CAPITOLUL 3

3.1 PROIECTAREA SISTEMULUI INFORMATIC

În acest capitol este prezentată etapa de proiectare a sistemului informatic, fiind elaborate elemente precum diagrama de clase detaliată, diagrama entitate-relație, interfețele cu utilizatorul, principalele formulare și rapoarte din cadrul acestora, dar și principalele tehnologii informatice utilizate în cadrul implementării aplicației.

3.1.1 Diagrama de clase detaliată

Am completat diagrama de clase prezentată în capitolul anterior prin adăugarea atributelor și metodelor specifice, dar și a tipului acestora, așa cum este reprezentat în *Figura 3.1*. Astfel, am reprezentat structura statică a sistemului, a claselor, a relațiilor dintre clase și a structurilor de moștenire.

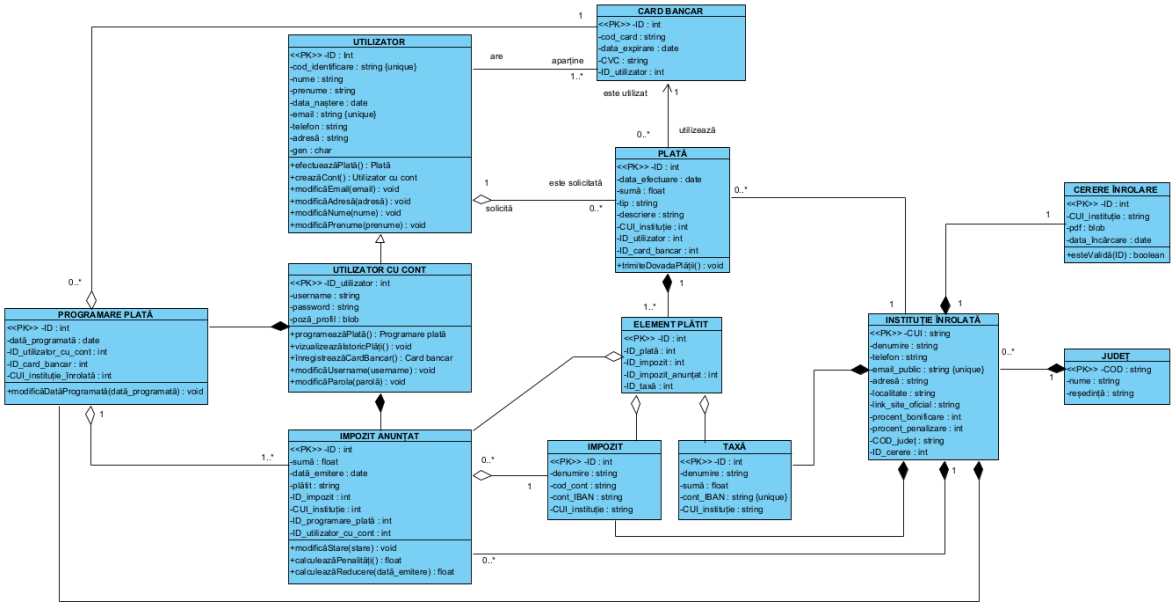

Figura 3.1 - Diagrama de clase detaliată

Spre exemplu, clasa ***Utilizator*** conține detalii despre utilizatorii care efectuează plăți în cadrul aplicației, precum: cod_identificare (CNP sau NIF în cazul cetățenilor străini), nume, prenume, *email*, telefon, adresă, data naștere sau gen. De asemenea, aceasta mai conține metode specifice rolului de utilizator precum metoda extrageDatăNaștere() care primește drept parametru CNP/NIF-ul utilizatorului și returnează data nașterii acestuia, sau metode de modificare a datelor personale furnizate, ca de exemplu: modificăEmail() sau modificăAdresă().

De asemenea este reprezentată relația de moștenire dintre clasa ***Utilizator*** și clasa ***Utilizator cu cont***, cea din urmă moștenind toate atributele și metodele clasei părinte. Pe lângă acestea, clasa derivată conține atribute și metode specifice ei precum: *password*, *username* sau vizualizeazăIstoricPlăți(), metodă de tip void ce replică una dintre opțiunile caracteristice utilizatorilor ce aleg să își deschidă cont, adică vizualizarea istoricului plăților efectiuate.

3.1.2 Proiectarea bazei de date

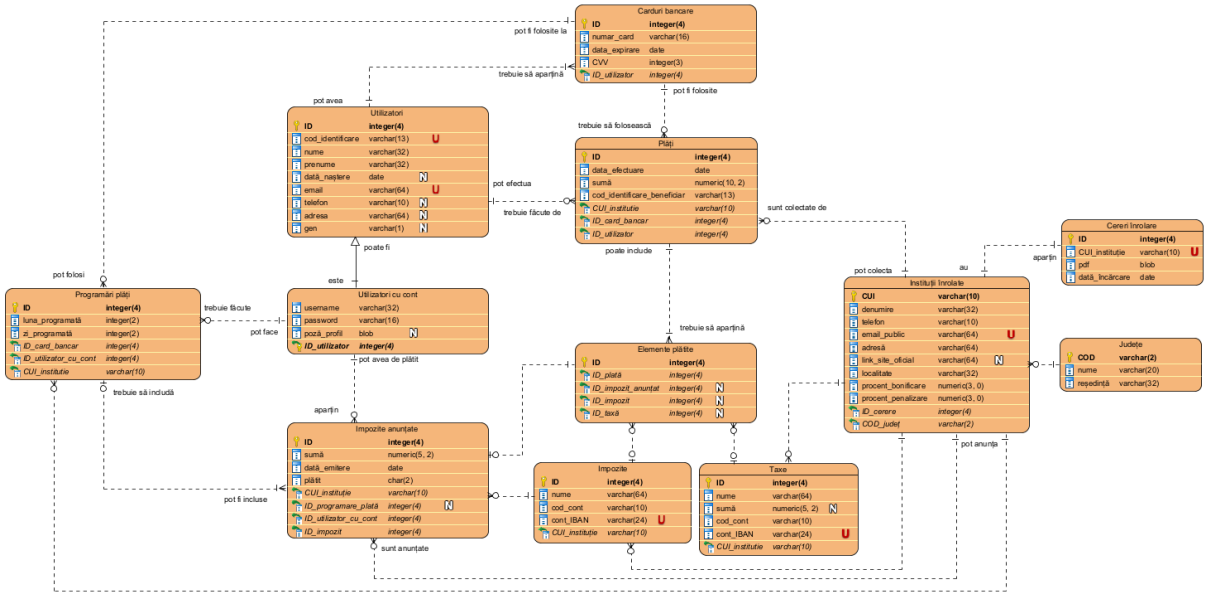

Figura 3.2 – Diagrama entitate-relație

TABELA UTILIZATORI:

