într-un tabel. Poate apoi alege un curs, selectând numele cursului respectiv din drop-down-ul “Select courses” și apăsând apoi butonul “add to my courses”. În interiorul câmpului “My courses”, toate cursurile selectate anterior sunt listate, pentru ca studentul să poată să observe dacă a ales un curs din greșeală, caz în care poate selecta din drop-down numele cursului pe care nu îl dorește și să apese butonul “delete from my courses”, în vederea eliminării acestuia.

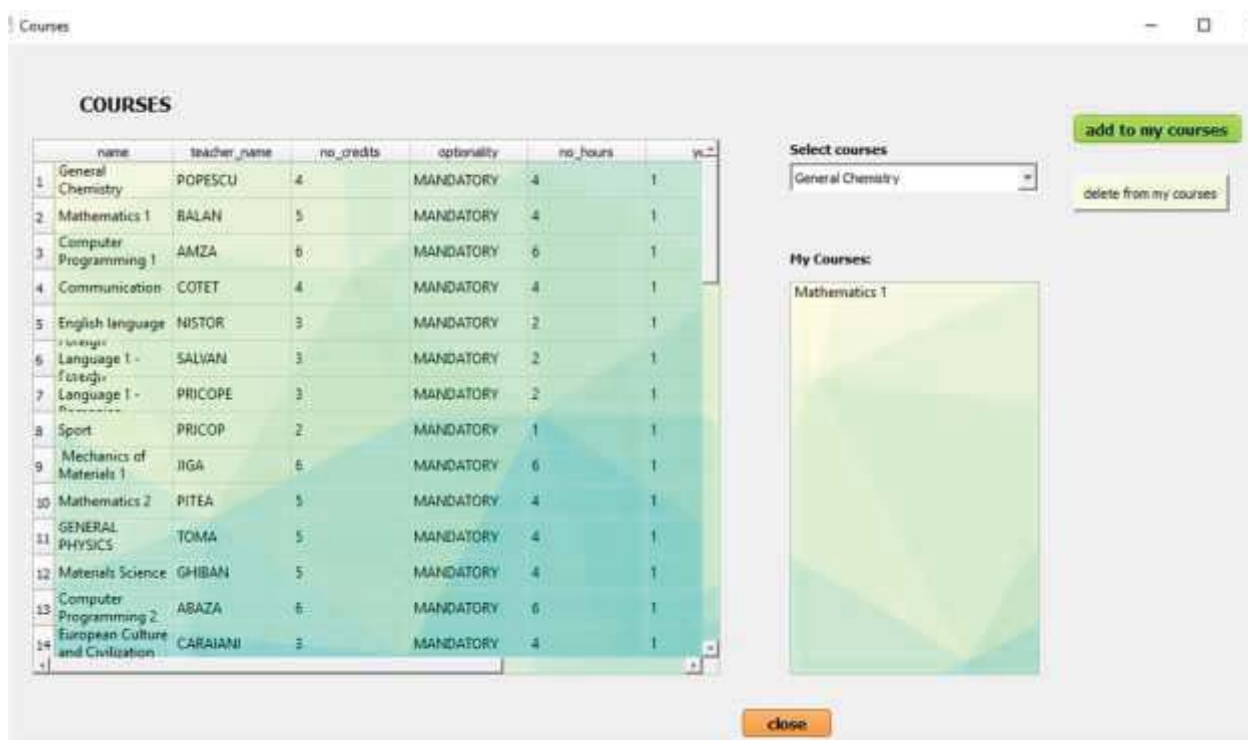


Fig. 3. Pagina de cursuri

Pagina “Details” (figura 4) afișează informații despre studenții care participă la un curs selectat. Un student poate fi căutat după nume, iar informația aferentă acestuia va fi afișată într-un tabel. Selectând un curs, și apăsând butonul “View course students”, toți studenții care participă la acel curs sunt afișați în tabel, împreună cu numărul de studenți, în câmpul “Nr students”. Un student poate fi selectat cu dublu click pe email-ul lui din tabel. După selectarea studentului, apăsând butonul “View Selected Student Details”, vor fi afișate informațiile despre toate cursurile la care acesta participă.

