



Universitatea
Ştefan cel Mare
Suceava

Facultatea de Inginerie Electrică
și Știința Calculatoarelor

PROGRAM DE STUDIU CALCULATOARE

PROIECT DE DIPLOMĂ

SISTEM DE EVALUARE PRIVIND CALITATEA ACTULUI DIDACTIC

COORDONATOR ȘTIINȚIFIC,
conf. univ. dr. ing. DANUBIANU Mirela

ABSOLVENT,
DINCO Andreea-Elena

Suceava 2023

Cuprins

Lista abrevierilor	2
Lista figurilor	4
Sinteza lucrării	7
Capitolul 1. Introducere	9
Capitolul 2. Tehnologii folosite	11
2.1 Tehnica de calcul serverless	11
2.2 AWS Lambda	11
2.3 Amazon RDS	14
2.4 Servicii AWS gratuite	18
2.5 Amazon S3	18
2.6 Amazon API Gateway	20
2.7 AWS CloudFormation.....	22
2.8 Xamarin	23
2.9 Xamarin.Forms	25
2.10 ASP.NET Core	25
Capitolul 3. Proiectare și implementare	27
3.1 Motivația	27
3.2 Chestionarul	28
3.3 Interfețele aplicației	29
3.4 Aplicația Xamarin.Forms.....	30
3.5 Interfața administrator	30
3.6 Interfața conectare utilizator	32
3.7 Interfața student.....	33
3.8 Interfața profesor	37
3.9 Servicii API.....	40
3.10 Baza de date	42
3.11 Tabele	43
3.12 Proceduri stocate	45
3.12 AWS Serverless ASP.NET Core Web API	46
3.13 Configurări	46
3.14 Autorizare cu token JWT	49
Capitolul 4. Manual de utilizare	54
4.1 Perspectiva administratorului	54
4.2 Perspectiva studentului.....	57
4.3 Perspectiva profesorului	60
Capitolul 5. Concluzii și dezvoltări viitoare.....	63
Bibliografie	64

Lista abrevierilor

AWS - Amazon Web Services

API - Application Programming Interface

IT - Information Technology

Amazon S3 - Amazon Simple Storage Service

IaaS - Infrastructure-as-a-Service

PaaS - Platform-as-a-Service

SaaS - Software-as-a-Service

FaaS - Function-as-a-Service

UCP - Unitatea centrală de prelucrare

DDoS - Distributed Denial-of-Service

HTTP - Hypertext Transfer Protocol

IoT - Internet of Things

VPC - Virtual Private Cloud

Amazon ECS - Amazon Elastic Container Service

Amazon EC2 - Amazon Elastic Compute Cloud

SDK - Software Development Kit

AZ - Availability Zones

Amazon RDS - Amazon Relational Database Service

EFS - Elastic File System

ML - Machine Learning

IAM - Identity and Access Management

SOC - System and Organization Controls

HIPAA - The Health Insurance Portability and Accountability Act of 1996

PCI - Payment Card Industry

ISO - International Organization for Standardization

DB - Database

CLI - Command Line Interface

DNS - Domain Name System

SQL - Structured Query Language

RAM - Random Access Memory

SSD - Solid State Drive

IPv4 - Internet Protocol version 4

RHEL - Red Hat Enterprise Linux

SLES - SUSE Linux Enterprise Server

MFA - Multi-Factor Authentication

U2F - Universal 2nd Factor

CRR - Cross-Region Replication

SRR - Same-Region Replication

REST - Representational State Transfer

ALB - Application Load Balancer

NLB - Network Load Balancer

APN - AWS Partner Network

JSON - JavaScript Object Notation

YAML - YAML Ain't Markup Language

AWS CDK - AWS Cloud Development Kit

SAM - Serverless Application Model

IL - Intermediate Language

JIT - Just-in-Time

ART - Android Runtime

MCW - Managed Callable Wrappers

ACW - Android Callable Wrappers

AOT - Ahead-of-Time

ARM - Advanced RISC Machine

XAML - Extensible Application Markup Language

UWP - Universal Windows Platform

RPC - Remote Procedure Call

JWT - JSON Web Token

URI - Uniform Resource Identifier

HMAC - Hash-based Message Authentication Code

Lista figurilor

Figura 2.1 Diagramă ce prezintă procesul de declanșare a unei funcții Lambda

Figura 2.2 Diagramă ce prezintă modul în care funcționează Amazon S3

Figura 2.3 Diagramă ce ilustrează modul în care funcționează Amazon API Gateway

Figura 2.4 Diagramă ce ilustrează crearea unei stive în AWS CloudFormation

Figura 2.5 Arhitectura aplicațiilor Xamarin

Figura 2.6 Arhitectura aplicațiilor Xamarin.Forms

Figura 3.1 Diagrama cazurilor generală

Figura 3.2 Funcția de deconectare

Figura 3.3 Interfață pagina de administrare (1)

Figura 3.4 Interfață pagina de administrare (2)

Figura 3.5 Interfață pagina de conectare

Figura 3.6 Funcția de conectare a utilizatorului

Figura 3.7 Interfață chestionar (1)

Figura 3.8 Interfață chestionar (2)

Figura 3.9 Interfață chestionar (3)

Figura 3.10 Interfață chestionar (4)

Figura 3.11 Trimitere răspunsuri în urma evaluării unei activități didactice de la finalul semestrului

Figura 3.12 Trimitere răspunsuri în urma evaluării unei activități didactice din timpul semestrului

Figura 3.13 Interfață pagina de rapoarte (1)

Figura 3.14 Interfață pagina de rapoarte (2)

Figura 3.15 Creare diagrame cu linii

Figura 3.16 Diagramă cu linii

Figura 3.17 Personalizare diagramă cu linii

Figura 3.18 Serviciu API pentru conectare

Figura 3.19 Specificare token de acces la o solicitare API ce necesită autorizare

Figura 3.20 Serviciu API pentru ștergerea utilizatorilor

Figura 3.21 Diagrama claselor

Figura 3.22 Schema bazei de date

Figura 3.23 Procedură stocată pentru salvarea răspunsurilor în urma evaluării unei activități didactice la sfârșitul semestrului (1)