Entitatea ***Utilizatori*** reprezintă un element esențial al aplicației și stochează în mod persistent informații despre utilizatorii care efectuează plăți prin intermediul acesteia. Această entitate este compusă dintr-o serie de atribute care descriu detaliile utilizatorilor. Atributele incluse în entitatea ***Utilizatori*** sunt următoarele: ID, care servește drept cheie primară a tabelei, cod_identificare (care poate fi CNP pentru utilizatorii cu cetățenie română sau NIF pentru străini), nume, prenume, *email*, adresă, gen, telefon și dată_naștere (de tip *Date*). Aceste atribute permit stocarea informațiilor relevante despre utilizatori, precum numele complet, adresa de *email*, adresa domiciliului, numărul de telefon și data nașterii.

Tabela ***Utilizatori*** este conectată printr-o relație de tip unu-la-mulți (1:M) cu tabela ***Carduri bancare***. Aceasta înseamnă că un utilizator fără cont poate utiliza de-a lungul timpului carduri bancare diferite pentru efectuarea plăților. Această relație permite gestionarea și asocierea cardurilor bancare specifice utilizatorilor. De asemenea, tabela ***Utilizatori*** este conectată printr-o relație de tip unu-la-mulți (1:M) cu tabela ***Plăți***. Această relație reflectă faptul că un utilizator poate

efectua mai multe plăți de-a lungul timpului și că aceste plăți sunt înregistrate și asociate cu utilizatorul corespunzător. În plus, tabela ***Utilizatori*** este legată, prin intermediul unei relații de specializare, de tabela ***Utilizatori cu cont***, care reprezintă un subtip al acesteia. Această relație de specializare permite gestionarea detaliilor specifice utilizatorilor cu cont, în timp ce restul atributelor rămân comune pentru ambele tabele.

Pentru a asigura integritatea datelor, restricțiile sunt aplicate asupra atributului ID, care este cheie primară a tabelei și asupra atributelor cod_identificare și *email*, care trebuie să fie unice. Atributele gen, adresă și telefon, sunt marcate ca *Nullable*, permițând absența unei valori în cadrul unei înregistrări.

TABELA UTILIZATORI CU CONT:

Utilizatori cu cont reprezintă entitatea care descrie utilizatorii aplicației *web* care optează să-și creeze un cont personal. Această tabelă conține informații specifice acestor utilizatori și este conectată la tabela părinte ***Utilizatori*** prin intermediul atributului ID_utilizator, care servește drept cheie primară și indică relația cu tabela părinte.

Atributele stocate în tabela ***Utilizatori cu cont*** includ *username* și *password*, care rețin informațiile de autentificare ale utilizatorilor. De asemenea, există și atributul poză_profil, de tip BLOB, care permite utilizatorilor să încarce o imagine de profil. În ceea ce privește restricțiile, atributul ID_utilizator are restricția de cheie primară (*Primary Key*), asigurând unicitatea valorilor. Atributul poză_profil este marcat ca *Nullable*, permițând absența unei imagini de profil, caz în care va fi afișată imaginea de profil *default*.

Specializarea tabelei ***Utilizatori cu cont*** se evidențiază prin relația sa cu tabela părinte ***Utilizatori***, indicând faptul că reprezintă un subtip specific al utilizatorilor. De asemenea, tabela este conectată printr-o relație de unu-la-mulți (1:M) cu tabela ***Programări plăți***, ceea ce permite utilizatorilor cu cont să creeze una sau mai multe plăți programate. De asemenea, există o relație de unu-la-mulți (1:M) cu tabela ***Impozite anunțate***, prin care utilizatorilor cu cont le sunt afișate obligațiile de plată direct în pagina personală.

TABELA CARDURI BANCARE:

Carduri bancare reprezintă entitatea în care sunt stocate informațiile despre cardurile utilizate pentru efectuarea plăților în cadrul aplicației *web*. Această tabelă conține următoarele atribute: ID (cheie primară) cu restricția de *Primary Key*, cod_card și CVC, care oferă detalii despre cardul bancar, precum și atributul dată_expirare (tip *Date*), care reține data expirării cardului.

Tabela ***Carduri bancare*** are trei relații: o relație de unu-la-mulți (1:M) cu tabela ***Plăți***, deoarece același card bancar poate fi utilizat pentru mai multe plăți distincte; o relație de unu-la-

mulți (1:M) cu tabela ***Programări plăți***, deoarece pot exista mai multe programări de plăți care folosesc același card bancar pentru realizarea efectivă a plății; o relație de mulți-la-unu (M:1) cu tabela ***Utilizatori***. Atributul specific al tabelei este ID_utilizator, care are restricția de cheie externă (*Foreign Key*) și este utilizat pentru a marca legătura cu atributul ID din tabela ***Utilizatori***. În cazul în care o instanță din tabela ***Utilizatori*** este ștearsă, această restricție permite ștergerea corespunzătoare a instanței asociate din tabela ***Carduri bancare***, conform principiului *on delete cascade* (ștergere în cascadă).

TABELA PLĂȚI:

Plăți reprezintă entitatea în care sunt stocate informațiile despre plățile efectuate prin intermediul aplicației *web*. Această tabelă conține următoarele atribute: ID (cheie primară), dată_efectuare (de tip *Date*), sumă (de tip numeric) și cod_identificare_beneficiar.