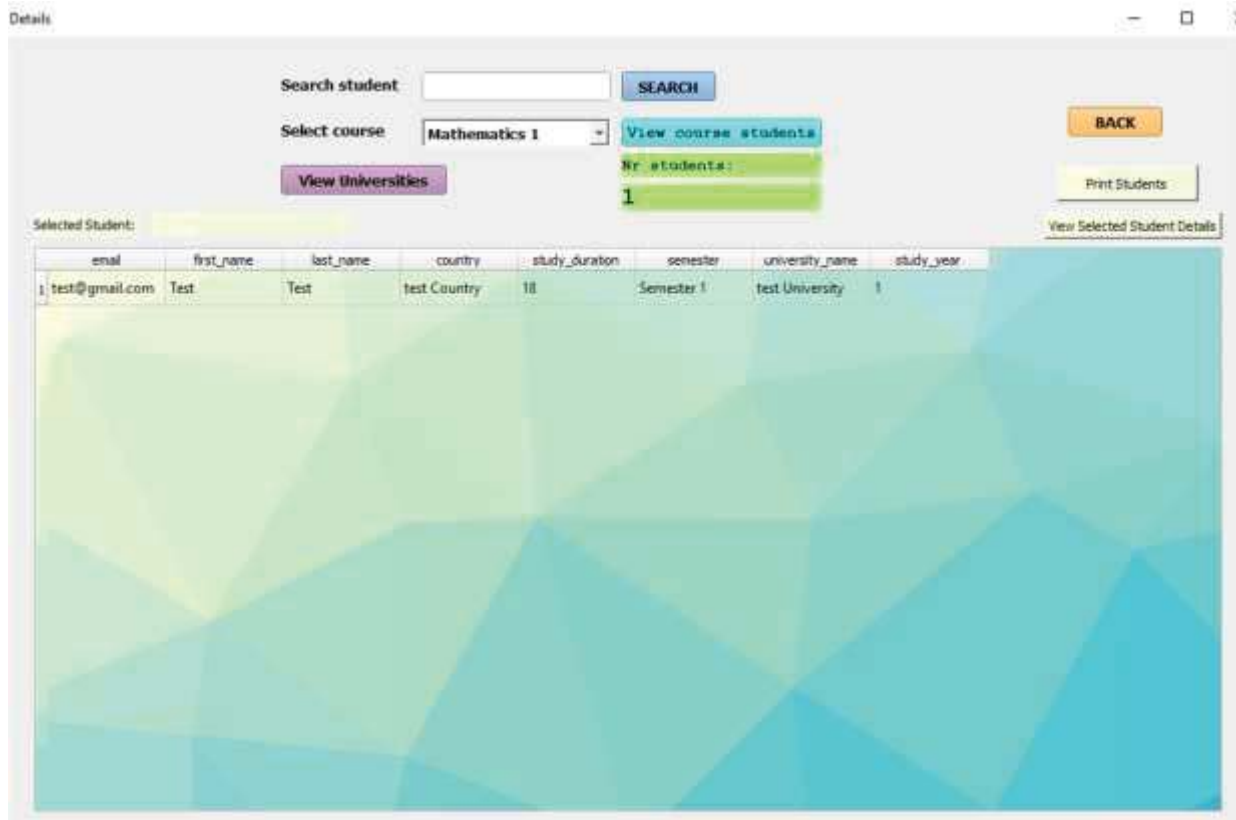


Fig. 4. Pagina de detalii

4. Implementarea în Java. Obiective. Analiză comparativă cu soluția precedentă

În urma analizei limitărilor aplicației precedente, s-au identificat 3 soluții:

1. Dezvoltarea a încă două versiuni descărcabile de pe imst.pub.ro, una pentru MAC OS și respectiv una pentru sistemul de operare Linux.
2. Dezvoltarea unei interfețe PHP care să comunice cu aplicația prin intermediul liniei de comandă.
3. Dezvoltarea unei aplicații de tip web (peste Java), care să poată fi integrată direct pe un subdomeniu al imst.pub.ro.