Figura 3.24 Procedură stocată pentru salvarea răspunsurilor în urma evaluării unei activități didactice la sfârșitul semestrului (2)

Figura 3.25 Publicare web API folosind AWS Lambda

Figura 3.26 Configurare aplicație pentru a putea fi publicată pe AWS Lambda

Figura 3.27 Conexiune baza de date

Figura 3.28 Configurare conexiune baza de date în appsettings.json

Figura 3.29 Configurări pentru gestionarea utilizatorilor și rolurilor

Figura 3.30 Permiterea autorizării și autentificării

Figura 3.31 Permite introducerea în header-ul solicitării a unui token JWT în interfața Swagger pentru autorizare

Figura 3.32 Permite autentificarea folosind un token JWT și validarea acestuia

Figura 3.33 Permite interacțiunea resurselor din domenii diferite

Figura 3.34 Creare token JWT

Figura 3.35 Verificare token reîmprospătare

Figura 3.36 Metodă logare utilizator ce verifică dacă credențialele trimise de către client sunt corecte și returnează un obiect TokenReimprospătare ce conține token-ul de acces nou generat, token-ul de reîmprospătare, identificatorul utilizatorului, numele de utilizator, data expirării token-urilor și rolul utilizatorului, în cazul în care credențialele sunt valide (1)

Figura 3.37 Metodă logare utilizator ce verifică dacă credențialele trimise de către client sunt corecte și returnează un obiect TokenReimprospătare ce conține token-ul de acces nou generat, token-ul de reîmprospătare, identificatorul utilizatorului, numele de utilizator, data expirării token-urilor și rolul utilizatorului, în cazul în care credențialele sunt valide (2)

Figura 3.38 Autorizare după rol

Figura 3.39 Metodă ce returnează un nou token de reîmprospătare și de acces clientului dacă token-ul trimis de acesta și cel existent în baza de date coincid

Figura 3.40 Diagrama claselor API

Figura 3.41 Metodă ce preia ciclurile de studii din baza de date și returnează o listă de obiecte AniDeStudiu

Figura 3.42 Metodă ce salvează în baza de date răspunsurile la întrebări oferite de studenți în cazul unei activități didactice din timpul semestrului și apelează o procedură stocată pentru a face asta, într-o operațiune asincronă

Figura 4.1 Interfață pagina de conectare și credențialele pentru un cont de student, profesor și administrator

Figura 4.2 Interfață din perspectiva administratorului (1)

Figura 4.3 Interfața din perspectiva administratorului (2)

Figura 4.4 Interfața din perspectiva administratorului (3)

Figura 4.5 Interfața din perspectiva administratorului (4)

Figura 4.6 Interfața din perspectiva studentului

Figura 4.7 Cuestionar complet (1)

Figura 4.8 Cuestionar complet (2)

Figura 4.9 Trimitere răspunsuri

Figura 4.10 Interfața din perspectiva profesorului

Figura 4.11 Introducere date

Figura 4.12 Afisare raport (1)

Figura 4.13 Afisare raport (2)

Figura 4.14 Afisare raport (3)

Sinteza lucrării

Am realizat acest proiect cu scopul de a veni în ajutor facultății și cadrelor didactice ce vor să își îmbunătățească prestația didactică, suportul de curs și materialele didactice utilizate la alte activități didactice din cadrul Facultății de Inginerie Electrică și Știința Calculatoarelor, cât și relația dintre cadrele didactice și studenți, o comunicare sinceră și constructivă dintre cei doi parteneri din procesul educațional fiind benefică pentru ambele părți ce au ca scop comun excelența academică și progresul continuu.

În cadrul acestui proiect am vrut să mă concentrez pe evaluarea materialelor didactice folosite de profesori la cursuri, laboratoare, seminare, proiecte, neluând în calcul evaluarea cadrului didactic și conduită acestuia în timpul activităților didactice.

În cadrul facultății există un sistem de evaluare al cadrelor didactice semestrială care se concentrează mai mult pe conduită și relația acestora cu studenții și mai puțin pe materialele didactice folosite, în special caracteristicile și funcționalitățile acestora.

Sistemul de evaluare din acest proiect este reprezentat de un chestionar ce va fi completat de către studenți după fiecare activitate didactică la care au fost prezenți, nefiind luate în considerare răspunsurile la activitățile la care nu au fost prezenți, asemănător cu sistemul de evaluare al profesorilor.

Ce este diferit la acest sistem de evaluare este faptul că evaluarea nu se va face semestrială, ci după fiecare activitate didactică. De asemenea, nu este o aplicație web în browser ci o aplicație mobilă Android și iOS. Utilizatorii acestei aplicații vor fi studenții, cadrele didactice și administratorul. Cele trei tipuri de utilizatori vor avea roluri diferite în cadrul aplicației și sunt bine delimitate. Conturile utilizatorilor vor fi deja create, asemănător cu cele folosite de studenți pe platforma Moodle, ceea ce înseamnă ca parola nu va putea fi modificată.

Rolul de student va presupune completarea unui chestionar ce va avea în vedere ciclul de studii al disciplinei ce dorește să o evalueze, anul de studiu în care se află, programul de studiu al disciplinei, disciplina, tipul de activitate la care a participat (seminar, curs, laborator, proiect), numărul activității/activităților pe care dorește să o/le evalueze (ex. cursul cu numărul șapte), activitățile la care a fost prezent, întrebări cu un singur răspuns, întrebări cu răspunsuri multiple, întrebări semantice în scară și întrebări deschise. Studentul va putea evalua doar o singură dată o anumită activitate didactică și nu va avea posibilitatea de a-și modifica răspunsurile.

Rolul de cadrul didactic (profesor) va reprezenta posibilitatea de a vizualiza un raport al evaluărilor făcute de studenți la orice disciplină din cadrul facultății. Aceasta va putea vedea toate răspunsurile studenților la întrebările cu răspuns liber și diagrame cu linii pentru întrebările semantice în scară, având o perspectivă mai bună asupra preferințelor studenților.

Rolul de administrator este reprezentat de gestionarea conturilor utilizatorilor, înregistrarea și ștergerea acestora.