Tabela ***Plăți*** are trei relații: o relație de mulți-la-unu (M:1) cu tabela ***Carduri bancare***, deoarece o plată este efectuată cu un singur card bancar într-un moment dat, dar același card poate fi utilizat pentru mai multe plăți în decursul timpului; o relație de mulți-la-unu (M:1) cu tabela ***Utilizatori***, deoarece un utilizator poate efectua mai multe plăți, dar o plată este realizată de către un singur utilizator, indiferent dacă are sau nu cont în aplicație; și o relație de mulți-la-unu (M:1) cu tabela ***Instituții publice***, deoarece o plată are o singură instituție publică destinatară a taxelor, iar o instituție publică poate colecta una sau mai multe plăți.

De asemenea, tabela conține atributele CUI_instituție, ID_utilizator și ID_card_bancar, toate având restricția de cheie externă (*Foreign Key),* referindu-se la atributul CUI din tabela ***Instituții publice***, atributul ID din tabela ***Utilizatori*** și respective la atributul ID din tabela ***Carduri bancare***.

TABELA INSTITUȚII ÎNROLATE:

Instituții înrolate este entitatea care reprezintă instituțiile publice colectoare de taxe înrolate în aplicația *web*. Tabela conține următoarele atribute: CUI (cheie primară), denumire, telefon, *email_public*, adresă, link_site_oficial și localitate, reprezentând detalii și informații de contact despre instituția publică, dar și procent_bonificare și procent_penalizare, reprezentând detalii specifice fiecărei instituții cu privire la aplicarea bonificației și/sau a penalizării asupra sumei impozitelor de colectat.

Din perspectiva restricțiilor, atributul *email_public* are restricția *Unique*, ceea ce înseamnă că fiecare instituție publică trebuie să aibă un *email* unic. De asemenea, atributelor cod_județ și ID_cerere li se aplică restricții de referințialitate *Foreign Key*. Tabela ***Instituții publice*** are o relație de unu-la-unu (1:1) cu tabela ***Cereri înrolare***, deoarece o instituție este înrolată o singură dată în sistem și necesită o cerere completată și validată pentru înrolare. De asemenea, există o relație de

mulți-la-unu (M:1) cu tabela ***Județe***, deoarece mai multe instituții publice pot aparține aceluiași județ. O altă relație de tipul unu-la-mulți (1:M) este cu tabela ***Plăți***, întrucât o instituție publică colectoare de taxe poate colecta mai multe plăți în același timp. Nu în ultimul rând, se remarcă legătura cu tabela ***Impozite anunțate***, de tipul unu-la-mulți (1:M), deoarece o instituție publică poate informa mai mulți utilizatori despre obligațiile de plată pe care le au.

TABELA CERERI ÎNROLARE:

Cereri înrolare este entitatea în care se stochează cererile de înrolare încărcate de către instituțiile publice colectoare de taxe pentru a fi înrolate în aplicația *web*. Această tabelă conține următoarele atribute: ID (cheie primară), CUI_instituție, pdf (de tip BLOB) în care se păstrează cererea în format electronic PDF și dată_încărcare (de tip *Date*) care reprezintă data la care cererea a fost încărcată în sistem. Tabela ***Cereri înrolare*** are o relație de unu-la-unu (1:1) cu tabela ***Instituții publice***, ceea ce reprezintă faptul că fiecare cerere de înrolare este asociată cu o singură instituție publică.

TABELA JUDEȚE:

Județe reprezintă entitatea în care sunt stocate informații despre județele din România. Aceasta este compusă din trei atribute, cod, nume și reședință, unde cod servește drept cheie primară a tabelei. Tabela ***Județe*** este legată de tabela **Instituții înrolate** printr-o relație de tip unu-la-mulți (1:M), deoarece într-un județ pot exista mai multe instituții publice colectoare de taxe. Această legătură permite asocierea instituțiilor cu județul din care fac parte, asigurând astfel o reprezentare corectă a ierarhiei teritoriale.

TABELA ELEMENTE PLĂTITE:

Elemente plătite reprezintă o tabelă intermediară care servește drept punte între tabelele ***Plăți***, ***Impozite***, ***Taxe*** și ***Impozite anunțate***. Această entitate are rolul de a lega taxele sau impozitele implicate într-o plată.

Tabela ***Elemente plătite*** este compusă din atributul ID, care reprezintă cheia primară a tabelei, și din următoarele atribute: ID_plată, ID_impozit, ID_impozit_anunțat și ID_taxă. Aceste atribute sunt restricționate prin chei externe (*Foreign Key*), deoarece fac referire la tabelele cu care ***Elemente plătite*** are relații de legătură. Astfel, există o relație de mulți-la-unu (M:1) între această tabelă și tabela ***Impozite***, o relație similară între ***Elemente plătite*** și tabela ***Taxe***, iar cu tabela ***Impozite anunțate*** există o relație de unu-la-unu (1:1).

Prin intermediul tabelei ***Elemente plătite***, se realizează o asociere corespunzătoare între plăți și taxe/impozite, facilitând astfel gestionarea și identificarea elementelor specifice fiecărei plăți efectuate. Această structură de bază asigură coerența și integritatea datelor în cadrul

sistemului, permițând analiza și raportarea corectă a informațiilor referitoare la plățile realizate și impozitele/taxele asociate acestora.

TABELA IMPOZITE:

Impozite este entitatea care stochează tipurile de impozite pe care instituțiile înrolate le pot colecta. Aceasta este definită de atributele nume, cod_cont și cont_IBAN, care reprezintă denumirea completă a impozitului și date despre contul fiscal asociat colectării banilor pentru respectivul impozit. Cheia primară a tabelei este atributul ID.

Tabela ***Impozite*** are o relație de unu-la-mulți (1:M) cu tabela ***Elemente plătite***, semnificând că fiecare înregistrare în tabela ***Impozite*** este asociată cu unul sau mai multe elemente plătite. Această relație permite legarea impozitelor de plăți specifice într-un mod precis și coerent.

De asemenea, există și o relație de unu-la-unu (1:1) între această tabelă și tabela ***Impozite anunțate***. Aceasta indică faptul că o înregistrare de tip impozit din tabela ***Impozite*** poate fi asociată cu un singur impozit anunțat, din cadrul tabelei ***Impozite anunțate***. Această relație facilitează gestiunea și urmărirea impozitelor anunțate și a modului în care acestea sunt atribuite și utilizate în plăți.

