

*Anexa 6 la Procedura de înscriere la examenele de finalizare a studiilor*

# PROIECT DE DIPLOMĂ

Îndrumător proiect/Coordonator științific,  
Ș.l.dr.ing. Mihaela ȚIPLEA

Absolvent,  
Liliana- Petruța ZANFIR

Galați  
2025

PROGRAMUL DE STUDII: Calculatoare și Tehnologia Informației

# **APLICAȚIE PENTRU PLANIFICARE INTELIGENTĂ UTILIZÂND TEHNICI DE AI**

Coordonator științific,  
Ș.l.dr.ing. Mihaela ȚIPLEA

Absolvent,  
Liliana- Petruța ZANFIR

Galați  
2025

## Rezumat

Proiectul „Aplicație pentru planificare inteligentă utilizând tehnici de AI” are ca scop realizarea unei soluții software moderne care automatizează procesul de generare a orarelor universitare, adaptată la nevoile reale ale instituțiilor de învățământ din România. Aceasta integrează tehnici de inteligență artificială și metode deterministe pentru a obține orare coerente, corecte și complete pentru toate grupele și subgrupele existente.

Aplicația permite definirea detaliată a regulilor educaționale (cursuri comune pe an, seminare și proiectele pe grupă, laboratoare pe subgrupă, pauze, săli disponibile, discipline, disponibilitatea profesorilor etc.), iar generarea se poate face atât cu ajutorul modelului GPT-4 (OpenAI), cât și printr-un algoritm clasic propriu.

În plus, sistemul oferă funcționalități de validare a structurii orarului generat, verificând absența suprapunerilor între profesori, săli sau activități, precum și respectarea tuturor constrângerilor educaționale impuse.

În urma analizei aplicațiilor existente, s-a constatat lipsa unei soluții complete, adaptate specificului educațional din România, care să combine inteligența artificială cu metode deterministe clasice pentru generarea unui orar coerent și corect. Prin urmare, dezvoltarea unei astfel de aplicații a fost considerată necesară.

Generarea orarului este susținută de modelul GPT-4 oferit de OpenAI, care propune orare complete conform regulilor introduse, și de un algoritm clasic propriu, care aplică o abordare logică deterministă asupra datelor disponibile. Cele două metode au fost evaluate comparativ, în funcție de flexibilitate, viteză de execuție și calitatea rezultatelor.

Lucrarea își propune furnizarea unei soluții moderne, prietenoase cu utilizatorul, clare și intuitive, care să sprijine procesul de planificare a resurselor educaționale și să răspundă cerințelor actuale ale mediului academic.

Rezultatele obținute confirmă eficiența unei aplicații care îmbină AI-ul cu reguli personalizabile, oferind un instrument fiabil pentru planificarea academică, ușor de utilizat și adaptabil la nevoile reale ale utilizatorilor.

Pe viitor, direcțiile de dezvoltare pot include extinderea aplicației pentru a deservi mai multe instituții de învățământ, integrarea cu platforme educaționale deja existente (precum cataloage online) și adăugarea unor mecanisme de învățare automată, capabile să adapteze regulile de generare a orarului în funcție de preferințele și istoricul utilizatorilor.

## CUPRINS

Introducere.....	2
Capitolul 1. STUDIUL PROBLEMEI ȘI ANALIZA SOLUȚIILOR EXISTENTE.....	4
1.1.    Motivația utilizării unei aplicații automate pentru generarea orarului universitar.....	4
1.2.    Analiza aplicațiilor existente pentru generarea orarului .....	6
1.2.1.    UniTime.....	6
1.2.2.    aSc TimeTables .....	7
1.2.3.    Generator-Orare (Horarium.ai) .....	9
1.3.    Profilul utilizatorului și scopul utilizării aplicației .....	10
1.4.    Identificarea operațiilor efectuate de utilizator în cadrul aplicației .....	11
Capitolul 2. Specificarea cerințelor .....	13
2.1 Cerințe privind gestionarea și prelucrarea datelor .....	13
2.2 Cerințe privind generarea și validarea orarului .....	14
Capitolul 3. Denumirea capitolului .....	15
3.1 Arhitectura și funcționalitatea proiectului.....	15
3.2 Diagrame UML pentru modelarea structurii .....	16
3.2.1. Diagrama pachetelor .....	16
3.2.2 Diagrama claselor .....	17
3.2.3 Diagrama de componente.....	19
3.3 Diagrame UML pentru modelarea comportamentului.....	19
3.3.1 Diagrama cazurilor de utilizare .....	20
3.3.2 Diagrama de secvență .....	20
3.3.3 Diagrama de activitate .....	21
Capitolul 4. Denumirea capitolului .....	22
Concluzii .....	23
Bibliografie .....	24
Anexa 1.....	25

## LISTA FIGURILOR

Fig. 1 Interfața aplicației UniTime – module de alocare, orar și sugestii automate.....	6
Fig. 2 Interfața aplicației aSc TimeTables – vizualizarea orarului pe clase și zile .....	8
Fig. 3 Interfața aplicației Horarium.ai – vizualizare orar generat.....	9
Fig. 4 Diagrama arhitecturală a aplicației.....	15

Fig. 5 Diagrama pachetelor- Backend .....	16
Fig. 6 Diagrama pachetelor- Frontend.....	16
Fig. 7 Diagrama pachetelor- Routes.....	17
Fig. 8 Diagrama pachetelor- Pages .....	17
Fig. 9 Diagrama clasei -Backend.....	18
Fig. 10 Diagrama claselor- Frontend .....	18
Fig. 11 Diagrama de componente ale aplicației.....	19
Fig. 12 Diagrama cazurilor de utilizare .....	20
Fig. 13 Diagrama de secvență.....	20
Fig. 14 Diagrama de activitate.....	21

## LISTA TABELELOR

Tabelul 1 Titlul tabelului.....	<b>Error! Bookmark not defined.</b>
---------------------------------	-------------------------------------

## INTRODUCERE

În prezent, organizarea eficientă a activităților educaționale reprezintă o provocare majoră pentru instituțiile de învățământ superior. Numărul mare de discipline, grupe, subgrupe, săli și profesori implică un proces de planificare complex, consumator de timp și predispus la erori umane. Generarea manuală a orarului universitar nu doar că presupune eforturi semnificative, dar adesea nu reușește să răspundă cerințelor reale ale studenților și cadrelor didactice.

Pe măsură ce tehnologia s-a dezvoltat de-a lungul timpului, nevoia de o soluție inteligentă și automatizată pentru a îndeplini această funcție a crescut odată cu avansarea inteligenței artificiale.

Astfel, proiectarea și dezvoltarea acestui tip de instrument, capabil să producă automat programe complete, validate și personalizate, devine obligatorie pentru instituțiile contemporane, dar și o oportunitate de eficientizare și digitalizare a procesului academic.

Platforma dezvoltată oferă o soluție completă de generare automată a orarului, adaptată structurii academice din România. Aceasta ține cont de tipul activității (curs, seminar, laborator, proiect), de distribuția pe ani, grupe și subgrupe, de disponibilitatea profesorilor, generând orare corecte și coerente pentru întreaga structură universitară. Platforma este gândită să fie ușor de utilizat, cu un timp de acomodare redus, oferind utilizatorului o experiență clară și fluentă.

Pentru a răspunde nevoilor diverse ale utilizatorilor, aplicația integrează două metode de generare: una bazată pe inteligență artificială (GPT-4 – OpenAI), care construiește orarul pe baza unor reguli scrise în limbaj natural, și o metodă clasică, algoritmică, ce respectă reguli stricte impuse prin logica implementată.

Funcționalitățile propuse pentru realizarea acestei aplicații includ următoarele:

- Generarea automată a orarului pentru toate grupele și subgrupurile unei facultăți, în funcție de datele introduse și regulile definite;
- Integrarea a două metode de generare: una bazată pe inteligență artificială (GPT-4), care prelucrează reguli în limbaj natural, și una clasică, deterministă, care aplică algoritmi proprii pentru plasarea activităților;
- Configurarea regulilor personalizate: intervale orare pentru licență/master, restricții privind pauzele, tipuri de activități per structură (cursuri pentru ani, seminare și proiecte pe grupe, laboratoare pe subgrupe);
- Gestionarea datelor academice: introducerea și modificarea profesorilor, disciplinelor, sălilor (curs, seminar, laborator, proiect) și a structurii de grupe/subgrupe;
- Validarea automată a orarului generat, prin detectarea suprapunerilor de profesori, săli sau activități și verificarea respectării tuturor regulilor impuse;
- Exportul orarului în format PDF și Excel, pentru utilizare oficială și distribuire rapidă;
- Interfață grafică modernă și receptivă, care permite navigarea ușoară și clară între componentele aplicației;

- Posibilitatea de extindere cu module suplimentare în viitor, cum ar fi planificarea examenelor sau a activităților extracurriculare.