În prezent nu există un sistem de evaluare privind calitatea actului didactic înglobat într-o aplicație mobilă, aceste evaluări fiind efectuate pe foi de hârtie sau într-o aplicație web în browser. Alte sisteme de evaluare al calității actului didactic vizează mai mult cadrul didactic și sistemul de examinare folosit de acesta, criteriile ce țin de materiale didactice nefiind evidențiate aşa bine.

Ca și realizări personale menționez dezvoltarea aplicației mobile Android și iOS folosind framework-ul Xamarin.Forms, crearea schemei bazei de date MySQL și găzduirea acesteia în cloud prin serviciul Amazon RDS și dezvoltarea API-ului prin framework-ul ASP.NET Core ce va asigura comunicarea dintre aplicația mobilă și baza de date, de asemenea găzduit în cloud pe platforma AWS Lambda.

În cadrul aplicației mobile am creat interfață grafică pentru pagina ce va afișa chestionarul, pagina de administrare a utilizatorilor, pagina de logare și pagina de rapoarte, mecanismul de conectare și deconectare a utilizatorului, metodele ce țin de preluarea datelor din chestionar și salvarea acestora în baza de date, metodele de prelucrare a răspunsurilor date de studenți și afișarea acestora sub forma unor grafice unde este cazul.

În baza de date am creat tabelele în care vor fi stocate date despre ciclurile de studii, programele de studiu, anii de studiu, disciplinele, tipurile de activități, întrebările din chestionar, răspunsurile date de studenți, datele despre evaluare, utilizatori, rolurile utilizatorilor, token-urile de acces și cele de reîmprospătare, atributele ce țin de roluri și utilizatori, migrațiile din cadrul proiectului și date despre conectarea utilizatorilor. De menționat că tabelele ce conțin date despre migrații, atributele rolurilor și utilizatorilor, conectarea utilizatorilor, rolurile și rolurile utilizatorilor, utilizatori și token-urile de acces ale acestora au fost generate automat de API-ul ASP.NET Core Identity pentru a implementa funcționalități de gestionare a utilizatorilor.

În API-ul web am creat metodele de preluare a datelor despre anii de studiu, ciclurile de studii, discipline, programe de studiu, tipurile de activități, rapoarte și metodele de salvare în baza de date a informațiilor despre utilizatorii noi înregistrați, datele de logare precum și acordarea token-urilor de acces și reîmprospătare și autorizarea și autentificarea utilizatorilor în funcție de rolul acestora.

Proiectul final reprezintă un sistem format din aceste trei componente interconectate între ele, sistem cu care interacționăm prin intermediul aplicației mobile ce se folosește de serviciile web care vor prelua și salva date din și în baza de date.

Prin intermediul acestui sistem de evaluare am reușit să efectuez evaluarea unei discipline din postura unui student și să afișez aceste date sub forma unor diagrame sau text din postura unui profesor ce ar putea fi folosite de acestia pentru a reflecta asupra prestației didactice personale și ce ar putea schimba pentru a îmbunătăți procesul educațional.

Capitolul 1. Introducere

În era digitală contemporană, aplicațiile mobile au devenit o parte indispensabilă a vieții noastre de zi cu zi, transformând modul în care comunicăm, lucrăm, învățăm și ne distrăm. Odată cu avansul rapid al tehnologiei, aplicațiile mobile au devenit instrumente puternice, care răspund la o gamă largă de nevoi și interese.

Dezvoltarea de aplicații mobile sunt esențiale pentru orice afacere online. Aceste aplicații pot ajuta clienții să interacționeze mai ușor și rapid cu serviciile oferite de companie, să fie notificați atunci când apar oferte noi (în cazul aplicațiilor de comerț) sau au primit un mesaj, un fișier sau un mesaj vocal în timp real.

Conform statisticilor, numărul de utilizatori ce dețin un telefon mobil a trecut de 6,3 miliarde în 2022 și se estimează că în 2028 numărul acestora va depăși 7,8 miliarde. Numărul utilizatorilor de aplicații mobile este în continuă creștere, piața aplicațiilor mobile având încă potențial să prospere și în următorii 5 ani [1].

Numărul utilizatorilor Android atinge 2,3 miliarde în 2023 și întrece numărul de utilizatori Windows, al doilea cel mai utilizat sistem de operare. În total numărul de utilizatori pe internet reprezintă mai mult de 60% din populația lumii [2]. Numărul de descărcări de aplicații mobile la nivel global a crescut considerabil din 2016 până acum, depășind 200 de miliarde în 2019 și 250 de miliarde în 2023, cu 80% mai mult față de cifrele înregistrate în 2016. Veniturile din aplicațiile mobile sunt estimate să atingă suma de 613 miliade de dolari în 2025, jocurile mobile fiind categoria care generează cel mai mare venit, în comparație cu celelalte categorii de aplicații [3].

În România, cea mai mare pondere a traficului web generat de telefoanele mobile provine de pe dispozitivele Android, dispozitivele iOS reprezentând aproximativ 25% din total [4]. Dispozitivele Android reprezentând majoritatea telefoanelor mobile folosite de români a fost și motivul pentru care m-am concentrat mai mult pe dezvoltarea sistemului de evaluare pentru acest sistem de operare, întâmpinând niște obstacole la testarea aplicației pe un dispozitiv iOS, cum ar fi necesitatea unui Mac sau plata înrolării în programul pentru dezvoltatori Apple.

Pentru acest sistem de evaluare am decis să folosesc servicii cloud pentru a stoca datele despre evaluări și răspunsurile studentilor și să creez un API pentru preluarea și stocarea acestor date cu Amazon Web Services. Avantajele utilizării serviciilor furnizate de AWS includ gratuitatea creării unei baze de date MySQL cu configurații minime dar satisfăcătoare pentru cerințele acestei aplicații și posibilitatea de a funcționa 24 de ore pe zi timp de 12 luni în cloud, și de asemenea ușurința cu care am putut crea și încărca un web API ASP.NET Core cu AWS Lambda fără a avea nevoie să gestionez un server, doar încărcând codul într-un bucket S3.

