



Card Studentesc

Proiectarea sistemelor de baze de date - Proiect

Profesor coordonator: Asist. drd. ing. Cătălin MIRONEANU

Alexandru GRIGORAȘ

Ioan SUFLET

1407B

1. Introducere

1.1. Descriere

Cardul studentesc este asemănător cu cel bancar care cuprinde toate metodele de plată, abonamentele, documentele și informațiile unui student. Acesta poate fi folosit în interiorul unui campus universitar pentru a efectua tranzacții la companiile afiliate, depune numerar, închiria, stoca formulare, dar și în exterior, la magazinele partenere.



1.2. Motivația alegerii temei

Pe parcursul anilor de studenție s-a observat o creștere în volumul de acte, documente, chitanțe, abonamente și adeverințe pe care un student trebuie să le prezinte în diverse situații. În cele mai multe cazuri se consumă timp inutil pentru multiplicare sau procurare de adeverințe. Totodată, în proximitatea unui campus studentesc se află numeroase magazine dedicate tinerilor, majoritatea acceptând plățile cu cardul. Având în vedere aceste aspecte, se pot scurta procesele unui student de a cumpăra, transfera bani, înnoi abonamente sau procura documente prin introducerea unui card care să înglobeze toate acestea și să fie folosite la momentul potrivit.

1.3. Descrierea aplicației

Aplicația este organizată sub forma unui Site Web. Aceasta permite crearea unui card virtual al studentului prin care acesta poate efectua următoarele operații:

- Vizualizarea contului curent;
- Vizualizarea istoricului tranzacțiilor;
- Efectuarea tranzacțiilor: cumpărare, vânzare și achiziționare abonament.

Utilizatorul (studentul aparținând unei facultăți acreditate) va putea interacționa cu aplicația printr-o platformă web interactivă optimizată pentru toate tipurile de device-uri (telefon, tabletă, laptop sau desktop pc).

Informațiile despre student, facultăți, universități, carduri, locații și tranzacții vor fi stocate într-o bază de date Oracle. Comunicarea cu aplicația web se va efectua prin serviciile REST (GET, POST, PUT, DELETE), răspunsul primit fiind în format JSON.

În interfața grafică, utilizatorul va putea să își creeze un cont unic bazat pe adresa de email și pe numărul de telefon. După ce studentul se va autentifica acesta va putea să își vizualizeze istoricul tranzacțiilor, posibilitățile de vânzare, cumpărare și achiziționare abonamentele.

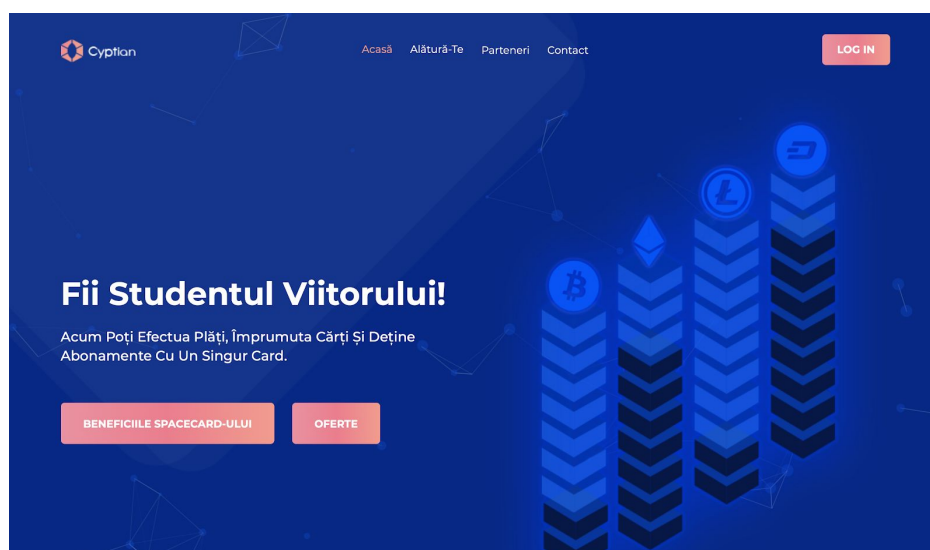
1.4. Accesul la aplicație al utilizatorilor

În aplicație există 2 tipuri de utilizatori (administrator și student) ce au acces diferențiat:

1. Administratorul are acces deplin la aplicație și poate să adauge, modifice și elimine datele, cum ar fi Universitățile, Facultățile, Studenții, Locațiile, dar și să modifice datele de pe carduri sau tranzacții în caz de erori (are acces direct la baza de date prin platforma APEX);
2. Studentul are acces restricționat. El poate să adauge, modifice și elimine informații personale, card-ul și doar să adauge tranzacții (abonament, cumpărare, vânzare) (are acces la baza de date doar prin serviciile REST puse la dispoziție de platforma APEX prin interfața grafică a aplicației).