Prin urmare, această aplicație aduce o contribuție semnificativă la modernizarea procesului educațional, oferind un instrument eficient pentru alocarea resurselor și generarea de orare fiabile, rapide și ușor accesibile. Soluția propusă susține digitalizarea completă a activităților academice și vine în sprijinul personalului didactic și administrativ, reducând sarcinile repetitive și optimizând procesul de planificare.

## CAPITOLUL 1. STUDIUL PROBLEMEI ȘI ANALIZA SOLUȚIILOR EXISTENTE

Pentru dezvoltarea unei aplicații sustenabile și eficace, a fost necesar un studiu preliminar care să includă o analiză generală a aplicațiilor deja existente pe piață. Această analiză pune accentul pe funcționalitățile oferite, modul în care utilizatorul interacționează cu interfața, precum și pe identificarea elementelor ce pot fi îmbunătățite pentru a asigura o navigare cât mai intuitivă și eficientă.

Ordonarea activităților academice într-un mod coerent și eficient reprezintă o sarcină esențială pentru orice instituție de învățământ superior. Generarea manuală a orarului universitar este un proces complex și consumator de timp, care trebuie să țină cont de numeroși factori: disponibilitatea cadrelor didactice, distribuția disciplinelor, capacitatea și tipul sălilor de curs, dar și compatibilitatea dintre grupe și subgrupe. În absența unei soluții automatizate, acest demers poate conduce la erori, suprapuneri sau programări ineficiente.

În acest context, implementarea unei aplicații automatizate pentru generarea orarului devine nu doar oportună, ci necesară. O astfel de soluție reduce considerabil efortul administrativ și oferă rezultate rapide, coerente și adaptabile specificului fiecărei facultăți sau specializări.

Acest capitol are ca scop evidențierea importanței unui sistem de tip TTGS (Timetable Generation System), analiza comparativă a aplicațiilor existente cu funcționalități similare, definirea profilului utilizatorului cărui i se adresează aplicația, precum și prezentarea funcționalităților de bază dezvoltate în cadrul acestei lucrări.

### 1.1. Motivația utilizării unei aplicații automate pentru generarea orarului universitar

Generarea orarului universitar reprezintă o activitate esențială, dar deosebit de complexă din punct de vedere administrativ. Aceasta necesită luarea în considerare a numeroși factori, precum: disponibilitatea profesorilor, alocarea disciplinelor, capacitatea și tipul sălilor în care se desfășoară activitățile, gruparea studenților în funcție de nivel (an, grupă sau subgrupă), precum și evitarea suprapunerilor sau a pauzelor nejustificate din program.

În prezent, în multe instituții de învățământ, acest proces este realizat manual sau cu ajutorul unor aplicații semi-automatizate. Drept urmare, se consumă mult timp, pot apărea erori frecvente, conflicte de program, distribuții dezechilibrate ale activităților sau chiar omiterea unor discipline. În plus, aceste metode sunt adesea rigide, greu de modificat și dificil de validat într-un mod transparent.

În ultimii ani, s-a observat o tendință clară spre automatizarea acestui proces prin utilizarea sistemelor informatice dedicate. Platforme precum UniTime [1], aSc TimeTables [2] sau GeneratorOrare [3] oferă facilități avansate pentru planificarea automată a orarelor. Acestea permit definirea de constrângeri rigide și flexibile (hard și soft constraints), gestionarea preferințelor cadrelor didactice și generarea orarelor



optimizate, folosind algoritmi de satisfacere a constrângerilor (CSP) sau algoritmi genetici.

Totodată, inteligența artificială devine o soluție tot mai răspândită pentru generarea automată a orarelor. Modele avansate, precum GPT-4 [4], permit formularea regulilor în limbaj natural și generarea de orare structurate (în format JSON, HTML etc.). Acest lucru deschide noi perspective pentru personalizarea și flexibilizarea planificării academice, chiar dacă aplicarea acestor tehnologii în mediul universitar este încă la început.

Utilizarea unei aplicații automatizate pentru generarea orarului este justificată printr-o serie de avantaje concrete:

- Reducerea timpului și a efortului administrativ: Generarea manuală a unui orar complet poate dura zile sau săptămâni, în special în instituțiile cu mai multe programe, grupe și cadre didactice. Un sistem automat poate realiza această sarcină în câteva secunde sau minute, în funcție de complexitate, eliminând munca repetitivă și costisitoare.
- Gestionarea eficientă a regulilor și constrângerilor: O aplicație automatizată poate procesa simultan un număr mare de reguli, atât rigide (ex. imposibilitatea ca un profesor sau o sală să fie alocate simultan la mai multe activități), cât și flexibile (ex. preferințe orare ale cadrelor didactice, evitarea pauzelor lungi, programări într-un anumit interval orar pentru o activitate specifică la nivel de facultate).
- Reducerea erorilor și validarea structurată a orarului: Automatizarea contribuie la eliminarea erorilor frecvente, cum ar fi suprapunerile de activități, lipsa sălilor disponibile sau omisiunea unor discipline. Sistemul oferă validări automate care asigură coerența și corectitudinea orarului generat.
- Flexibilitate și adaptabilitate la modificări: În cazul modificării unor parametri (ex. indisponibilitatea unui profesor sau rezervarea unei săli), aplicația permite regenerarea orarului păstrând, pe cât posibil, structura inițială. Astfel, adaptarea la schimbări este rapidă și fără impact major asupra planificării generale.
- Transparență și validare accesibilă: Orarul generat este prezentat într-un format clar și structurat, putând fi exportat în formate uzuale (PDF, Excel, HTML). Acest lucru facilitează revizuirea, validarea și distribuirea către studenți, cadre didactice sau personal administrativ, contribuind la un proces educațional transparent și eficient.

În concluzie, generarea manuală a orarului este un proces laborios și predispus la erori. Soluțiile moderne, bazate pe algoritmi sau inteligență artificială, oferă un cadru mai eficient, mai flexibil și mai sigur pentru planificarea academică. Acestea pot transforma semnificativ modul în care instituțiile de învățământ organizează activitățile didactice, contribuind la un sistem educațional mai bine structurat și adaptat nevoilor actuale.

## 1.2. Analiza aplicațiilor existente pentru generarea orarului

Pentru a fundamenta procesul de proiectare al aplicației propuse, a fost realizată o analiză comparativă a principalelor soluții existente pe piață destinate generării automate a orarelor universitare. Această analiză s-a concentrat pe modul de funcționare al aplicațiilor, gradul de interactivitate oferit utilizatorului, particularitățile interfeței grafice și aspectele care pot fi optimizate pentru a facilita o utilizare mai intuitivă și eficientă.

În urma analizării mai multor platforme, s-a constatat că multe aplicații actuale dispun de interfețe greoaie, dificil de utilizat și configurat, necesitând adesea instalare locală și un consum ridicat de resurse. În plus, unele soluții sunt costisitoare, au timpuri mari de procesare și presupun o curbă de învățare ridicată pentru utilizatorii fără experiență tehnică. Suplimentar, afișarea simultană a unor volume mari de informații într-o singură vizualizare poate afecta negativ experiența de utilizare, ducând la confuzie sau omisiuni.

Pentru această analiză, au fost selectate trei aplicații reprezentative: UniTime, aSc TimeTables și Generator-Orare (Horarium.ai). Criteriile de selecție au inclus notorietatea în domeniu, diversitatea funcționalităților oferite și diferențele de abordare tehnică și practică. Scopul acestei analize a fost identificarea punctelor forte și a limitărilor fiecărei aplicații, în vederea extragerii unor concluzii utile pentru dezvoltarea unei soluții personalizate, adaptate nevoilor specifice ale instituțiilor de învățământ superior.