Analizând argumentele pro și contra, respectiv resursele echipei de dezvoltare, s-a ales soluția numărul 3. Astfel, aplicația și-a modificat specificațiile tehnice (funcționalitățile oferite rămânând aceleași, după cum urmează):

Tabelul 1. Tehnologiile folosite

CERINȚE	RĂSPUNS ACTUAL	RĂSPUNS INIȚIAL
Tip bază de date	PostgreSQL	PostgreSQL
Limbaj de programare	JAVA Open JDK 1.8	QtCreator
Framework	Spring și SpringBoot	QT
Front-end	Thymeleaf	QtCreator

Tehnologiile folosite în dezvoltarea noii aplicații au fost alese datorită unor beneficii evidente, pentru a depăși limitările întâlnite în prima aplicație:

- Securitatea precară a fost îmbunătățită în noua variantă prin intermediul frameworkului folosit, Spring.

Astfel, tot procesul de autentificare, împreună cu limitarea accesului la datele aplicației sunt securizate folosind Spring Security.

- Accesibilitatea WEB, inițial inexistentă, este acum modul principal de utilizare al aplicației. Legătura cu partea de WEB este realizată prin intermediul Thymleaf + Spring.

- Alegerea acestei abordări rezolvă totodată și problema accesării aplicației de pe dispozitive mobile, singura condiție ce trebuie îndeplinită de un astfel de dispozitiv fiind conexiunea la internet și existența unui browser. De asemenea, au fost depășite în totalitate limitările de design ale aplicației precedente.

Odată cu necesitatea lansării în mediul Online, a apărut și necesitatea eliminării primei pagini, care oferea cele două posibilități: student și profesor. Scopul aplicației este înregistrarea studenților de la universități străine, astfel s-a eliminat posibilitatea ca și profesorii din UPB să își poată face cont.

Aceștia vor avea credentiale prestabilite, create în prealabil de către administratorul sistemului. De asemenea, la câmpurile din baza de date s-a mai adăugat o coloană, odată cu apariția posibilității de a încărca fișiere de tip text, sau multimedia, precum o fotografie, scanări ale documentelor personale, certificate de limbă, eseu motivațional etc.

La accesarea paginii web aferente aplicației se deschide în fereastra browserului pagina de autentificare, comună studentului și profesorului. În continuare, un student își poate crea un cont, sau se poate autentifica. După autentificare, în partea dreaptă se afișează un meniu cu fiecare secțiune ce trebuie completată (date personale, certificări de limbă etc.). Ultima pagină din meniul lateral o constituie trimiterea aplicației, care nu este activă până când toate câmpurile din secțiunile precedente au fost completate.

5. Concluzii

Rezultatul proiectului este o aplicație utilizabilă, funcțională, ce facilitează managementul datelor studenților ce beneficiază de bursă Erasmus, înlocuind necesitatea gestiunii manuale a înregistrărilor. De asemenea, oferă accesibilitate la informații pentru un număr mare de indivizi și reduce riscul erorii umane.

În plus, proiectul subliniază utilitatea propriu-zisă a programării orientate pe obiect în lumea reală și este un exemplu coerent pentru existența unei probleme ce poate fi rezolvată prin intermediul unei metode bazate pe programarea orientate pe obiect.

Prin dezvoltarea celei de a doua aplicații, bazată pe WEB, soluția oferită este concepută având în vedere scalabilitatea. Astfel, posibilitatea extinderii aplicației la nivel național sau european este un proces în totalitate fezabil.

6. Bibliografie

[1]. Mihai Caramihai, 2018, OBJECT ORIENTED PROGRAMMING COURSE

[2]. QT Documentation Retrieved from <http://doc.qt.io/>

[3]. 1.2. Architectural Fundamentals – Chapter 1. Getting Started Retrieved from <https://www.postgresql.org/docs/10/tutorial-arch.html>