2. Structura aplicației

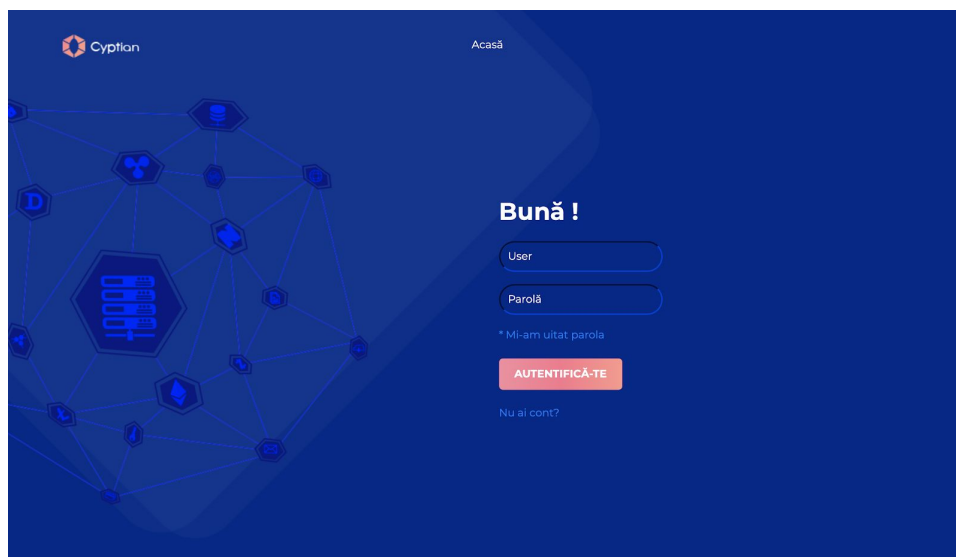
2.1. Pagina de start



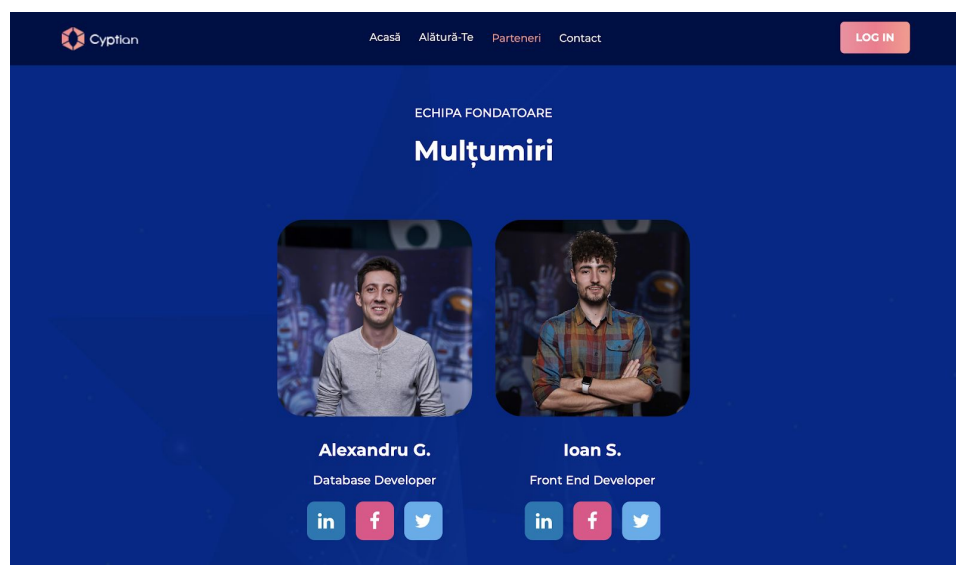
2.2. Pagina de înregistrare

The screenshot shows the registration page of the Cyprian website. The header includes the Cyprian logo and the 'Acasă' link. The main heading is 'Ne Bucurăm Să Te Avem Alături De Noi John'. Below this, a subheading reads 'Haide sa ne cunoastem'. The registration form consists of several input fields: 'Prenume:' (with 'John' entered), 'Nume:', 'User:', 'Parolă:', 'Email:', 'Numar telefon:', and 'Adresa:'. Each field has a placeholder text 'Enter [field name] Here'. At the bottom, there is a checkbox labeled 'La ce universitate studiezi, John?' with the option 'Universitatea "Alexandru Ioan Cuza"' selected.