TABELA TAXE:

Entitatea ***Taxe*** stochează informații despre tipurile de taxe pe care instituțiile înrolate le pot colecta. Tabela este compusă din următoarele atribute: ID, reprezentând cheia primară a tabelei, nume, cod_cont, cont_IBAN și sumă de tip numeric. Atributele nume și sumă descriu numele taxei și respective suma predefinită de plată asociată acesteia. Atributul sumă are restricția *Nullable*, indicând faptul că este opțional.

Există o relație de tip unu-la-mulți (1:M) între tabela ***Taxe*** și tabela ***Elemente plătite***, deoarece o taxă poate fi unul dintre elementele componente ale unei plăți. Această relație facilitează asocierea taxelor specifice unei plăți și urmărirea lor în cadrul aplicației.

TABELA PROGRAMĂRI PLĂȚI:

Entitatea denumită ***Programări plăți*** stochează informații despre plățile programate pentru a fi achitate la o dată prestabilită de către utilizatorii care au un cont în aplicație. Tabelul conține următoarele atribute: ID reprezentând cheia primară, lună și zi, care indică data la care plata trebuie să fie executată automat, ID_utilizator_cu_cont, ID_card_bancar și CUI_instituție care sunt restricționate de referențialitate de tip *Foreign Key*.

Astfel, există legătura de mulți-la-unu (M:1) între tabela ***Programări plăți*** și tabela ***Utilizatori cu cont***, deoarece un utilizator cu cont poate programa una sau mai multe plăți care să fie efectuate automat în cadrul aplicației. De asemenea, există o relație de unu-la-mulți (1:M) între

tabela ***Programări plăți*** și tabela ***Impozite anunțate***, indicând faptul că o programare poate include cel puțin un impozit dintre impozitele anunțate. O altă legătură este cu tabela ***Instituții înrolate***, iar în final, există o legătură de mulți-la-unu (M:1) între tabela ***Programări plăți*** și tabela ***Carduri bancare***, deoarece o plată programată poate fi asociată cu un singur card bancar pentru efectuarea plății efective.

TABELA IMPOZITE ANUNȚATE:

Impozite anunțate reprezintă entitatea care stochează informații despre obligațiile de plată sub formă de impozite, pe care utilizatorii cu cont în aplicația *web* trebuie să le achite. Aceste impozite sunt anunțate direct de către instituțiile publice colectoare de taxe înrolate în aplicație.

Tabela ***Impozite anunțate*** este compusă din următoarele atribute: ID, reprezentând cheia primară a tabelei, sumă de tip numeric, reprezentând suma care trebuie plătită, data_emitere, de tip *Date*, reprezentând data la care acestea au fost introduse în baza de date. De asemenea, tabela conține și atributele plătit, care poate lua valoarea de 'DA' sau 'NU'.

Există următoarele relații între tabela ***Impozite anunțate*** și alte tabele: o relație de mulți-la-unu (M:1) cu tabela ***Utilizatori cu cont***, indicând faptul că un utilizator cu cont poate fi înștiințat cu privire la una sau mai multe impozite pe care are obligația de a le plăti; o relație de mulți-la-unu (M:1) cu tabela ***Programări plăți***, evidențiind faptul că mai multe impozite anunțate pot fi incluse într-o singură programare de plată; o relație de mulți-la-unu (M:1) cu tabela ***Impozite***, reflectând faptul că mai multe impozite anunțate pot fi asociate cu același tip de impozit; o relație de unu-la-unu (1:1) cu tabela ***Elemente plătite***, semnificând că o înregistrare de impozit anunțat poate corespunde cu o singură înregistrare; o relație de mulți-la-unu (M:1) cu tabela ***Instituții publice***, indicând că mai multe impozite anunțate pot fi asociate cu aceeași instituție publică colectoare de taxe.

3.2 PROIECTAREA INTERFEȚEI CU UTILIZATORUL

Harta de navigare constituie o prezentare de ansamblu a structurii aplicației *web* din perspectiva interfeței cu utilizarorul (UI), evidențiind componentele sale și relațiile dintre acestea, precum și ierarhia și fluxul de navigare disponibil. Pagina principală constituie prima interacțiune a utilizatorilor cu aplicația *web*, servind drept punct de plecare către paginile de autentificare, înregistrare, efectuare a plăților și de prezentare a instituțiilor, respectiv a informațiilor legislative. Astfel, prin intermediul acestei scheme structurate, se facilitează o experiență coerentă și fluidă, permitând utilizatorilor să acceseze cu ușurință informațiile și funcționalitățile necesare în cadrul aplicației *web*.