Calculul în cloud (în engleză „cloud computing”) a reprezentat o schimbare majoră în rândul companiilor când vine vorba de resursele IT folosite și costurile acestora. Serviciile de calcul livrate către clienți includ furnizarea de servere, memorie de stocare, baze de date, rețele, software și.a. Utilizarea acestor servicii înseamnă eliminarea costurilor ce ar fi fost dedicate componentelor hardware și software, instalarea și configurarea, mențenanța, consumul de energie electrică, cât și personalul ce s-ar fi ocupat de instalarea, configurarea și mențenanța acestora. Astfel, companiile se pot concentra pe dezvoltarea serviciilor și aplicațiilor proprii fără a fi nevoie de gestionarea infrastructurii. Performanța este asigurată de furnizorii de

servicii cloud care folosesc ultimele tehnologii și componente hardware de pe piață pentru a livra clienților o experiență plăcută și soluții fiabile nevoilor acestora.

Cele patru categorii în care se pot încadra serviciile cloud sunt: infrastructură ca serviciu (IaaS), platformă ca serviciu (PaaS), serverless și software ca serviciu (SaaS). Acestea mai sunt considerate ca o stivă a ceea ce presupune serviciile cloud, deoarece se construiesc una peste alta.

Cu infrastructura ca serviciu (IaaS), clienții pot închiria servere și mașini virtuale, memorie de stocare, sisteme de operare, rețele, de la un furnizor de servicii în cloud ce livrează aceste servicii după modelul de plată în care plătești doar ce folosești.

Platforma ca serviciu (PaaS) reprezintă serviciile de calcul în cloud care furnizează un mediu la cererea clienților pentru a dezvolta, testa, livra și gestiona aplicațiile software. PaaS a fost conceput pentru a ușura crearea de aplicații mobile sau aplicații web, eliminând mentenanța infrastructurii serverelor, memoriei de stocare, rețelei și bazelor de date.

Având multe caracteristici în comun cu PaaS, tehnicele de calcul serverless se preocupă de dezvoltarea funcționalităților aplicației fără a pierde timp cu gestionarea infrastructurii necesare. Furnizorul de servicii cloud se ocupă de configurarea, planificarea capacitații și gestionarea serverului. Arhitecturile serverless sunt foarte scalabile și se bazează pe răspunsul la evenimente, folosind resurse doar la apariția sau declanșarea unui eveniment.

Software-ul ca serviciu (SaaS) reprezintă o metodă de furnizare a aplicațiilor software, la cerere sau pe baza unui abonament. Astfel, furnizorii de cloud găzduiesc și se ocupă cu gestionarea aplicațiilor software și infrastructura de bază. Accesul la aceste aplicații se face de obicei dintr-un browser pe un dispozitiv mobil sau desktop [5].

Se estimează că cheltuielile utilizatorilor la nivel mondial pentru serviciile cloud publice vor crește cu 20,7%, până la 591,8 miliarde de dolari în 2023, față de 490,3 miliarde de dolari în 2022, conform celei mai recente programe de la Gartner. Aceasta este mai mare decât programele de creștere de 18,8% pentru 2022 [6].

Până în 2027, peste 50% dintre întreprinderi vor folosi platformele cloud din industrie pentru a-și accelera inițiativele de afaceri. Platformele cloud din industrie permit trecerea de la soluții generice la platforme concepute pentru a se potrivi specificului industriei utilizatorului [7].

Datorită climatului macroeconomic actual, liderii cloud spun că își sporesc utilizarea serviciilor și produselor bazate pe cloud (41,4%), planuiesc să migreze de la software-ul de întreprindere moștenit la instrumente bazate pe cloud (33,4%) și migrează sarcinile de lucru la nivel local către cloud (32,8%) [8].

În 2022, 93% dintre liderii din tehnologie au declarat că sunt „în mare parte cloud” într-o anumită formă – în creștere față de 83% în urmă cu doi ani – și 48% au spus că sunt „în mare parte hibride”, în creștere față de 40% în urmă cu doi ani. Între timp, numărul respondenților care au spus că sunt „în mare parte local” a scăzut la jumătate, până la 7% [9].

Tendințele și statisticile prezentate anterior sunt în favoarea adoptării soluțiilor cloud pentru majoritatea aplicațiilor care își doresc productivitate crescută, scalabilitate la nivel global, fiabilitate, viteză, performanță și securitate de top, fără grija gestionării unei infrastructuri.

Capitolul 2. Tehnologii folosite

2.1 Tehnica de calcul serverless

Tehnica de calcul serverless este o paradigmă în curs de dezvoltare în care furnizarea și scalarea resurselor unei aplicații sunt gestionate de servicii din terțe părți. Exemplele includ AWS Lambda, Azure Functions și Google Cloud Functions. În spatele API-urilor ușor de utilizat ale acestor servicii există infrastructuri complexe, opace și ecosisteme de gestionare.

Serviciile de calcul în cloud au permis menținării infrastructurii backend să devină din ce în ce mai decuplată de dezvoltarea aplicației. Tehnica de calcul serverless (sau function-as-a-service, FaaS) este o arhitectură de implementare a aplicațiilor în curs de dezvoltare care ascunde complet gestionarea serverelor față de clienți. Clienții primesc acces minim la configurația unei aplicații în timpul rulării. Acest lucru permite clientilor să se concentreze pe dezvoltarea funcțiilor lor — aplicații de dimensiuni mici dedicate unei anumite sarcini. O funcție se execută de obicei într-o instanță a funcției (un container de exemplu) cu resurse limitate, cum ar fi timpul UCP și memoria.

Spre deosebire de mașinile virtuale folosite pe platforme infrastructure-as-a-service (IaaS) tradiționale, o instanță a funcției va fi lansată numai atunci când funcția este invocată și este oprită după ce s-a ocupat de solicitare. Clienții sunt taxatați pe bază de invocare, fără a plăti pentru resursele neutilizate.