### 1.2.1. UniTime

UniTime este un sistem open-source de programare academică, dezvoltat inițial de Universitatea Indiana și ulterior adoptat de numeroase colegii și universități din întreaga lume [1]. Popularitatea sa se datorează nivelului ridicat de configurabilitate și capacității de extindere, fiind potrivit mai ales pentru instituții de mari dimensiuni (Figura 1).

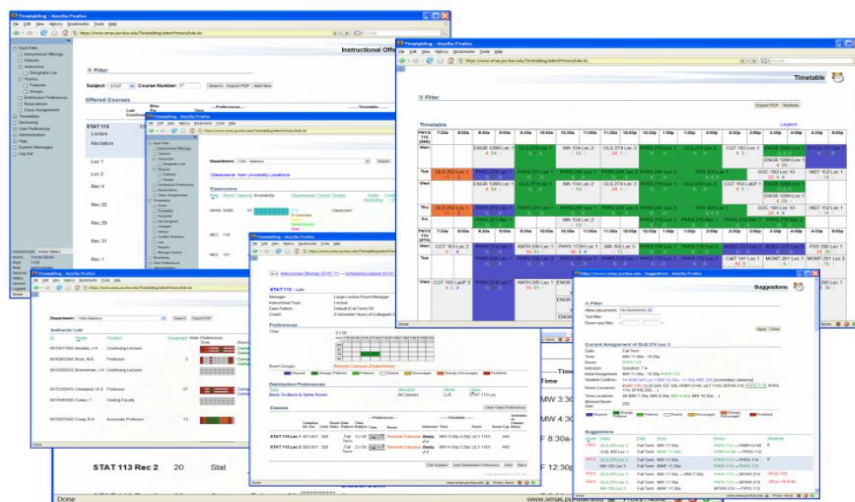


Fig. 1 Interfața aplicației UniTime – module de alocare, orar și sugestii automate

Aplicația oferă funcționalități avansate, printre care:

- definirea preferințelor cadrelor didactice;
- alocarea automată a resurselor (săli, clădiri, echipamente);
- aplicarea de constrângeri rigide și flexibile (hard/soft constraints);
- selecția algoritmului de programare (solver);
- exportul orarului în format iCalendar;
- integrarea cu sisteme externe existente;
- vizualizarea programelor personalizate pentru studenți și profesori.

Un avantaj semnificativ este posibilitatea de a configura în detaliu regulile și constrângerile, adaptând planificarea la particularitățile instituției. Interfața de administrare permite gestionarea sesiunilor, resurselor, activităților și aspectelor vizuale ale programului.

Totuși, în ciuda versatilității sale, UniTime prezintă și o serie de dezavantaje majore:

- Instalarea este complexă, necesitând cunoștințe tehnice avansate și configurarea unor componente multiple precum Java, Apache Tomcat, MySQL și fișiere de sistem (ex: custom.properties). Instituțiile fără o echipă IT dedicată pot întâmpina dificultăți în utilizarea platformei.
- Interfața grafică, deși funcțională, nu este prietenoasă pentru utilizatorii non-tehnici. Navigarea poate fi dificilă la început, în special din cauza lipsei unei documentații clare și concentrate pe utilizatorul final. De asemenea, aplicația nu este optimizată pentru dispozitive mobile.
- Comunitatea de utilizatori este limitată, iar suportul tehnic oficial lipsește, ceea ce face rezolvarea problemelor dependentă de documentația tehnică, adesea fragmentată și dificil de înțeles pentru personalul administrativ.

În concluzie, UniTime este o soluție foarte puternică și flexibilă pentru programarea academică, recomandată în special instituțiilor mari, care pot valorifica avantajele sale prin personal tehnic specializat. Cu toate acestea, complexitatea sa de configurare și utilizare o face mai puțin potrivită pentru instituțiile de dimensiuni mici sau pentru utilizatorii fără competențe tehnice solide.

### 1.2.2. aSc TimeTables

aSc TimeTables este una dintre cele mai cunoscute aplicații comerciale pentru generarea automată a orarelor, fiind utilizată pe scară largă în instituții de învățământ din țări precum Germania, Statele Unite, India, Brazilia, dar și în numeroase state din Africa și Australia [2]. Aplicația este disponibilă atât sub formă de program desktop pentru Windows, cât și ca serviciu online, oferind planuri tarifare variate – de la 99euro (pentru școli primare) până la 1995 euro (versiunea profesională).

Aplicația este apreciată pentru gradul ridicat de automatizare, interfața prietenoasă și simplitatea cu care un orar complet poate fi generat în doar câteva clicuri. Introducerea datelor inițiale – precum clase, profesori, discipline și reguli de planificare – se face intuitiv, fără a necesita pregătire tehnică prealabilă. Algoritmul încorporat

evaluează milioane de combinații posibile în câteva secunde, generând un orar optimizat care respectă toate constrângerile: absența suprapunerilor, menținerea perioadelor libere pentru profesori, utilizarea eficientă a sălilor și evitarea blocajelor orare (Figura 2).

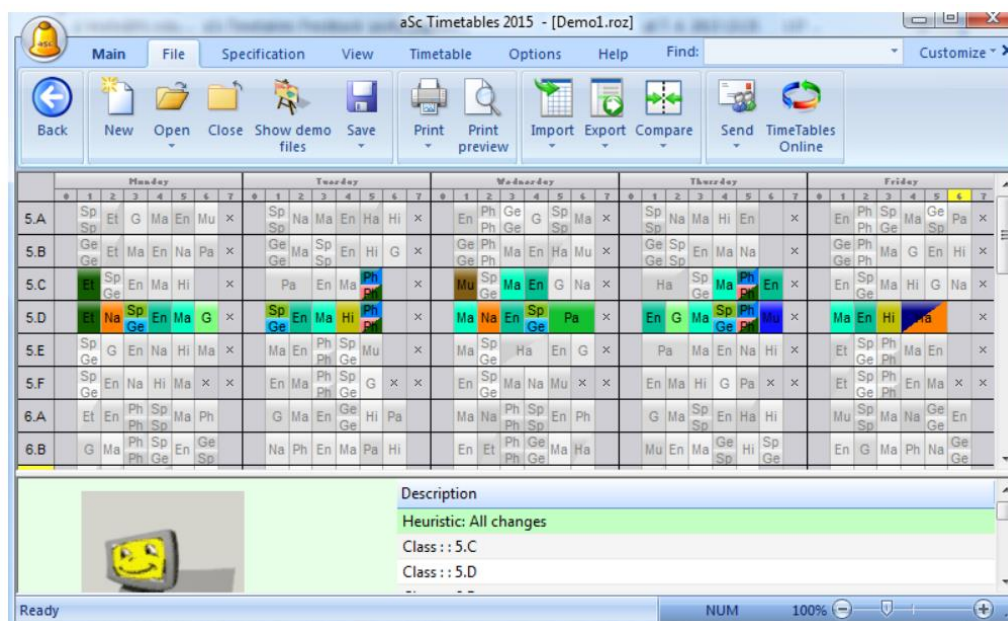


Fig. 2 Interfața aplicației aSc TimeTables – vizualizarea orarului pe clase și zile

Un alt punct forte este posibilitatea de export al orarului în multiple formate – PDF, Excel, HTML sau imagine – precum și integrarea cu platforma EduPage, care permite gestionarea elevilor, cataloage electronice, prezență și comunicarea cu părinții. Disponibilitatea aplicației în limba română și existența unei versiuni adaptate sistemului educațional românesc constituie un avantaj semnificativ pentru utilizatorii locali. Suportul tehnic este prompt, iar documentația detaliată și tutorialele ușor de urmărit facilitează accesul chiar și pentru utilizatorii fără cunoștințe IT avansate.

Totuși, aplicația prezintă și limitări importante:

- Versiunea gratuită este extrem de restrictivă, lipsind funcționalități esențiale precum exportul și salvarea orarului, ceea ce o face aproape inutilizabilă fără achiziționarea unei licențe.
- Gradul de personalizare este limitat comparativ cu alte platforme mai flexibile. Nu permite definirea de reguli avansate prin cod sau configurarea unor algoritmi proprii.
- Versiunea desktop necesită instalare locală și funcționarea ei completă depinde de conexiunea la internet, mai ales pentru funcțiile avansate precum integrarea cu EduPage sau partajarea online.