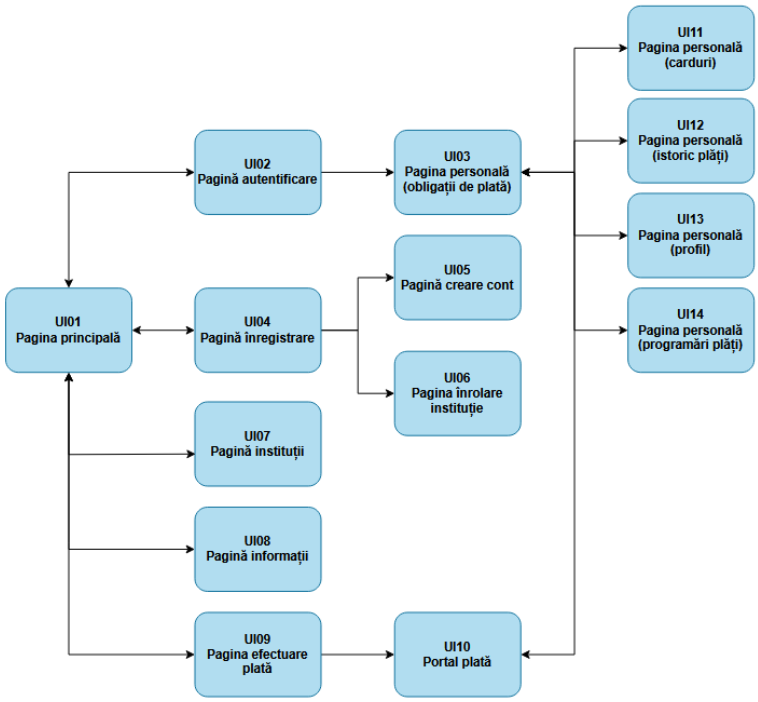

Figura 3.3 - Harta de navigare a aplicației web

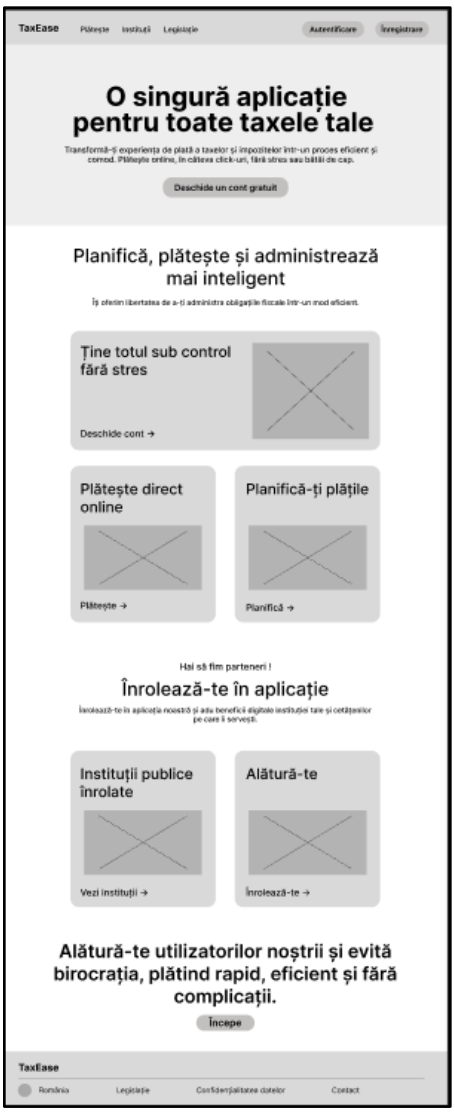

Figura 3.4 – Interfața cu utilizatorul: Pagina principală (UI01)

Pagina principală reprezintă prima interacțiune a unui utilizator cu aplicația *web*. Aceasta are rolul de a oferi utilizatorilor o primă impresie și un punct de plecare în explorarea și utilizarea aplicației. Astfel, prin pagina principală sunt furnizate informații relevante și esențiale despre aplicație, precum funcționalități, caracteristici și servicii disponibile. De asemenea, aceasta servește drept un punct central de pornire pentru navigarea în aplicație, utilizatorii putând accesa de aici pagini și funcționalități cheie care facilitează orientarea și accesul la conținutul dorit. Nu în ultimul rând, pagina principală contribuie la crearea unei experiențe intuitive pentru utilizatori, promovând într-un mod atractiv din punct de vedere vizual conținut relevant sau informații importante și stimulând participarea și interacțiunea utilizatorului cu aplicația.

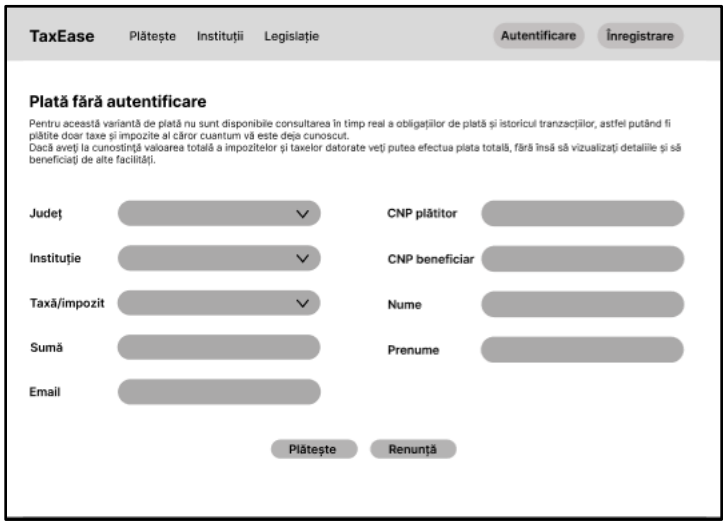

Figura 3.5 - Interfața cu utilizatorul: Pagină efectuare plată (UI09)

Pagina de efectuare a unei plăți este specifică plăților fără autentificare, adică acele plăți care vizează obligațiile de plată unde este necesară cunoașterea sumei datorate în prealabil. Astfel, această pagină este dedicată utilizatorilor fără cont sau utilizatorilor cu cont care doresc să plătească în numele altei persoane fizice. În acest formular, utilizatorul va trebui să introducă manual datele cerute, relevante procesului de plată, urmând să fie direcționat către un portal de plăți pentru finalizare.

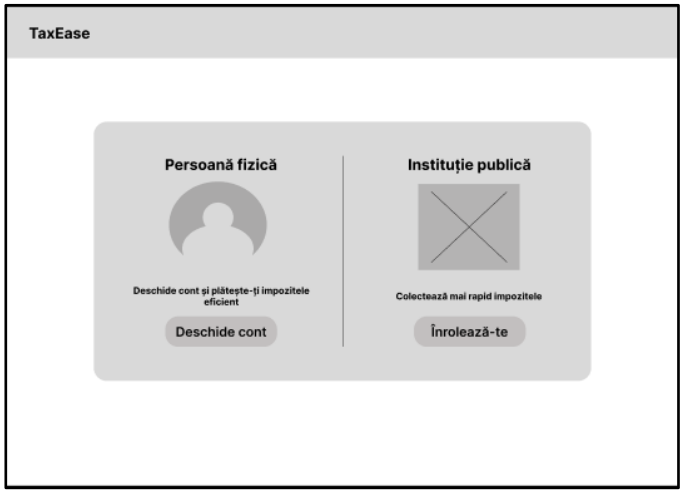

Figura 3.6 - Interfața cu utilizatorul: Pagina de înregistrare (UI02)

Pagina de înregistrare poate fi accesată prin apăsarea butonului ***Înregistrare*** de pe pagina principală. Aceasta servește prin diferențierea proceselor de înregistrare în aplicație pentru o persoană fizică față de o instituție publică colectoare de taxe. Astfel, în cadrul acestei pagini poate fi ales modul de înregistrare, urmând ca utilizatorul să fie direcționat către pagina corespunzătoare.