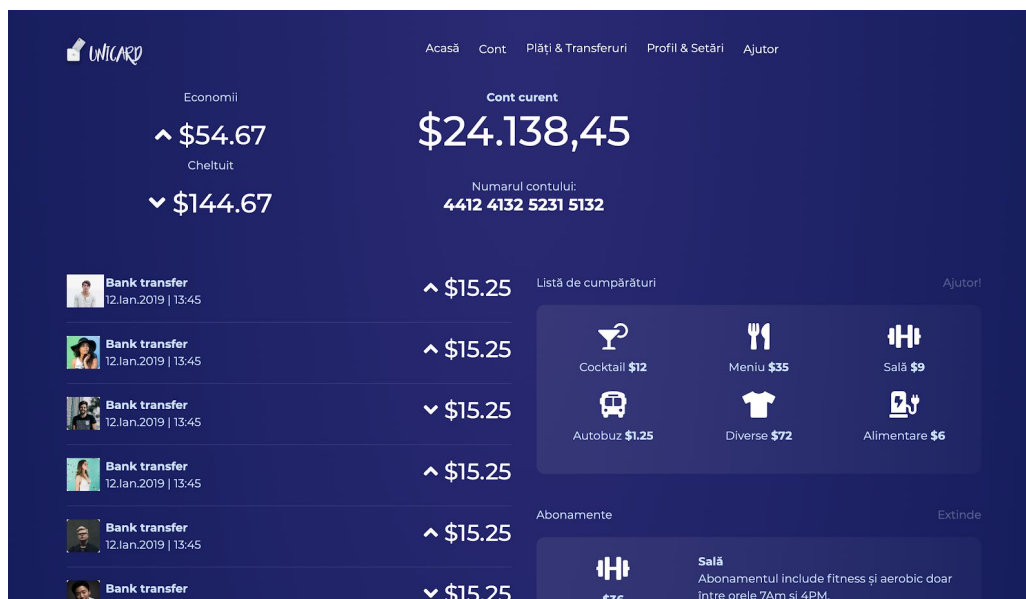
2.3. Pagina de autentificare



2.4. Pagina cu informații personale

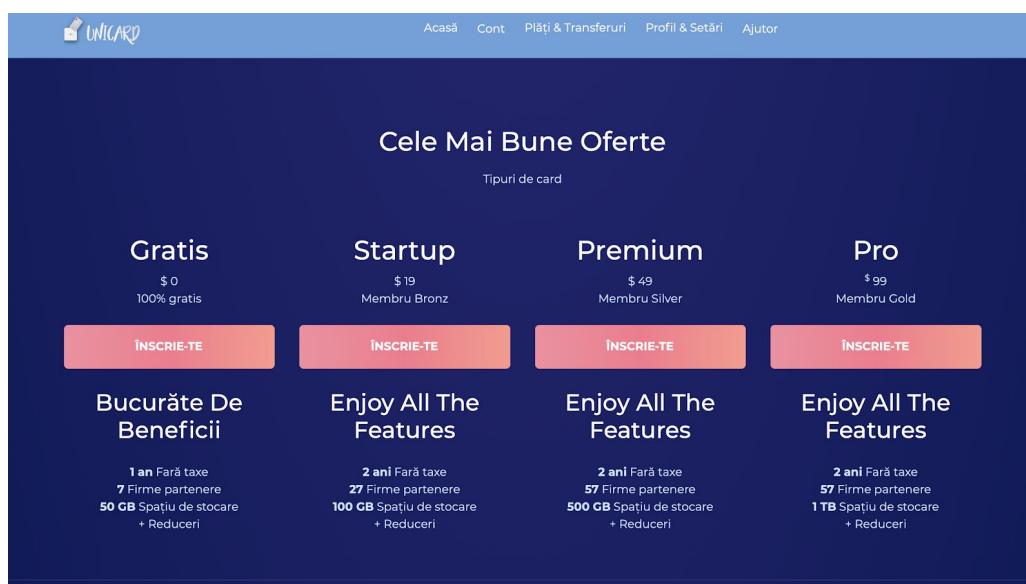


2.5. Pagina cont curent și tranzacții



2.6. Pagina de efectuare tranzacții

- Oferire la vânzare
- Cumpărare dintr-o listă de oferte
- Achiziționare / Reîncărcare abonament