În concluzie, aSc TimeTables este o soluție performantă pentru generarea automată a orarelor, recomandată în special instituțiilor de învățământ preuniversitar sau universitar cu cerințe de complexitate medie. Deși este mai puțin flexibilă și extensibilă decât alte aplicații avansate, rămâne o alegere excelentă datorită interfeței intuitive, rapidității de generare și simplității în utilizare.

### 1.2.3. Generator-Orare (Horarium.ai)

Generator-Orare, redenumit recent Horarium.ai, este o aplicație online dedicată programării automate a orarelor școlare [3]. Spre deosebire de alte soluții mai tehnice sau greu de configurat, Horarium se remarcă prin simplitatea utilizării și prin interfața intuitivă, fiind accesibil direct din browser, fără a necesita instalare locală sau configurații suplimentare (Figura 3).

INFO Ecaterina Boarna Laborator	MAT Cristian Stanciu	BIO Stefania Gheorghescu
MAT Cristian Stanciu	ENG Larisa Cristea Ib.1	EDFIZ Cosmin Popescu Ib.1 Saia de sport
ROM Monica Stoian 🔒	ROM Monica Stoian	GEOGRAFIE Marian Cretu
INFO Ecaterina Boarna Laborator	INFO Ecaterina Boarna Laborator	MAT Cristian Stanciu
ROM Monica Stoian	MAT Cristian Stanciu	BIO Stefania Gheorghescu

Fig. 3 Interfața aplicației Horarium.ai – vizualizare orar generat

Sistemul este complet automatizat și ghidează utilizatorul prin fiecare etapă a procesului. După introducerea elementelor esențiale (profesori, materii, clase, săli și reguli generale), aplicația generează orarul respectând constrângerile tipice impuse în sistemul educațional. Interfața este adaptată utilizatorilor non-tehnici, ceea ce o face ideală pentru personalul secretarial sau administrativ fără pregătire IT avansată. Navigarea în aplicație este fluidă, iar rezultatele sunt afișate într-un mod clar, lizibil și structurat.

Un avantaj important este faptul că platforma nu solicită efort tehnic din partea utilizatorului – majoritatea acțiunilor fiind automatizate și ușor de urmărit. Designul este responsive și poate fi accesat și de pe dispozitive mobile, însă experiența optimă rămâne pe ecranele mari, pentru o vizualizare completă a orarului.

Totuși, Horarium.ai prezintă și unele limitări semnificative:

- Nu dispune de o versiune mobilă dedicată și, deși interfața este responsivă, funcționalitatea completă nu este optimizată pentru utilizarea pe telefoane sau tablete.
- Nu permite configurarea de constrângeri avansate sau definirea unor reguli complexe, precum cele legate de preferințele profesorilor, intervale orare rezervate sau interdependențe între activități.
- Nu oferă posibilitatea de import/export extins sau integrare cu sisteme externe prin API.

- Platforma este accesibilă contra cost, iar modelul de tarification poate deveni o barieră pentru unele instituții, în lipsa unei versiuni gratuite funcționale.

În concluzie, Horarium.ai este o soluție eficientă pentru instituțiile care își doresc un instrument simplu, rapid și accesibil pentru generarea orarelor. Este ideală pentru utilizatorii care nu dispun de resurse tehnice avansate și doresc un rezultat funcțional cu un efort minim. Cu toate acestea, pentru aplicații de nivel universitar sau pentru integrarea într-un sistem educațional complex, limitările sale privind flexibilitatea și personalizarea pot deveni constrângătoare.

### 1.3. Profilul utilizatorului și scopul utilizării aplicației

Aplicația descrisă în acest document a fost creată folosind un model centralizat de utilizare, astfel încât același utilizator, având rolul de administrator complet, este responsabil de toate etapele de configurare și de generare a orarului universitar. Adaptarea acestui model pentru un singur utilizator se datorează simplității și eficienței sale funcționale și, de asemenea, pentru că reproduce lumea reală de operare care se întâmplă în multe instituții mici și mijlocii, unde o singură persoană (responsabilă cu orarul) este însărcinată cu generarea orarelor (în general, această persoană face parte din personalul secretariatului sau este responsabilă de activitățile academice).

Utilizatorul final nu este împărțit în roluri (de exemplu, studenți, profesori, secretariat sau biroul decanului), având acces complet la toate părțile sistemului. Acest utilizator elaborează informațiile necesare despre personalul didactic, disciplinele aferente, grupurile și subgrupurile de studenți, sălile disponibile și, de asemenea, despre regulile și constrângerile impuse în procesul de planificare. În acest sens, aplicația nu este un mediu colaborativ, ci funcționează ca un instrument de lucru pentru utilizatorul care deține toate drepturile asupra întregii proceduri.

Avantajul acestui model este că oferă consistență operațională, simplifică administrarea - nu este nevoie de autentificare multiplă, nu este nevoie de drepturi de acces diferite și nu este nevoie de sincronizarea diferitelor roluri de utilizator. Totul este într-un singur loc, controlat global, cu un flux direct, rapid și previzibil.

Aplicația se adresează în special personalului administrativ implicat în organizarea programului academic, oferindu-i un instrument modern, rapid și ușor de utilizat pentru generarea orarelor universitare. Automatizarea procesului contribuie la reducerea efortului manual și a erorilor umane, păstrând în același timp un nivel ridicat de control asupra datelor.

Obiectivele aplicației sunt:

- Centralizarea și intuitivizarea procesului de planificare academică: Platforma oferă o interfață prietenoasă, modernă și centralizată, care facilitează accesul utilizatorilor la toate funcționalitățile necesare generării și ajustării unui orar universitar complet.
- Accelerarea procesului de generare a orarului: Comparativ cu metodele tradiționale de planificare (precum utilizarea fișierelor Excel sau completarea manuală), aplicația permite construirea unui orar coerent într-un interval de

timp mult mai scurt, crescând semnificativ eficiența personalului administrativ.

- Aplicație statică bazată pe reguli și constrângeri instituționale: Motorul de generare ține cont de regulile rigide impuse de instituție, cum ar fi evitarea suprapunerii activităților, alocarea corectă a sălilor, respectarea disponibilității profesorilor și a structurii grupelor și subgrupelor, minimizând astfel erorile și conflictele.
- Asigurarea flexibilității în reorganizarea orarului: Modificările punctuale sau de ansamblu pot fi realizate fără a compromite structura generală a orarului. Aplicația este suficient de robustă pentru a susține scenarii de reconfigurare rapidă, păstrând totodată consistența planificării.
- Export facil în formate lizibile și compatibile: Orarul generat poate fi exportat în formate comune și ușor de distribuit (PDF, Excel, HTML), fără a necesita intervenții ulterioare de rearanjare sau conversie. Acest aspect facilitează publicarea orarului pe platforme interne sau afișarea acestuia în format tipărit.

Utilizarea unei astfel de aplicații nu doar că optimizează munca personalului academic, ci contribuie direct la creșterea transparenței și previzibilității programului pentru studenți și cadre didactice. Prin reducerea erorilor umane și a timpului de lucru necesar planificării, instituția beneficiază de un proces mai clar, mai rapid și mai puțin susceptibil la modificări neplanificate.

Astfel, prin adoptarea unui model unitar și centralizat de utilizare, aplicația propusă reușește să simplifice întregul proces de generare a orarului universitar, oferind un control deplin asupra datelor și regulilor implicate. Lipsa unui sistem multi-utilizator este o alegere intenționată, orientată spre eficiență și operativitate în contextul instituțiilor cu resurse limitate, unde un singur responsabil poate gestiona întreaga activitate.

#### 1.4. Identificarea operațiilor efectuate de utilizator în cadrul aplicației

Aplicația de generare automată a orarului universitar este concepută pentru a oferi utilizatorului control complet asupra întregului flux de creare, personalizare și export al unui orar academic valid. Printr-o interfață intuitivă, fiecare operație realizată de utilizator contribuie direct la formarea unei structuri educaționale coerente, respectând regulile și constrângerile impuse de instituție. Toate funcționalitățile sunt centralizate într-un panou unic, adaptat unui sistem cu un singur utilizator, fără roluri distincte sau drepturi diferențiate.