Tehnica de calcul serverless a apărut ca un model pentru gestionarea sarcinilor de lucru scurte, cum ar fi procesările ce apar ca răspuns la modificările rare ale fișierelor stocate în cloud. Acum este folosit ca model de programare simplu pentru o varietate de aplicații. Ascunderea gestionării resurselor față de clienți e ideea din spatele acestui model, dar opacitatea rezultată împiedică adoptarea de către mulți potențiali utilizatori, care și-au exprimat temerile ce vizează: securitatea în ceea ce privește calitatea izolării, rezistență la DDoS și multe altele; nevoia de a înțelege gestionarea resurselor pentru a îmbunătăți performanța aplicațiilor și abilitatea platformelor de a oferi această performanță [10].

2.2 AWS Lambda

AWS Lambda reprezintă un serviciu de calcul în cloud care permite rularea codului fără a necesita servere, ca răspuns la un eveniment și gestionează în mod automat resursele de calcul. Aceste evenimente pot include modificări de stare sau o actualizare, cum ar fi un utilizator care plasează un articol într-un coș de cumpărături pe un site de comerț electronic. AWS Lambda poate fi folosit pentru a extinde alte servicii AWS cu o logică furnizată de client sau se poate implementa propriile servicii de backend. AWS Lambda rulează automat cod ca răspuns la mai multe evenimente, cum ar fi solicitări HTTP prin Amazon API Gateway, modificări ale obiectelor din compartimentele Amazon Simple Storage Service (Amazon S3), actualizări de tabel în Amazon DynamoDB și tranzacții de stare în AWS Step Functions [12].



Figura 2.1 Diagramă ce prezintă procesul de declanșare a unei funcții Lambda [11]

Cu Lambda este necesar doar furnizarea codului într-o din limbajele acceptate de acesta deoarece acest serviciu se ocupă de administrarea infrastructurii de calcul și jurnalizarea.

Codul este organizat în funcții Lambda. Serviciul Lambda rulează funcția numai atunci când este necesar și se scalează automat. Plata acestui serviciu se face doar când se execută codul.

Lambda este un serviciu de calcul potrivit pentru scenariile în care aplicațiile trebuie să se extindă rapid și să reducă consumul până la zero atunci când nu sunt solicitate. Aceste servicii pot fi folosite pentru:

- procesarea fișierelor: se utilizează Amazon Simple Storage Service (Amazon S3) pentru a declanșa procesarea datelor în timp real după o încărcare.
- procesarea fluxului: se utilizează Lambda și Amazon Kinesis pentru a procesa date în flux în timp real pentru urmărirea activității aplicațiilor, procesarea comenziilor de tranzacție, analiza fluxului de clic, curățarea datelor, filtrarea jurnalelor, indexarea, analiza rețelelor sociale, telemetria datelor dispozitivelor Internet of Things (IoT) și contorizare.
- aplicații web: se combină Lambda cu alte servicii AWS pentru a crea aplicații web puternice, care se scalează automat și care rulează într-o configurație optimă de performanță în mai multe centre de date.
- backend-uri pentru aplicațiile IoT: se pot crea backend-uri fără server folosind Lambda pentru a gestiona solicitările API de pe web, mobil, IoT și terțe părți.
- backend-uri pentru aplicațiile mobile: se pot crea backend-uri folosind Lambda și Amazon API Gateway pentru a autentifica și procesa solicitările API. Se utilizează AWS Amplify pentru a se integra cu ușurință clienți iOS, Android, Web și React Native.

Utilizatorul acestor servicii este responsabil doar de codul său. Deoarece Lambda gestionează resursele de calcul și infrastructura, nu este posibilă personalizarea sistemelor de operare sau accesarea instanțelor de calcul.

Lambda efectuează activități operaționale și administrative, inclusiv gestionarea capacitatei, monitorizarea și jurnalizarea funcțiilor Lambda.

Pentru gestionarea resurselor de calcul, AWS are alte servicii de calcul de luat în considerare, cum ar fi:

- AWS App Runner construiește și implementează automat aplicații web înglobate, echilibrează încărcarea traficului prin criptare, se scalează pentru a satisface nevoile utilizatorului de trafic și permite configurarea modului în care serviciile sunt accesate și comunicarea cu alte aplicații AWS într-un Amazon VPC privat.

- AWS Fargate împreună cu Amazon ECS rulează containere fără a fi nevoie să furnizeze, să configureze sau să scaleze clustere de mașini virtuale.
- Amazon EC2 permite configurarea sistemului de operare, setările de rețea și de securitate și întreaga stivă de programe. Utilizatorul este responsabil pentru furnizarea capacitatii, monitorizarea sănătății și performanței flotei și utilizarea zonelor de disponibilitate pentru a opera fără întreruperi [13].

Cu AWS Lambda, se pot folosi limbajele, instrumentele sau framework-urile deja existente. Se poate utiliza orice bibliotecă din terțe părți, chiar și pe cele native. De asemenea, se poate împacheta orice cod (framework-uri, SDK-uri, biblioteci și multe altele) ca strat Lambda și se pot gestiona și partaja cu ușurință prin mai multe funcții. Lambda acceptă nativ Java, Go, PowerShell, Node.js, C#, Python și Ruby și oferă un API Runtime care permite utilizarea oricăror limbaje de programare suplimentare pentru a crea funcțiile.

AWS Lambda menține capacitatea de calcul în mai multe zone de disponibilitate (AZ) din fiecare regiune AWS pentru a proteja codul împotriva defecțiunilor individuale ale mașinii sau ale centrului de date. Atât AWS Lambda, cât și funcțiile care rulează pe acest serviciu oferă performanțe operaționale previzibile și fiabile. AWS Lambda este conceput pentru a oferi disponibilitate ridicată atât pentru serviciul în sine, cât și pentru funcțiile pe care le operează. Nu există intervale de menenanță sau pauze programate.

AWS Lambda suportă împachetarea funcțiilor și implementarea lor ca imagini container, facilitând crearea de aplicații bazate pe Lambda, utilizând instrumente, procese și dependențe existente pentru imagini container. Clienții beneficiază, de asemenea, de simplitatea operațională Lambda, scalarea automată cu timpi de pornire sub o secundă, disponibilitatea ridicată, modelul de facturare în care plătești doar ce folosești și integrările native cu peste 200 de servicii AWS și aplicații software-as-a-service (SaaS). Întreprinderile pot folosi un set de instrumente atât cu aplicațiile lor Lambda, cât și cu aplicațiile containerizate, simplificând gestionarea acestora, cum ar fi scanarea de securitate și adăugarea semnăturilor digitale.