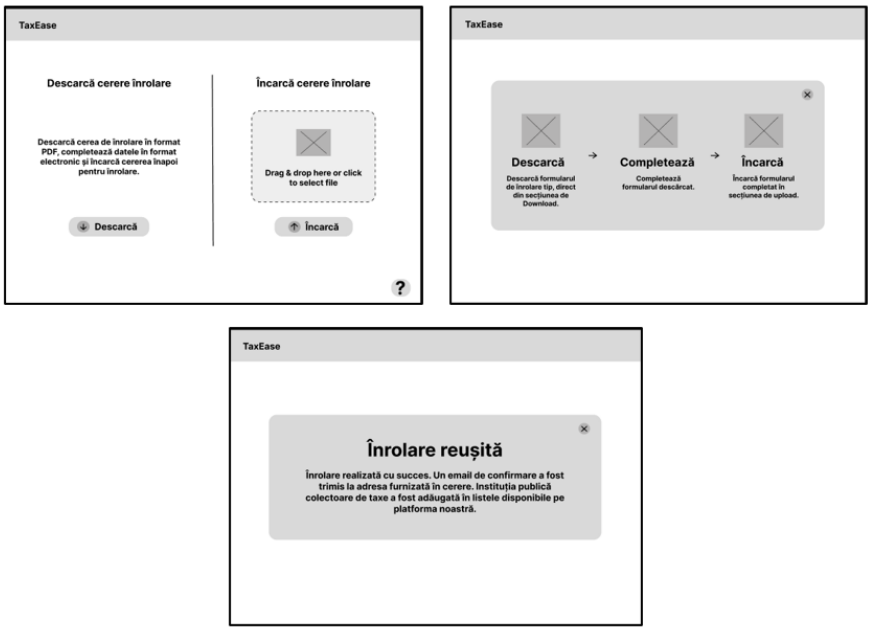

Figura 3.7 - Ansamblul de interfețe cu utilizatorul specific Paginii de înrolare instituție (UI06)

Interfețele prezentate în *Figura 3.7* prezintă fluxul specific procesului de înrolare a unei instituții publice în cadrul aplicației. Astfel, odată aleasă opțiunea de ***Înrolează-te*** din cadrul paginii de înregistrare (*Figura 3.6*), instituția este direcționată spre pagina de înrolare compusă din două secțiuni, de descărcare a cererii de înrolare și respectiv de încărcare a cererii, după completare. De asemenea, acționarea butonului de ajutor, marcat cu semnul întrebării în colțul din dreapta jos a paginii, va deschide fereastra de notificare în care sunt prezentați succinct pașii necesari de parcurs pentru a completa procesul de înrolare. Dacă înrolarea a fost efectuată cu succes, o altă fereastră va apărea cu mesajul „Înrolare reușită”, iar instituția va fi adaugată în baza de date și în listele disponibile din cadrul aplicației.

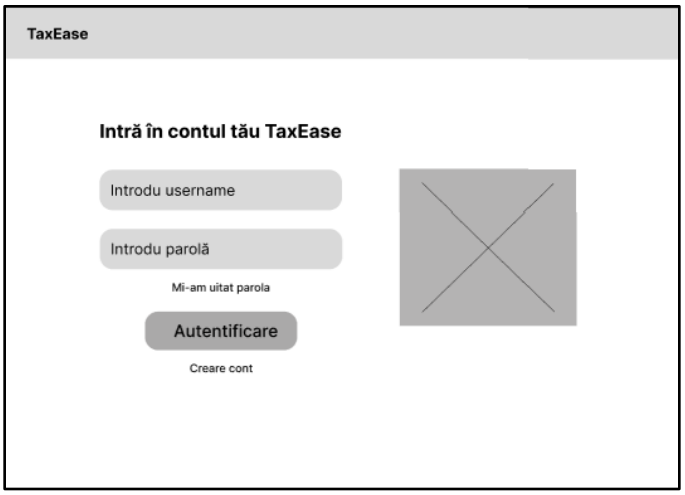

Figura 3.8 - Interfața cu utilizatorul: Pagina de autentificare (UI02)

Pagina de autentificare poate fi accesată prin intermediul butonului ***Autentificare*** prezent în pagina principală și reprezintă formularul de autentificare al unui utilizator cu cont (persoană fizică) în contul său. Utilizatorul este rugat să își introducă credențialele, iar în cazul în care acesta își uită parola, opțiunea „Mi-am uitat parola" îl va direcționa către un formular pentru schimbarea acesteia. De asemenea, în cazul în care un utilizatorul accesează această pagină fără însă a deține un cont în cadrul aplicației, opțiunea „Creare cont" îl va direcționa către interfața U05 (Pagina de creare cont).

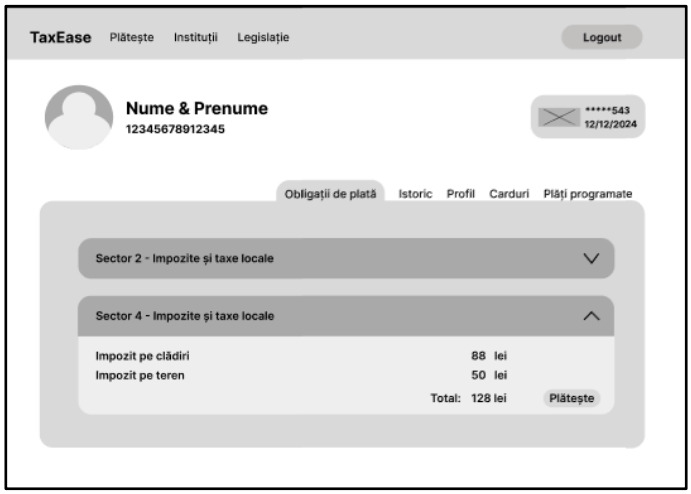

Figura 3.9 - Interfața cu utilizatorul: Pagina personală, afișare obligații de plată (UI03)

Pagina personală, prezentată în *Figura 3.9*, reprezintă interfața cu utilizatorul odată ce acesta se loghează în contul său. Din această pagină, utilizatorul poate vizualiza obligațiile de plată pe care le are de achitat, istoricul plăților efectuate, datele personale înregistrate în cadrul profilului, cardurile înregistrate și eventualele plăți programate, în cazul în care optează pentru crearea acestora.