În prima etapă, utilizatorul introduce toate datele relevante necesare pentru generarea orarului. Aceste date includ configurarea anilor de studiu, grupelor și subgrupelor, introducerea sălilor de activitate (de tip curs, seminar, laborator și proiect), precum și definirea completă a cadrelor didactice – cu nume, discipline predate, tipuri de activități, niveluri de studii la care vor preda și disponibilitate detaliată. De asemenea, utilizatorul are posibilitatea de a crea, modifica sau șterge seturi

de reguli care ghidează procesul de generare automată a orarului. Aceste reguli sunt redactate în format structurat și salvat în baza de date pentru a fi reutilizate.

Ulterior, utilizatorul declanșează procesul de generare a orarului, alegând între două metode disponibile: algoritmul clasic implementat local sau generatorul bazat pe inteligență artificială (GPT-4). Indiferent de metodă, utilizatorul nu intervine direct în procesul de alocare a activităților, însă are control asupra datelor de intrare și poate relansa procesul cu noi condiții, dacă este necesar. După generare, utilizatorul analizează orarul rezultat și poate opta pentru salvarea acestuia, redenumirea sa, încărcarea unui orar anterior sau ștergerea versiunilor vechi care nu mai sunt relevante.

În final, aplicația oferă funcționalități extinse pentru vizualizarea orarului într-un format tabelar, filtrarea acestuia după nivelul de studii și anul academic, precum și exportul în formate uzuale – PDF sau Excel. Aceste operații sunt realizate direct de utilizator prin intermediul interfeței grafice, fără a necesita cunoștințe tehnice avansate.

Prin urmare, operațiile realizate de utilizator acoperă întregul spectru de activități: introducerea, modificarea și ștergerea datelor, setarea regulilor de generare, alegerea metodei de generare, gestionarea versiunilor de orar și exportul acestora, reflectând o interacțiune completă, coerentă și facilă cu aplicația.



## CAPITOLUL 2. SPECIFICAREA CERINTELOR

Aplicația propusă are ca obiectiv principal furnizarea unei soluții digitale moderne pentru generarea automată a orarului universitar, utilizând reguli și constrângeri prestabilite, precum și date reale extrase din baza de date (ani de studiu, grupe și subgrupe, săli, cadre didactice, discipline). Procesul tradițional de creare a orarelor academice este adesea anevoios, consumator de timp, predispus la erori și dificil de adaptat la modificări ulterioare.

Prin această aplicație, se urmărește automatizarea întregului flux de planificare, oferind utilizatorului o interfață intuitivă și un set complet de funcționalități pentru introducerea datelor, generarea și validarea unui orar coerent, complet și adaptat cerințelor instituționale.

### 2.1 Cerințe privind gestionarea și prelucrarea datelor

Pentru a asigura un proces complet de generare a orarului, aplicația trebuie să gestioneze în mod eficient toate datele esențiale implicate în organizarea academică. Această componentă a sistemului vizează operațiile necesare pentru introducerea, modificarea, căutarea și stocarea datelor referitoare la grupe, săli, profesori și reguli de generare. Gestionarea corectă a acestor informații constituie baza funcțională pentru o planificare coerentă și automatizată a activităților didactice.

Sistemul va permite următoarele operații:

#### 1. Introducerea datelor:

- Definirea grupelor și subgrupelor, incluzând informații precum anul de studiu, denumirea și nivelul (Licență/Master);
- Înregistrarea sălilor disponibile, cu specificarea tipului (curs, seminar, laborator, proiect);
- Adăugarea cadrelor didactice, împreună cu disciplinele predate, nivelurile de predare, tipurile de activități asociate și intervalele de disponibilitate;
- Stabilirea regulilor de generare, care includ intervale orare permise, durate de pauză, programul general și formatul afișării activităților în orar.

#### 2. Modificarea datelor:

- Actualizarea disciplinelor asociate unui profesor;
- Modificarea disponibilității acestuia;
- Editarea sau redenumirea sălilor de curs;
- Restructurarea grupelor și subgrupelor existente;
- Încărcarea și editarea regulilor din baza de date.

#### 3. Căutare și filtrare:

- Căutarea rapidă după numele profesorului, grupei sau disciplinei;
- Filtrarea după nivel de studiu, an, grupă, subgrupă sau tip de activitate.

#### 4. Gestionarea orarelor salvate:

- Salvarea orarelor generate în baza de date;

- Reîncărcarea și consultarea ulterioară a acestora;
- Vizualizarea unui istoric al orarelor create anterior.

Prin centralizarea acestor operații într-un sistem unitar și accesibil, aplicația oferă utilizatorului control complet asupra tuturor datelor necesare generării orarului. Flexibilitatea în actualizarea și filtrarea informațiilor, alături de posibilitatea de salvare și reutilizare a datelor introduse, contribuie la optimizarea procesului de planificare academică și la reducerea semnificativă a timpului alocat gestionării manuale a acestora.

## 2.2 Cerințe privind generarea și validarea orarului

După colectarea și structurarea datelor relevante, etapa centrală a aplicației constă în generarea efectivă a orarului universitar. Această funcționalitate presupune alocarea inteligentă și automată a activităților didactice în funcție de reguli prestabilite, disponibilitatea resurselor și tipologia activităților. În paralel, sistemul trebuie să includă mecanisme de validare a orarului rezultat, pentru a detecta posibile conflicte și incoerențe. Astfel, sunt asigurate coerența, fezabilitatea și calitatea planificării orarelor generate.

Aplicația va include funcționalități avansate pentru generarea și validarea orarului:

### 1. Generarea orarului:

- Selectarea metodei de generare: Utilizarea unui model AI (ex. GPT-4)/ utilizarea unui algoritm determinist clasic;
- Alocarea automată a activităților, ținând cont de disponibilități, tipuri de activități și regulile configurate de utilizator.

### 2. Validarea orarului:

- Suprapuneri de profesori sau săli;
- Lipsa activităților într-o zi sau existența unor intervale orare neacoperite;
- Desincronizări ale cursurilor comune la nivel de an.

### 3. Exportul orarului:

- Exportul local în format PDF sau Excel;
- Posibilitatea de partajare sau arhivare a orarului generat.

Prin integrarea acestor funcționalități, aplicația asigură nu doar automatizarea procesului de generare a orarului, ci și validarea riguroasă a rezultatului, eliminând conflictele și optimizând distribuția resurselor. Posibilitatea de a exporta orarul în formate accesibile și reutilizabile consolidează caracterul practic al aplicației și facilitează utilizarea directă a rezultatelor în mediul academic.

Prin definirea clară a cerințelor privind atât gestionarea datelor, cât și procesul de generare și validare a orarului, aplicația propusă se conturează ca un instrument complet și eficient pentru planificarea academică. Îmbinând funcționalități avansate de introducere, actualizare și filtrare a datelor cu algoritmi inteligenți de generare și mecanisme automate de verificare a conflictelor, sistemul oferă un mediu unificat, fiabil și ușor de utilizat.

## CAPITOLUL 3. DENUMIREA CAPITOLULUI

Proiectarea aplicației pentru planificare inteligentă are ca obiectiv dezvoltarea unui sistem informatic capabil să genereze automat orare universitare, prin integrarea tehnologiilor moderne de inteligență artificială. Această etapă presupune definirea unei arhitecturi robuste, care să gestioneze eficient componentele esențiale ale aplicației – interfața utilizator (frontend), logica aplicației și serviciile (backend), precum și sistemul de gestiune a datelor (baza de date).

Un element distinctiv al proiectului îl reprezintă componenta de inteligență artificială, integrată pentru a automatiza procesul de alocare a resurselor educaționale conform unui set de reguli și constrângeri predefinite. Astfel, sistemul oferă o soluție modernă și adaptabilă, care combină o interfață intuitivă cu o logică de planificare avansată, permițând utilizatorului să introducă și să gestioneze date complexe într-un mod simplificat, obținând în final un orar valid, coerent și complet, generat în mod automat.

### 3.1 Arhitectura și funcționalitatea proiectului

Arhitectura aplicației este de tip client-server și evidențiază separarea clară între componentele principale: interfața frontend realizată în React.js, logica de procesare din backend (Flask API), baza de date MySQL și modulele de generare și validare a orarului.