3. Structura bazei de date

3.1. Descriere

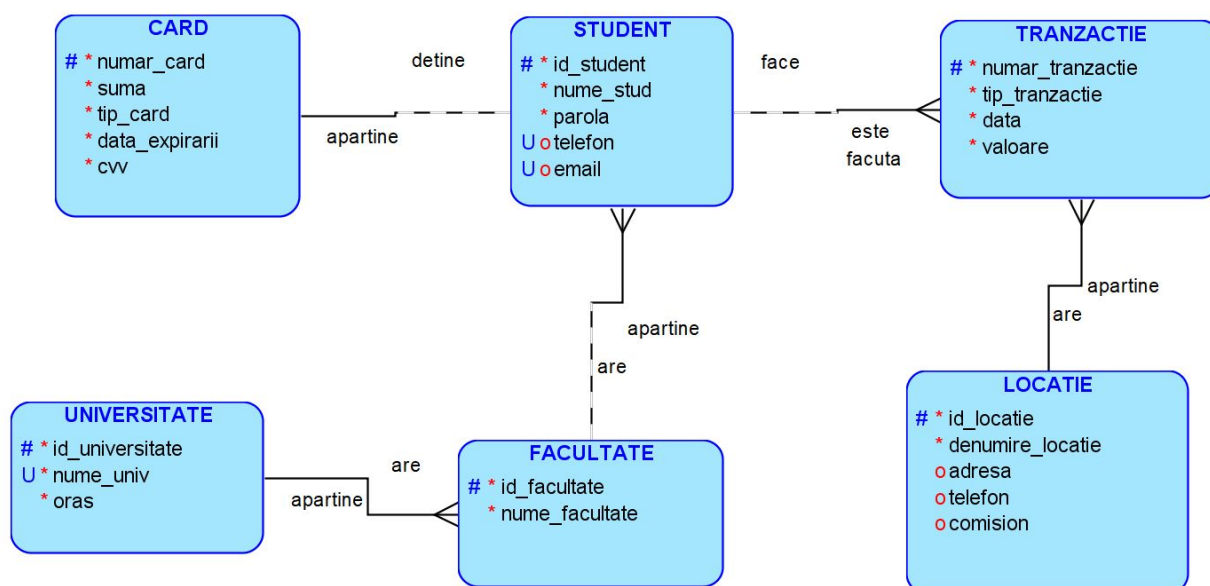
Baza de date conține următoarele tabele: CARD, STUDENT, TRANZACȚIE, UNIVERSITATE, FACULTATE, LOCAȚIE:

- **UNIVERSITATE** - înregistrările Universităților partenere: id unic ce se incrementează automat și reprezintă universitatea (id_universitate), numele universității (nume_univ) și orașul din care face parte (oras); o universitate are mai multe facultăți; constrângerile tabelii sunt: înregistrările coloanelor nu pot fi nule, id-ul este cheie primară, iar numele este unic;
- **FACULTATE** - înregistrările Facultăților aparținând unei Universități: id unic ce se incrementează automat și reprezintă facultatea (id_facultate) și numele facultății (nume_facultate); o facultate aparține unei universități și are mai mulți studenți; constrângerile sunt: înregistrările coloanelor nu pot fi nule, id-ul este cheie primară, iar coloana id_universitate este cheie străină și reprezintă id-ul universității din tabela Universitate;
- **LOCAȚIE** - înregistrările Locațiilor partenere la care se pot efectua tranzacții: id unic ce se incrementează automat și reprezintă locația (id_locatie), denumirea locației (denumire_locatie), adresa la care se află (adresa), telefonul la care poate fi contactată (telefon) și comisionul practicat (comision); o locație poate avea mai multe tranzacții efectuate; constrângerile tabelii sunt: id-ul și denumirea nu pot fi nule, comisionul trebuie să fie mai mare ca 0 și id-ul este cheie primară;
- **STUDENT** - înregistrările Studenților ce dețin un card și pot efectua tranzacții: id unic ce reprezintă studentul (id_student), numele studentului (nume_stud), telefonul la care poate fi contactat (telefon), email-ul folosit (email) și parola contului studentului (parola); un student poate aparține unei facultăți, poate avea un card și poate realiza mai multe tranzacții; constrângerile sunt: id-ul și numele nu pot fi nule, email-ul trebuie să aibă formatul corect, id-ul este cheie primară, telefonul și email-ul sunt unice, iar id_facultate este cheie