AWS Lambda rulează codul doar la apariția unui eveniment și se scalează automat pentru a suporta rata de solicitări primite fără nicio configurație manuală. Nu există limită pentru numărul de solicitări pe care codul le poate gestiona. AWS Lambda începe de obicei să ruleze codul în câteva milisecunde de la apariția unui eveniment. Deoarece Lambda se scalează automat, performanța rămâne constantă pe măsură ce frecvența evenimentelor crește. Deoarece codul nu are o stare anume, Lambda poate porni câte instanțe sunt necesare, fără întârzieri îndelungate de instalare și configurare.

Se poate utiliza Amazon RDS Proxy pentru a profita de pool-urile de conexiuni complet gestionate pentru bazele de date relaționale. RDS Proxy gestionează eficient mii de conexiuni concurente la baze de date relaționale, facilitând dezvoltarea de aplicații serverless, foarte scalabile și sigure, bazate pe Lambda, care interacționează cu bazele de date relaționale. În prezent, RDS Proxy oferă suport pentru MySQL și Aurora.

Concurența provizionată (Provisioned Concurrency) oferă un control mai mare asupra performanței aplicațiilor serverless. Când este pornit, Provisioned Concurrency păstrează funcțiile inițializate și pregătite să răspundă în doar câteva zeci de milisecunde. Concurența provizionată este ideală pentru orice aplicație AWS Lambda care necesita un control mai mare asupra timpului de pornire a funcției. Se poate profita de concurența provizionată pentru a obține performanțe consistente pentru aplicațiile sensibile la latență fără a modifica codul sau a gestiona resursele de calcul.

Cu Amazon Elastic File System (EFS) pentru AWS Lambda, se pot citi, scrie și asigura persistența în siguranță a volume mari de date la latență scăzută, la orice scară. Nu este nevoie de cod sau de descărcarea datelor în stocarea temporară pentru a le procesa. Acest lucru economisește timp și simplifică codul, astfel încât clienții să se concentreze doar asupra aplicației. EFS pentru Lambda este ideal pentru o serie de cazuri de utilizare, inclusiv procesarea sau copierea de rezervă a unor cantități mari de date și încărcarea fișierelor sau modelelor de referință mari. De asemenea, se pot partaja fișiere între instanțe serverless sau aplicații bazate pe containere și chiar se pot rula inferențe de învățare automată (ML) folosind EFS pentru AWS Lambda.

Cu Lambda@Edge, AWS Lambda poate rula codul în locațiile AWS la nivel global, ca răspuns la evenimentele Amazon CloudFront, cum ar fi solicitările de conținut către sau de la serverele ce se ocupă de cererile de la vizitatorii unui site web. Acest lucru facilitează furnizarea de conținut mai bogat și mai personalizat utilizatorilor finali, cu o latență mai mică.

Se pot crea procese folosind AWS Step Functions pentru a coordona mai multe funcții AWS Lambda pentru sarcini complexe sau de lungă durată. Step Functions permite definirea proceselor care declanșează o colecție de funcții Lambda utilizând pași secvențiali, paraleli, ramificați și de gestionare a erorilor. Cu Step Functions și Lambda, se pot construi procese ce rețin stările anterioare, pe lungă durată, pentru aplicații și backend.

Kitul de dezvoltare software (SDK) încorporat al AWS Lambda se integrează cu AWS Identity and Access Management (IAM) pentru a asigura accesul securizat la alte servicii AWS. AWS Lambda rulează codul într-un Amazon Virtual Private Cloud (VPC) în mod implicit. Opțional, se poate configura accesul la resursele AWS Lambda în spatele propriului VPC pentru a utiliza grupuri de securitate personalizate și liste de control al accesului la rețea. Acest lucru oferă acces securizat pentru funcția Lambda la resursele dintr-un VPC. AWS Lambda este compatibil cu SOC, HIPAA, PCI și ISO.

Semnarea digitală a codului pentru AWS Lambda permite verificarea faptului că numai codul nealterat publicat de dezvoltatorii aprobați este implementat în funcțiile Lambda ale clienților. Se creează artefacte de cod semnate digital și se configerează funcțiile Lambda pentru a verifica semnăturile la implementare. Acest lucru crește viteza și agilitatea dezvoltării aplicațiilor, chiar și în cadrul unor echipe mari, impunând în același timp standarde înalte de securitate [12].

2.3 Amazon RDS

Amazon Relational Database Service (Amazon RDS) este un serviciu web în cloud care ușurează configurarea, operarea și scalarea unei baze de date relaționale în AWS Cloud. Furnizează o capacitate rentabilă, redimensionabilă pentru o bază de date relațională standard și gestionează sarcinile de administrare a bazei de date.

Amazon Elastic Compute Cloud (Amazon EC2) furnizează capacitate de calcul în cloud scalabilă în AWS Cloud.

Când cumpărăm un server local, UCP-ul, memoria, spațiul de stocare și operațiile de intrare/ieșire pe secundă sunt toate împreună. Cu Amazon EC2, acestea sunt împărțite astfel încât să poată fi scalate independent.

Pentru o bază de date relațională într-un server local, toată responsabilitatea pentru server, sistem de operare și software revine utilizatorului. Pentru o bază de date pe o instanță Amazon EC2, AWS gestionează straturile de sub sistemul de operare. În acest fel, Amazon EC2 elimină o parte din sarcina administrării unui server de baze de date local.

Amazon EC2 nu este un serviciu complet gestionat. Astfel, atunci când o bază de date este rulată pe Amazon EC2, predispunerea la erori de utilizator crește. De exemplu, atunci când se actualizează manual sistemul de operare sau software-ul bazei de date, este posibil să se cauzeze accidental o perioadă de întrerupere a aplicației. Identificarea și remedierea problemelor poate dura foarte mult.

Amazon RDS este un serviciu de baze de date gestionate. Este responsabil pentru majoritatea sarcinilor de gestionare.