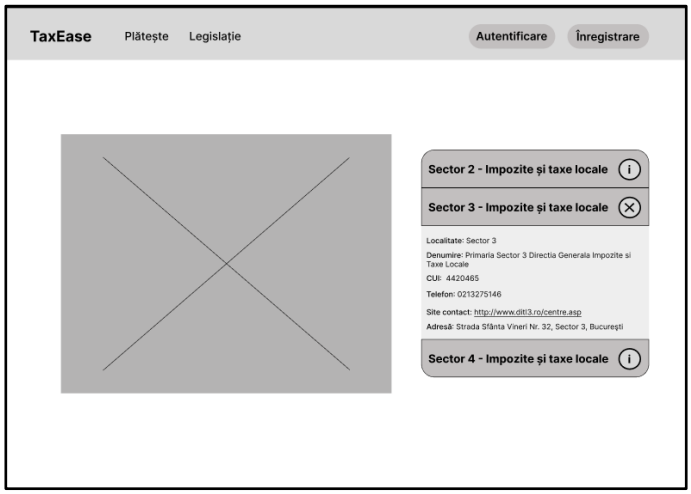

Figura 3.10 - Interfața cu utilizatorul: Pagină instituții (UI07)

Pagina din *Figura 3.10* prezintă lista instituțiilor înrolate în cadrul aplicației *web*, alături de o hartă interactivă a României, împărțită pe județe, astfel încât vor fi afișate instituțiile publice colectoare de taxe disponibile în cadrul aplicației, din județul selectat de către utilizator. De asemenea, sunt prezentate informații generale și de contact despre instituțiile afișate, precum denumirea, telefonul sau *link* către *site*-ul oficial.

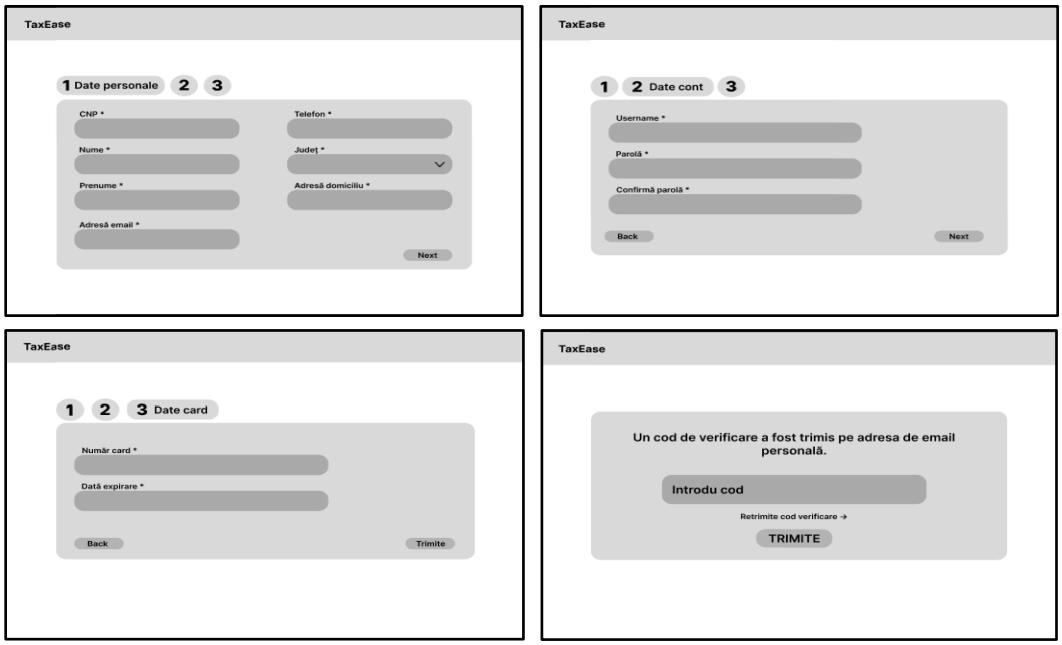

Figura 3.11 - Ansamblul de interfețe cu utilizatorul specific Paginii de creare cont (UI05)

Ansamblul de interfețe prezentat în *Figura 3.11* ilustrează procesul de creare a unui cont în cadrul aplicației *web* pentru o persoană fizică. Astfel, este necesară completarea datelor cerute cu privire la datele personale, datele specifice contului și datele unui card bancar valid, ce poate fi utilizat în realizarea plăților mai departe. De asemenea este reprezentată si interfața de verificare a *email*-ului introdus de către utilizator, acesta fiind nevoit să introducă un cod de verificare primit pe adresa furnizată pentru a încheia procesul de creare a contului.

3.3 TEHNOLOGII INFORMATICE UTILIZATE

Tehnologiile informatice reprezintă un element fundamental ce joacă un rol crucial în dezvoltarea oricărui sistem informatic. Acest lucru este evident și în cazul aplicației de achitare și administrare a taxelor și impozitelor locale, care integrează tehnologii din două domenii distincte: baze de date și concepte *web*. Această combinație de tehnologii permite o funcționalitate complexă și eficientă, asigurând o gestionare adecvată a plăților realizate prin intermediul aplicației *web* și a datelor asociate acestora.