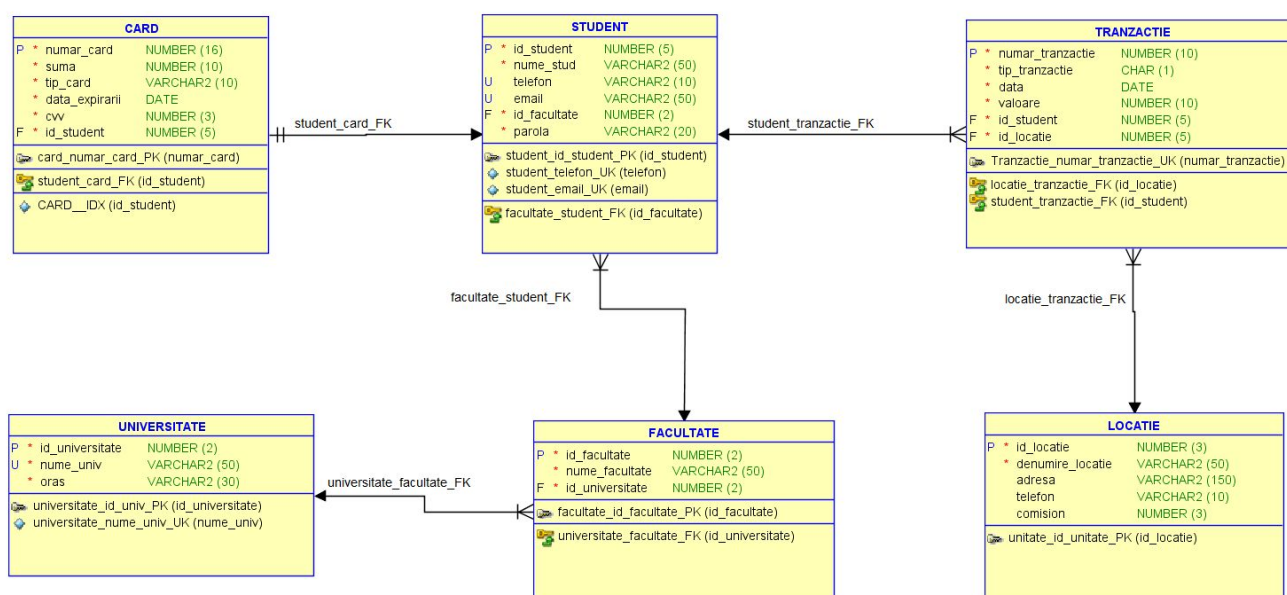
străină ce are o valoare care nu este nulă și referă coloana id_facultate din tabela facultate;

- **CARD** - înregistrările Card-urilor studenților: număr unic al card-ului (numar_card) ce este cheie primară, suma de pe acesta (suma) ce trebuie să fie mai mare sau egală ca 0, tip-ul cardului (tip_card): GENERIC, MAESTRO, MASTERCARD, VISA, data expirării acestuia (data_expirarii) ce trebuie să fie mare mare decât data curentă, numărul cvv de securitate (cvv) și id-ul studentului (id_student) ce este cheie străină și referă coloana id_student din tabela student; card-ul aparține unui singur student;
- **TRANZACȚIE** - înregistrările Tranzacțiilor studenților: numărul tranzacției efectuate (numar_tranzactie), tipul tranzacției (tip_tranzactie) ce poate fi: Abonament, Cumpărare, Vânzare, data tranzacției (data), valoarea ei (valoare) ce trebuie să fie mai mare ca 0, id-ul studentului ce o efectuează (id_student) ce este cheie străină și referă coloana id_student din tabela student și id-ul locației (id_locatie) ce este cheie străină și referă coloana id_locatie din tabela locatie; toate înregistrările tabelii trebuie să nu fie nule; o tranzacție aparține unei locații și este făcută de un student.

3.2. Schema logică a bazei de date



3.3. Schema relațională a bazei de date



3.4. Conectarea cu baza de date

3.4.1. Descriere

Baza de date este realizată în Oracle Application Express (APEX) utilizând Oracle PL/SQL.