Amazon RDS oferă următoarele avantaje specifice față de implementările de baze de date care nu sunt gestionate complet:

- se pot utiliza baze de date populare precum: MariaDB, Microsoft SQL Server, MySQL, Oracle și PostgreSQL.
- Amazon RDS gestionează copiile de rezervă, actualizările software, detectarea automată a erorilor și recuperarea datelor.
- se pot activa copiile de rezervă automate sau se pot crea manual propriile copii de rezervă. Se pot utiliza aceste copii de siguranță pentru a restaura o bază de date. Procesul de restaurare Amazon RDS funcționează în mod fiabil și eficient.
- se pot obține performanțe optime ridicate cu o instanță principală și o instanță secundară sincronă la care se poate trece atunci când apar probleme. De asemenea, se pot utiliza copii de citire pentru a crește scalarea citirii datelor.
- pe lângă securitatea din pachetul de baze de date, se poate controla cine poate accesa bazele de date RDS. Pentru a face acest lucru, se poate utiliza AWS Identity and Access Management (IAM) pentru a defini utilizatorii și permisiunile. De asemenea, se poate proteja baza de date într-un cloud privat virtual (VPC) [14].

O instanță DB este un mediu de baze de date izolat în AWS Cloud. Elementul de bază al Amazon RDS este instanța DB.

Instanța DB poate conține una sau mai multe baze de date create de utilizator. Se poate accesa instanța DB folosind instrumente și aplicații care sunt utilizate cu o instanță de bază de date independentă.

Fiecare instanță DB are un identificator de instanță DB. Acest nume furnizat de utilizator identifică în mod unic instanța DB atunci când interacționează cu API Amazon RDS și cu comenzi AWS CLI. Identificatorul instanței DB trebuie să fie unic pentru acel client într-o regiune AWS.

Identificatorul este utilizat ca parte a numelui de gazdă DNS alocat instanței utilizatorului de către RDS. De exemplu, dacă se specifică db1 ca identificator al instanței DB, atunci RDS va aloca automat un endpoint DNS pentru instanța respectivă, cum ar fi db1.123456789012.us-east-1.rds.amazonaws.com, unde 123456789012 este identificatorul fix pentru o regiune specifică pentru contul utilizatorului.

Fiecare instanță DB are un motor de bază de date. Amazon RDS acceptă în prezent motoarele de baze de date MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server și Amazon Aurora.

La crearea unei instanțe DB, unele motoare de baze de date necesită specificarea unui nume de bază de date. O instanță DB poate găzdui mai multe baze de date sau o singură bază de date Oracle cu mai multe scheme. Valoarea numelui bazei de date depinde de motorul bazei de date:

- pentru motoarele de baze de date MySQL și MariaDB, numele bazei de date este numele unei baze de date găzduite în instanța DB. Bazele de date găzduite de aceeași instanță DB trebuie să aibă un nume unic în cadrul instanței respective.
- pentru motorul de baze de date Oracle, numele bazei de date este utilizat pentru a stabili valoarea ORACLE_SID, care trebuie furnizată la conectarea la instanța Oracle RDS.
- pentru motorul bazei de date Microsoft SQL Server, numele bazei de date nu este un parametru acceptat.
- pentru motorul de baze de date PostgreSQL, numele bazei de date este numele unei baze de date găzduite în instanța DB. Un nume de bază de date nu este necesar la crearea unei instanțe DB. Bazele de date găzduite de aceeași instanță DB trebuie să aibă un nume unic în cadrul acelei instanțe.

Amazon RDS creează un cont de utilizator principal pentru instanța unei de baze de date ca parte a procesului de creare. Acest utilizator principal are permisiuni de a crea baze de date și de a efectua operații de creare, ștergere, selectare, actualizare și inserare a tabelelor pe care le creează utilizatorul principal. Trebuie setată parola utilizatorului principal atunci când se creează o instanță DB. De asemenea, se poate modifica parola utilizatorului principal și se pot gestiona utilizatorii utilizând comenzi SQL standard.

Clasa unei instanțe DB determină capacitatea de calcul și de memorie a unei instanțe DB Amazon RDS. Clasa unei instanțe DB de care poate avea nevoie un utilizator depinde de puterea de procesare și de cerințele de memorie.

Clasa unei instanțe DB este formată atât din tipul de clasă de instanță DB, cât și din dimensiunea acesteia. De exemplu, db.r6g este un tip de clasă a unei instanțe DB cu memorie optimizată, alimentată de procesoarele AWS Graviton2. În cadrul tipului de clasă a unei instanțe db.r6g, db.r6g.2xlarge este clasa instanței DB. Dimensiunea acestei clase este 2xlarge.

Amazon RDS acceptă clase ale unei instanțe DB pentru următoarele cazuri de utilizare:

- uz general
- memorie optimizată
- performanțe explozive doar când cererile sunt foarte mari

Sunt disponibile următoarele tipuri de clase ale unei instanțe DB cu performanță explozivă în vârfuri:

- db.t4g - Clase ale unei instanțe DB de uz general alimentate de procesoarele AWS Graviton2 bazate pe Arm. Aceste clase oferă o performanță mai bună raportat la preț față de clasele anterioare cu performanțe explozive pentru un set larg de sarcini de lucru mari de uz general. Instanțele Amazon RDS T4g sunt configurate pentru modul Nelimitat. Acest lucru înseamnă că acestea își pot crește performanța dincolo de limita de bază într-un interval de 24 de ore, pentru o taxă suplimentară.

- db.t3 – Clase ale unei instanțe DB care oferă un nivel de performanță de bază, cu posibilitatea de a ajunge la utilizarea completă a UCP-ului. Instantele T3 sunt configurate pentru modul Nelimitat. Aceste clase oferă o capacitate de calcul mai mare decât clasele unei instanțe db.t2. Ele sunt alimentate de sistemul AWS Nitro, o combinație de hardware dedicat și un hipervizor cu cerințe mai mici.
- db.t2 - Clase care oferă un nivel de performanță de bază, cu posibilitatea de a ajunge la utilizarea completă a UCP-ului. Aceste clase ale unei instanțe DB sunt recomandate numai pentru serverele de dezvoltare și de testare sau alte servere care nu sunt folosite în producție [15].