Pentru a stoca, gestiona și accesa într-un mod eficient datele din cadrul aplicației *web*, am utilizat o bază de date *Oracle Database*. Sistem de gestionare a bazelor de date relaționale (RDBMS) dezvoltat de compania *Oracle*, aceasta este populară în domeniul informatic și este utilizată pe scară largă în dezvoltarea și gestionarea bazelor de date, ca urmare a performanței sale și capacității de a gestiona volume mari de date și aplicații critice.[NET07]

Datorită importanței sale în gestionarea datelor și a utilizării largi în domeniul informatic, SQL (*Structured Query Language*) este considerat o tehnologie-cheie și este esențial pentru dezvoltatorii și administratorii de baze de date. Considerat un limbaj de programare puternic, utilizat în mod extensiv în dezvoltarea aplicațiilor și sistemelor bazate pe baze de date relaționale, acesta este esențial pentru manipularea, gestionarea și vizualizarea datelor în sistemul de gestiune al bazelor de date. Astfel, SQL servește ca punte între aspectele conceptuale, logice și fizice ale bazelor de date relaționale.[NET07]

Prin intermediul limbajului SQL am definit structura bazei de date și restricțiile de integritate asociate, utilizând comenzile de definire a datelor (LDD). De asemenea, prin intermediul SQL am creat entitățile necesare bazei de date a aplicației web, asigurând coerența datelor stocate, iar folosind comenzi de manipulare a datelor (LMD) am populat baza de date cu instanțe și am efectuat operații de tipul CRUD (*Create, Read, Update, Delete*) precum operații de inserție, actualizare și ștergere.

În plus, în cadrul aplicației implementate, care îmbină tehnologii din bazele de date și conceptele *web*, limbajul de interogare SQL nu este singura tehnologie utilizată de manipulare a mediului de stocare a datelor. Un alt exemplu important este PL/SQL (*Procedural Language extensions to the Structured Query Language*), un limbaj de programare procedural și o extensie a SQL [NET07].

PL/SQL permite dezvoltatorilor să scrie cod procedural direct în interiorul bazei de date *Oracle*. Acest lucru facilitează manipularea datelor, implementarea logicii de afaceri și crearea de aplicații bazate pe baze de date mai complexe. Prin intermediul PL/SQL, am creat proceduri

stocate, funcții, declarații de variabile și alte componente necesare pentru extinderea funcționalității bazei de date [NET07].

Din punctul de vedere al dezvoltării modulului *web,* implementarea a fost abordată atât pe partea de *backend*, ce include serverele, bazele de date și logica de programare care stau la baza funcționării aplicației, cât și pe partea de *frontend*, ce se referă la partea vizibilă și interactivă a aplicației, cu care utilizatorii interacționează direct.

Astfel, funcționalitățile aplicației *web* și dezvoltarea *server-side* au fost implementate utilizând *Node.js*, ce reprezintă un mediu de execuție *JavaScript* construit pe motorul V8 *JavaScript* dezvoltat de *Google* [NET04]. Acesta permite rularea codului *JavaScript* în afara unui browser *web*, pe servere și dispozitive diverse. *Node.js* se bazează pe un model de evenimente și un sistem de I/O non-blocant, ceea ce îl face ideal pentru aplicații cu trafic intens, solicitări multiple și procesare eficientă a datelor [NET03].

Pe de altă parte, elementele de interfață utilizator (UI) ale aplicației au fost implementate cu ajutorul *React* și Sass, asigurând interacțiunea cu utilizatorul și prezentarea coerentă și plăcută din punct de vedere vizual a informațiilor.

React este o bibliotecă *JavaScript open-source* utilizată pentru dezvoltarea interfețelor utilizator în aplicații *web* și mobile. A fost creată inițial de către *Facebook* și este folosită de multe alte companii și dezvoltatori în întreaga lume. *React* utilizează un model de programare denumit *component-based* (bazat pe componente), care facilitează dezvoltarea și reutilizarea codului în cadrul unei aplicații [NET01].

Sass (*Syntactically Awesome Style Sheets*) este un limbaj de preprocesare CSS, ceea ce înseamnă că este o extensie a limbajului CSS folosit pentru stilizarea paginilor *web*. Sass adaugă funcționalități suplimentare la CSS și facilitează scrierea și organizarea codului CSS. Prin utilizarea variabilelor, operatorilor, buclelor și funcțiilor, Sass permite crearea unor stiluri mai complexe și reutilizabile. Fișierele Sass sunt compilate în fișiere CSS, care apoi sunt utilizate în aplicațiile *web*. Astfel, Sass a devenit popular în comunitatea dezvoltatorilor *web* datorită capacității sale de a face stilizarea mai eficientă și mai ușor de gestionat [NET02].

Mai mult decât atât, pentru a crea, gestiona și distribui aplicația *web* implementată într-un mod eficient și ușor de reprodus, am utilizat *Docker*, ce oferă beneficii precum izolarea aplicațiilor, portabilitatea, scalabilitatea și eficiența resurselor.

Docker este o soluție *open-source* care a fost concepută pentru a facilita dezvoltarea, distribuția și executarea aplicațiilor într-un mod izolat, utilizând conceptul de containere *software*. Aceasta oferă un mediu izolat și portabil în care aplicațiile pot funcționa independent de sistemul de operare gazdă [NET05]. Acesta utilizează tehnologia de virtualizare la nivelul sistemului de operare, prin crearea de "containere", pentru a împacheta aplicațiile și toate dependențele lor într-un format standardizat. Astfel, este facilitat transportul și distribuția aplicațiilor într-un mod

consistent și reproductibil pe diverse medii și infrastructuri, fie că este vorba de medii de dezvoltare, testare sau producție [NET06].

Așadar, etapa de proiectare este importantă pentru a asigura o implementare coerentă și eficientă a sistemului, contribuind la evitarea erorilor și neclarităților în etapele ulterioare ale dezvoltării. Astfel, în acest capitol am oferit baza pentru programarea, testarea și implementarea efectivă a aplicației, ce va fi prezentată în capitolul următor.

Bibliografie

[NET01] https://react.dev

[NET02] https://sass-lang.com/

[NET03] https://nodejs.org/en/about

[NET04] https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs

[NET05] https://www.docker.com/resources/what-container/

[NET06] https://www.oracle.com/ro/cloud/cloud-native/container-registry/what-is-docker/

[NET07] https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch11.htm