Conectarea cu baza de date se face prin servicii REST, aplicația APEX furnizând acest serviciu. Metodele HTTP ce pot fi utilizate pentru manipularea resurselor sunt:

- GET - Preia o reprezentare a unei resurse;
- POST - Creează o nouă resursă sau adaugă o resursă într-o colecție;
- PUT - Actualizează o resursă existentă;
- DELETE - Șterge o resursă existentă.

3.4.2. Structura cererilor HTTP

- **card/**
 - GET - returnează cardurile înregistrate;
 - POST (IN: cvv, data_expirarii, id_student, numar_card, suma, tip_card; OUT: status) - adaugă un nou card;
 - DELETE (IN: nume; OUT: status) - șterge card-ul utilizatorului;
- **card/{nume}**
 - GET (IN: nume) - returnează cardul utilizatorului cu numele;
- **facultate/**
 - GET - returnează facultățile înregistrate;
 - POST (IN: nume, univ; OUT: status) - adaugă o nouă facultate;
 - PUT (IN: id, nume, univ; OUT: status) - actualizează datele unei facultăți;
- **locație/**
 - GET - returnează locațiile înregistrate;
 - POST (IN: adr, com, den_loc, tel; OUT: status) - adaugă o nouă locație;
 - PUT (IN: id_loc, adr, com, den_loc, tel; OUT: status) - actualizează datele unei locații;
- **student/**
 - POST (IN: email, nume, n_facultate, n_universitate, parola, telefon; OUT: status) - adaugă un nou student;
 - PUT (IN: id_stud, email, nume, n_facultate, n_universitate, parola, telefon; OUT: status) - actualizează datele studentului;
- **student/id/{id}**
 - GET (IN: id) - returnează studentul cu id-ul specificat;
- **student/info/**
 - GET - returnează toți studenții (pagini de 25 înregistrări);

- **student/login/**
 - GET (IN: nume, parola) - returnează mesajul de confirmare dacă combinația {nume, parolă} ale utilizatorului este validă;
- **student/nume/{nume}**
 - GET (IN: nume) - returnează studentul cu numele specificat;
 - DELETE (IN: nume; OUT: status) - elimină studentul cu numele dat;
- **tranzacție/**
 - GET - returnează toate tranzacțiile efectuate;
 - POST (IN: data_tranz, id_loc, id_stud, tip_tranz, val_tranz; OUT: status) - adaugă o nouă tranzacție și efectuează operațiile necesare;
- **tranzacție/{nume}**
 - GET (IN: nume) - returnează tranzacțiile studentului cu numele dat;
- **universitate/**
 - GET - returnează toate universitățile înregistrate;
 - POST (IN: nume, oras; OUT: status) - adaugă o nouă universitate;
 - PUT (IN: id_univ, nume, oras; OUT: status) - actualizează datele universității;
- **versiune/**
 - GET - returnează versiunea aplicației;

4. Concluzie

Cu toate că acest proiect nu are o complexitate ridicată, nu s-au întâmpinat dificultăți majore în realizarea proiectului mulțumită aplicației ORACLE Application Express (APEX) care a facilitat comunicarea cu baza de date. Ulterior răspunsul a fost prelucrat pentru a avea un aspect frumos. Din punctul de vedere al utilizatorului, aplicația web are un aspect plăcut și este simplu de folosit având în vedere elementele cu scop practic din pagină. În sine aplicația conduce clientul către pașii următori, acesta obținând orice informație cu ușurință. În cazul în care apar probleme în utilizare, există opțiunea de a ne contacta pentru a remedia situația.

Din punct de vedere al dezvoltării ulterioare, aplicația de banking are potențialul de a monitoriza acțiunile clienților pentru a face predicții în legătură cu preferințele lor în materie de servicii, îmbrăcăminte, destinații turistice, tendințe de plată a asigurărilor etc.

În concluzie, proiectul are potențial, iar acest lucru poate fi studiat pe măsura utilizării acestuia, o dovada fiind implementarea deja a sistemului de card pentru studenți pentru transportul în comun.

5. Referințe

- Datele de intrare obținute de pe <https://www.mockaroo.com/>
- Documentație servicii REST pe platforma APEX
https://www.oracle.com/webfolder/technetwork/tutorials/obe/db/apex/r51/restful_web_services/restful_web_services.html#section3
- Documentație servicii web REST
https://www.tutorialspoint.com/restful/restful_introduction.htm