După autentificare, utilizatorul interacționează cu interfața aplicației pentru a introduce date despre grupe, săli, profesori și reguli, care sunt apoi procesate prin rutele corespunzătoare din backend. Datele sunt stocate într-o bază relațională, iar generarea orarului este realizată fie cu ajutorul unui model AI (GPT-4), fie cu un algoritm clasic implementat local. Diagrama arhitecturală (Figura 4) reflectă această structură modulară și fluxul logic al aplicației.

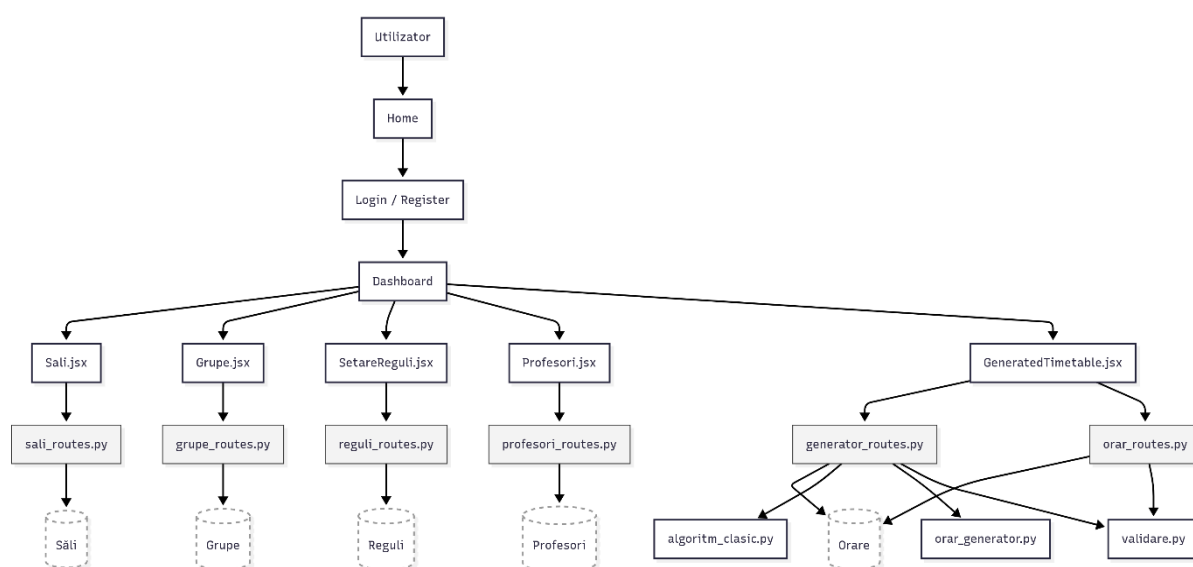


Fig. 4 Diagrama arhitecturală a aplicației

### 3.2 Diagrame UML pentru modelarea structurii

Diagramele UML (Unified Modeling Language) sunt reprezentări grafice standardizate utilizate pentru a vizualiza, specifica, construi și documenta componentele unui sistem software. Acestea ajută la înțelegerea arhitecturii aplicației și facilitează comunicarea între membrii echipei de dezvoltare.[6]

#### 3.2.1. Diagrama pachetelor

Diagramele de pachete sunt utilizate pentru a reprezenta organizarea logică a componentelor unui sistem software, grupate în module sau pachete. Ele evidențiază dependențele dintre module și oferă o imagine de ansamblu asupra structurii proiectului.[7]

În cadrul aplicației dezvoltate, au fost create mai multe diagrame de pachete pentru:

1. Backend-ul aplicației este structurat modular, respectând principiile unei arhitecturi bine organizate și scalabile. Componentele principale sunt grupate în pachete distincte, fiecare având un rol clar în funcționarea sistemului. Diagrama din Figura 5 evidențiază această organizare logică, în care sunt delimitate modulele pentru conexiunea la baza de date, logica de generare și validare a orarului, rutarea cererilor, testarea funcționalităților și inițializarea aplicației. Această abordare facilitează dezvoltarea clară și întreținerea eficientă a codului.

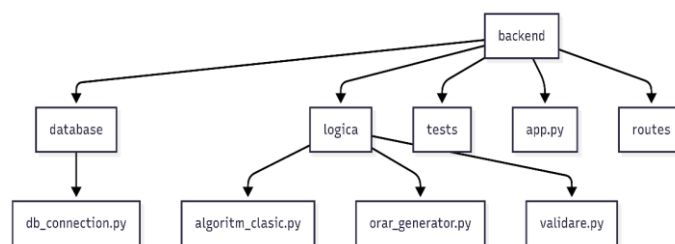


Fig. 5 Diagrama pachetelor- Backend

2. Structura frontend-ului este organizată clar și logic, având ca punct central directorul src/, care reunește toate componentele necesare funcționării interfeței aplicației. Așa cum este ilustrat în Figura 6, codul este împărțit în module pentru pagini, funcționalități logice, testare, stilizare și inițializare. Această organizare facilitează atât dezvoltarea, cât și mentenanța aplicației, oferind o bază solidă pentru extinderea ulterioară a funcționalităților.

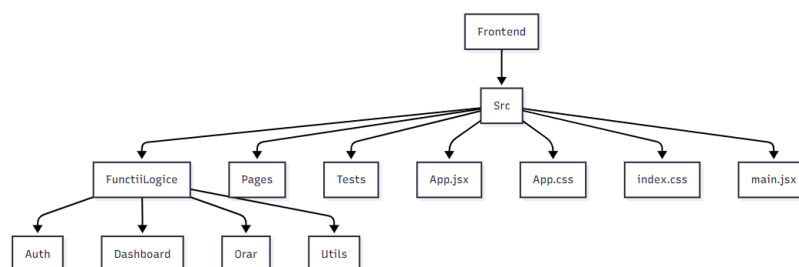


Fig. 6 Diagrama pachetelor- Frontend

3. Rutele din backend sunt organizate în mod modular, fiecare fișier având o responsabilitate clară în gestionarea unei funcționalități specifice. Așa cum se observă în Figura 7, toate aceste rute sunt centralizate în modulul routes, care le grupează și le integrează în aplicația principală. Această structurare permite o mai bună organizare a codului, o gestionare eficientă a cererilor API și o extensibilitate ușoară în cazul adăugării de noi funcționalități.

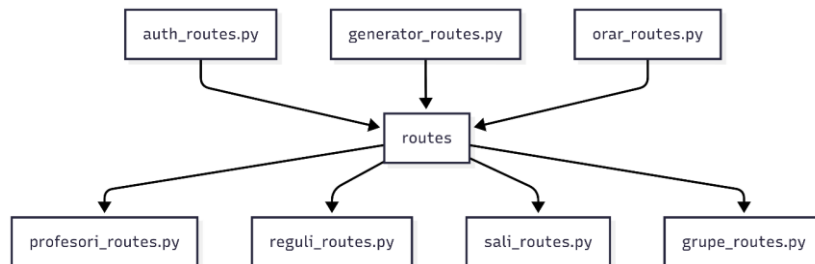


Fig. 7 Diagrama pachetelor- Routes

4. Directorul Pages reunește toate componentele vizuale principale ale aplicației, fiecare dintre ele corespunde unei funcționalități esențiale. După cum este ilustrat în Figura 8, aceste pagini sunt responsabile pentru interacțiunea directă cu utilizatorul – de la autentificare și administrare, până la generarea și vizualizarea orarului. Structura este clară și intuitivă, permițând o navigare fluentă între secțiunile aplicației și o experiență de utilizare coerentă.

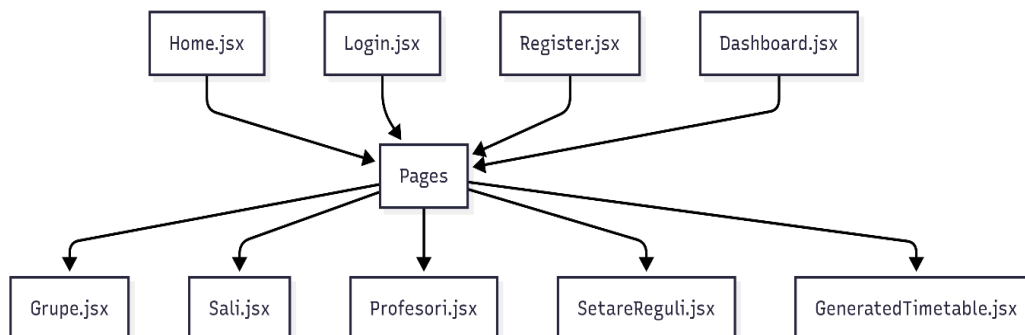


Fig. 8 Diagrama pachetelor- Pages

### 3.2.2 Diagrama claselor

Diagrama de clase oferă o vedere de ansamblu asupra componentelor logice implicate în generarea orarului și modul în care acestea interacționează. Așa cum este ilustrat în Figura 9, sunt reprezentate clasele principale ale backend-ului, împreună cu atributele și metodele relevante.

Structura evidențiază clasele responsabile de generarea orarului (fie prin algoritm AI, fie prin algoritm clasic), validarea acestuia și accesul la date. Relațiile dintre clase sunt marcate prin asocieri clare, ilustrând dependențele funcționale din cadrul sistemului. Acest model contribuie la o înțelegere clară a arhitecturii interne și oferă o bază solidă pentru extinderea ulterioară a funcționalităților.

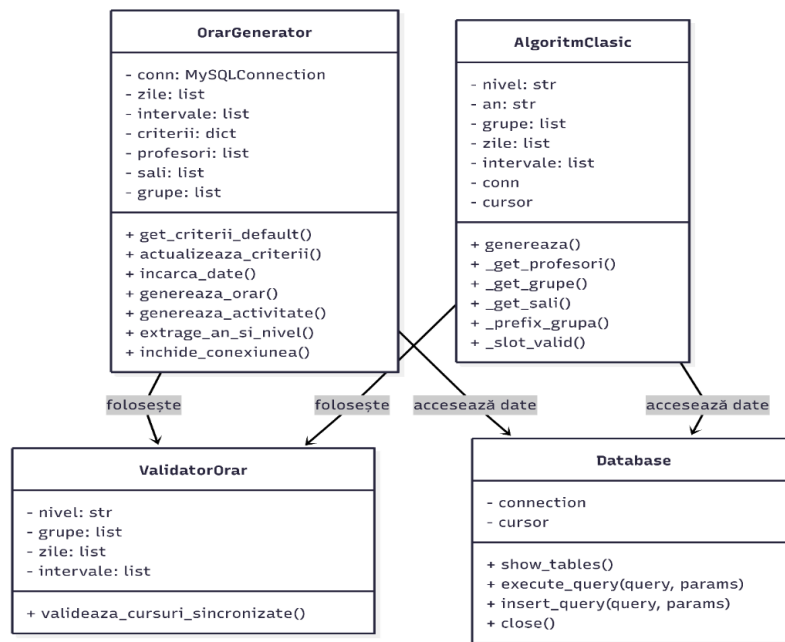


Fig. 9 Diagrama clasei -Backend

Figura 10 prezintă modelarea logică a componentelor frontend implicate în procesul de configurare a regulilor și generarea efectivă a orarului. Diagrama evidențiază relațiile dintre componentele vizuale React (precum SetareReguli și GeneratedTimetable) și hook-urile logice asociate (useSetariReguli, useGeneratedTimetable, useOrarGenerator etc.).

Fiecare element gestionează atribute și metode proprii, specifice rolului său în aplicație. Relațiile dintre componente sunt exprimate clar, prin interacțiuni precum utilizare, transmitere de reguli sau validare. Această structură modulară asigură o separare eficientă între interfață și logică, contribuind la claritatea și întreținerea codului în aplicația React.

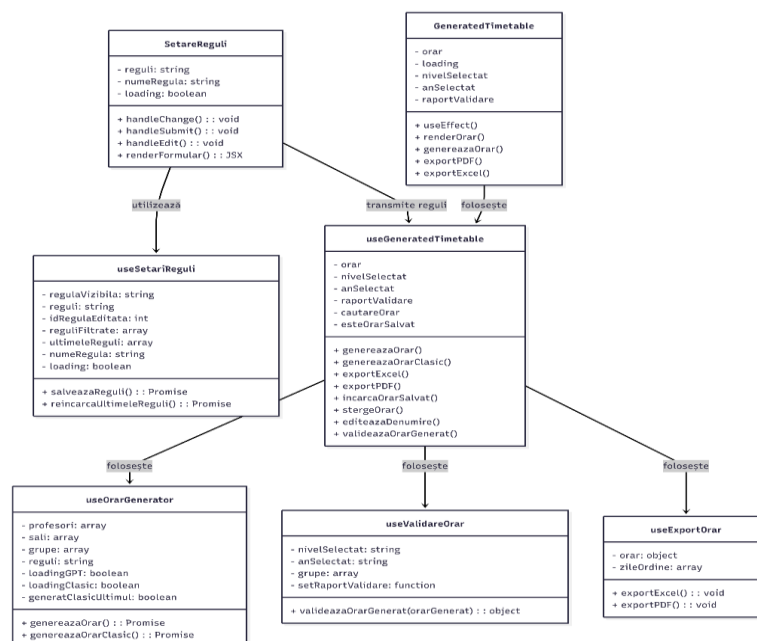


Fig. 10 Diagrama claselor- Frontend

### 3.2.3 Diagrama de componente

Diagrama prezentată în Figura 11 oferă o vedere de ansamblu asupra arhitecturii aplicației, concentrându-se pe conexiunile dintre componentele frontend, backend și baza de date. Se evidențiază fluxul de date de la interfața utilizatorului până la stocarea informațiilor, subliniind comunicarea dintre module.

Componentele React din frontend interacționează cu backend-ul Flask prin rute specifice, fiecare dintre acestea corespunde unei entități din baza de date MySQL. Fiecare cerere trimisă din interfață este gestionată de un modul backend care efectuează operații asupra unei tabele dedicate. Diagrama reflectă astfel organizarea clară a aplicației și separarea logică a responsabilităților.

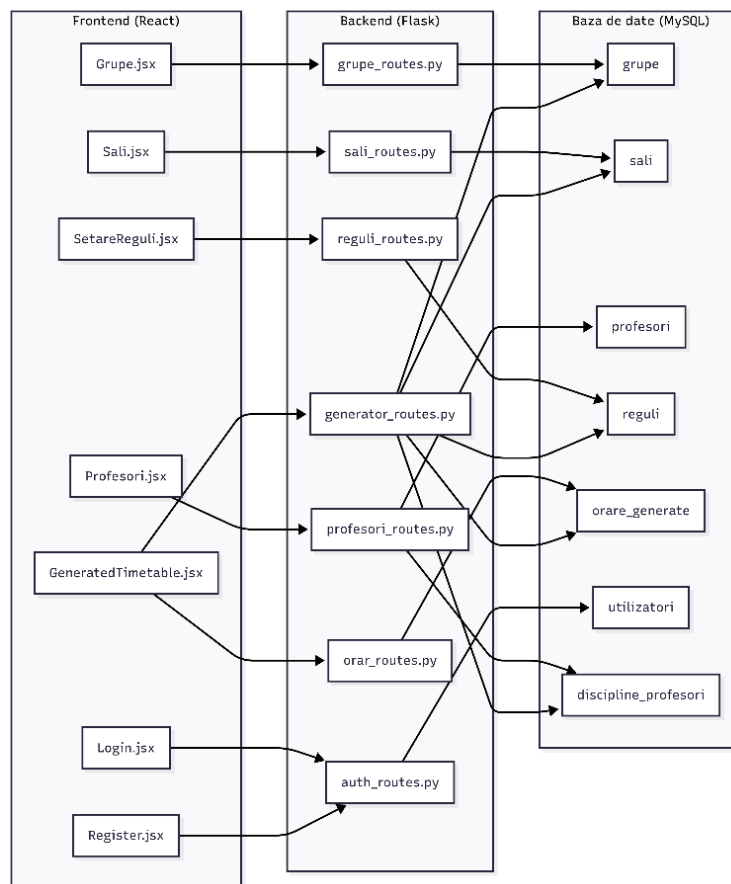


Fig. 11 Diagrama de componente ale aplicației

### 3.3 Diagrame UML pentru modelarea comportamentului

Pentru a surprinde comportamentul sistemului în raport cu utilizatorul și cu procesele interne, au fost utilizate mai multe diagrame UML comportamentale. Acestea evidențiază fluxurile logice, interacțiunile dinamice dintre componente și cazurile de utilizare ale aplicației.

### 3.3.1 Diagrama cazurilor de utilizare

Figura 12 ilustrează cazurile de utilizare asociate interacțiunii utilizatorului cu aplicația. Diagrama evidențiază principalele funcționalități accesibile acestuia, oferind o vedere de ansamblu asupra fluxului general de lucru.

Utilizatorul are posibilitatea să se autentifice, să configureze regulile necesare generării orarului, să declanșeze procesul de generare propriu-zis, să vizualizeze rezultatul și, la final, să exporte orarul în format PDF sau Excel. Aceste acțiuni acoperă întregul ciclu de utilizare al aplicației și reflectă funcționalitățile esențiale puse la dispoziție într-un mod intuitiv.

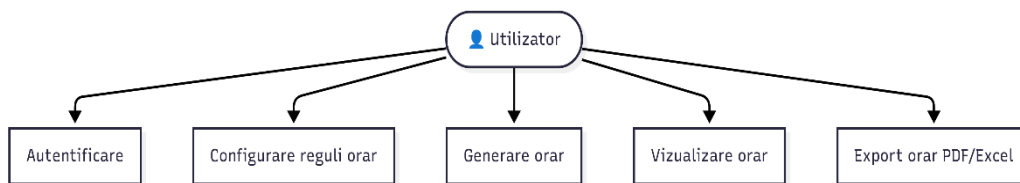


Fig. 12 Diagrama cazurilor de utilizare

### 3.3.2 Diagrama de secvență

Figura 13 ilustrează diagrama de secvență aferentă procesului de generare a orarului cu ajutorul modelului AI. Aceasta evidențiază fluxul cronologic de mesaje între principalele componente ale aplicației: utilizator, interfața React, backend-ul Flask, modulul de generare (OrarGenerator), API-ul OpenAI și baza de date MySQL.

Procesul începe cu acțiunea utilizatorului, care declanșează cererea de generare. Datele și regulile sunt procesate secvențial, transmise către modelul AI, iar răspunsul obținut este salvat și apoi returnat către frontend pentru afișare. Diagrama oferă o imagine clară asupra colaborării dintre module în cadrul acestui flux automatizat.

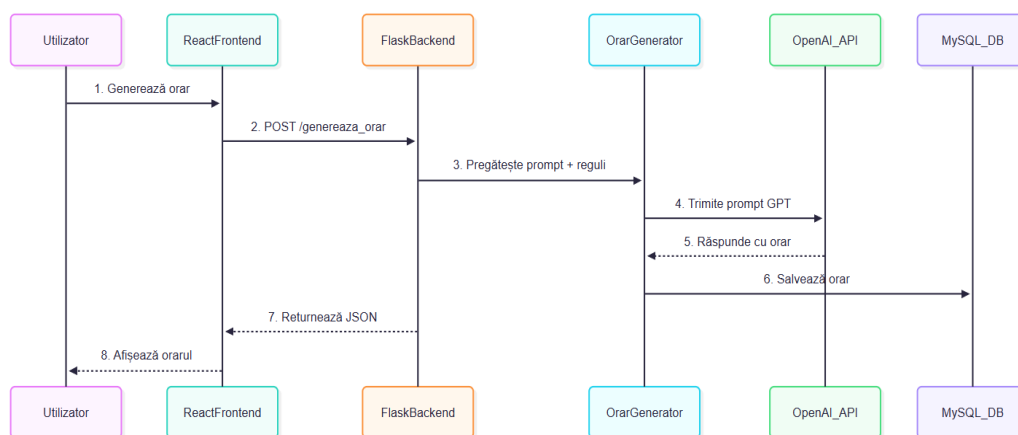


Fig. 13 Diagrama de secvență



### 3.3.3 Diagrama de activitate

Figura 14 prezintă diagrama de activitate care descrie, pas cu pas, logica funcțională a procesului de generare a orarului. Diagrama urmărește traseul parcurs de la inițierea cererii până la afișarea sau exportul rezultatului.

Procesul începe cu introducerea regulilor de către utilizator, continuă cu transmiterea acestora către backend, colectarea datelor din baza de date, generarea propriu-zisă a orarului (prin AI sau algoritm clasic) și validarea rezultatului. În funcție de validare, sistemul decide dacă orarul este afișat și poate fi exportat sau dacă se generează un mesaj de eroare. Acest model vizual evidențiază deciziile critice din fluxul aplicației și modul în care sunt gestionate rezultatele.

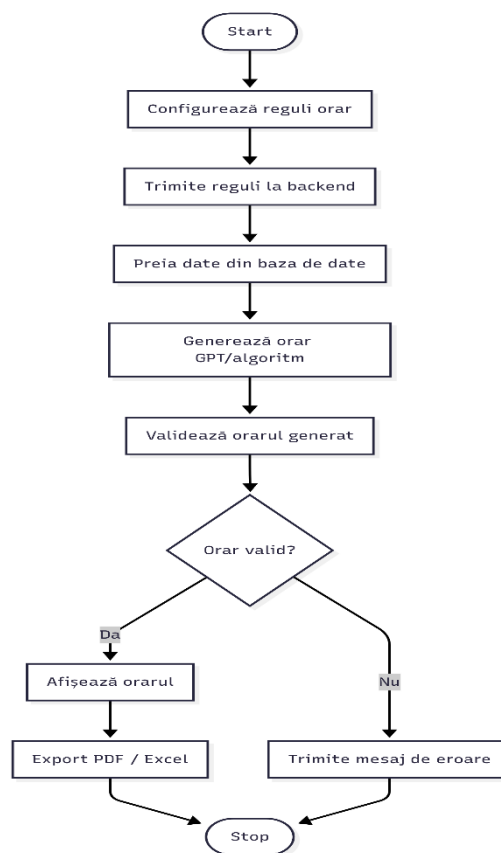


Fig. 14 Diagrama de activitate

## **CAPITOLUL 4. DENUMIREA CAPITOLULUI**

## **CONCLUZII**

Vor cuprinde într-o formă cât mai concisă principale rezultatele obținute în tema tratată, subliniindu-se contribuția adusă prin propriile cercetări. Se vor scoate în evidență elementele de noutate ale proiectului/lucrării. Dacă rezultatele obținute pot fi aplicate în activitatea de cercetare, producție sau în alte domenii de activitate, economică sau socială, se vor face recomandările corespunzătoare.

Un bilanț al aspectelor pozitive și negative din activitatea de dezvoltare a proiectului de diplomă sau a lucrării de disertație va încheia partea scrisă a lucrării.

## BIBLIOGRAFIE

Bibliografia cuprinde cărți, capitole din cărți, articole tipărite, articole și lucrări prezentate la conferințe și disponibile on line, site-uri.

Redactarea se va face aplicând criteriul alfabetic numelor autorilor. Dacă același autor a scris mai multe lucrări care apar în bibliografie, acestea se vor include în ordine cronologică, de la cea mai veche spre cea editată recent. Vor fi menționate: numele autorului/autorilor, titlul integral al lucrării, editura, locul publicării, anul apariției lucrării.

Pentru articolele apărute în reviste, se menționează în ordine: numele autorului/autorilor, titlul articolului, titlul revistei, ISBN/ISSN, anul apariției, numărul.

Paginile web se vor ordona alfabetic și vor avea menționată data accesării lor. Această precizare va fi făcută deoarece există posibilitatea ca pagina citată de pe internet să fie ulterior ștearsă.

- [1] I. Voncilă, D. Călușanu, N. Badea, R. Buhosu, Cr. Munteanu, „Mașini Electrice”, Editura Fundației Universitare „Dunărea de Jos” din Galați, 2003;
- [2] I. Șuşnea, M. Vlase, „A software instrument for the assessment of creativity in the educational environment”, The Annals of the University "Dunărea de Jos" of Galați Romania Fascicle III Electrotechnics Electronics Automatic Control Informatics, ISSN 1221-454X, 2016, Volum 39, Numărul 1;
- [3] <https://ro.wikipedia.org/wiki/Tranzistor>, 23-06-2016.

## **ANEXA 1**

Anexele (dacă există) – nu se numerează ca și capitol, se numerează crescător (Anexa 1, Anexa 2, etc).

Anexele vor conține elemente precum:

- porțiuni de cod;
- tabele de date;
- tabele de rezultate de ieșire;
- alte elemente specifice la care s-a făcut referire în cadrul proiectului/lucrării.