În acest proiect am folosit o clasă a unei instanțe DB de tip db.t2 ce se încadrează în ultimul caz de utilizare de mai sus deoarece este acoperit de pachetul gratis oferit de AWS.

Fiecare locație a unui centru de date ce găzduiește resurse de calcul în cloud se numește regiune AWS.

Fiecare regiune AWS conține mai multe locații distincte numite zone de disponibilitate, sau AZ-uri în engleză. Fiecare zonă de disponibilitate este proiectată pentru a fi izolată de defecțiunile din alte zone de disponibilitate. Fiecare zonă este proiectată pentru a oferi conectivitate la prețuri mici și cu latență redusă la alte zone de disponibilitate din aceeași regiune AWS. Prin lansarea instanțelor în zone de disponibilitate separate, aplicațiile vor fi protejate de defecțiunea unei singure locații.

O instanță DB poate fi activă în mai multe zone de disponibilitate. Atunci când se alege această opțiune, Amazon furnizează și menține automat una sau mai multe instanțe DB secundare de rezervă într-o altă zonă de disponibilitate. Instanța DB primară este replicată prin zonele de disponibilitate către fiecare instanță DB secundară. Într-un cluster DB Multi-AZ, instanțele DB secundare pot servi, de asemenea, traficului de citire de date.

Baza de date din acest proiect are următoarele specificații:

- motor bază de date MySQL Community
- clasa db.t2.micro
- se află în regiunea AWS eu-central-1a (Frankfurt)
- versiune motor instanță 8.0.28
- un UCP virtualizat
- 1 GB RAM
- Memorie de stocare de 10 GiB (SSD uz general)
- fără criptare a datelor
- fără autoscalare a memoriei de stocare
- fără copii de rezerve automate
- este publică
- protocol IPv4
- motor de stocare InnoDB

Am considerat crearea unei baze de date cu aceste configurații deoarece este acoperit de un cont gratis fără costuri lunare timp de 12 luni.

2.4 Servicii AWS gratuite

Pachetul gratis oferit de AWS presupune:

- Amazon EC2 (12 luni gratuitate): 750 de ore pe lună de calcul în cloud redimensionabil pentru o instanță t4g.small (deinde de regiune). 750 de ore pe lună pentru o instanță Linux, RHEL sau SLES t2.micro/t3.micro (deinde de regiune), 750 de ore pe lună pentru o instanță Windows t2.micro/t3.micro (deinde de regiune).
- Amazon S3 (12 luni gratuitate): stocare securizată, de durată și scalabilă a obiectelor de 5 GB, 20000 de solicitări de preluare a datelor și 2000 de solicitări de actualizare a datelor.
- Amazon RDS (12 luni gratuitate): 750 de ore pe lună pentru instanțe Amazon RDS MySQL, MariaDB, PostgreSQL de clasă db.t2.micro, db.t3.micro și db.t4g.micro, 20 GB de memorie de stocare (SSD), 20 GB de memorie de stocare pentru copiile de rezervă și copiile stării sistemului.
- AWS Lambda (gratis oricând): servicii de calcul care rulează codul ca răspuns la evenimente și care gestionează automat resursele de calcul; un milion de cereri pe lună gratis și 3,2 milioane de secunde de timp de calcul pe lună.
- Amazon API Gateway (12 luni gratuitate): publică, monitorizează, menține și securizează API-uri la orice scară, permite un milion de cereri API pe lună.
- AWS CloudFormation (gratis oricând): modeleză, furnizează și gestionează resurse AWS sau din terță părți tratând infrastructura ca și cod, permitând 1000 de operații pe lună pe cont, operațiile ce depășesc această limită vor fi taxate cu 0,0009 dolari pe operație și 0,00008 dolari pe operațiile care durează mai mult de 30 de secunde [16].
- s.a.m.d

2.5 Amazon S3

Amazon S3 are diverse funcții care pot fi utilizate pentru a organiza și gestionă datele în moduri care să accepte cazuri de utilizare specifice, să permită eficiența costurilor, să impună securitatea și să îndeplinească cerințele de conformitate. Datele sunt stocate ca obiecte în resurse numite „găleți”, iar un singur obiect poate avea o dimensiune de până la 5 teraocteți [17].

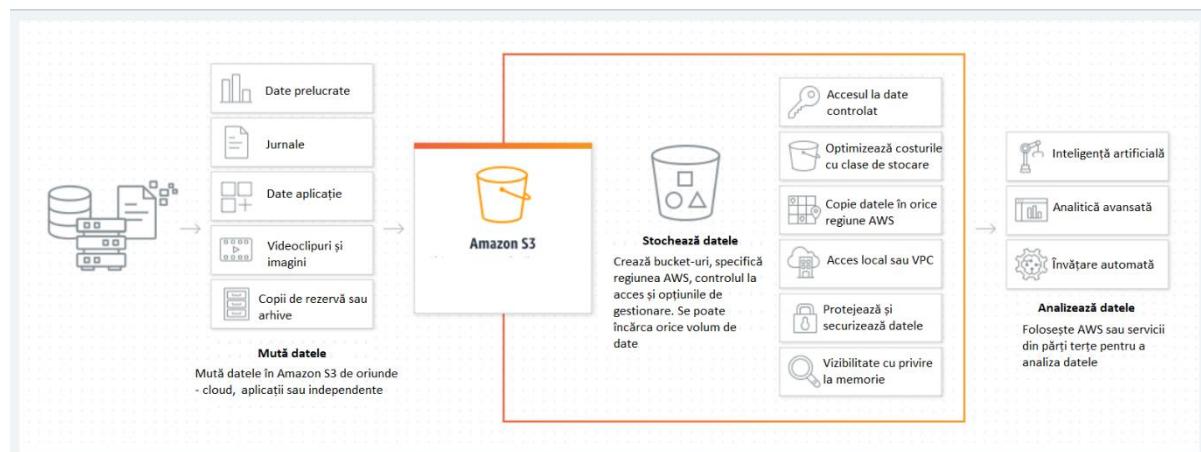


Figura 2.2 Diagramă ce prezintă modul în care funcționează Amazon S3 [18]

Diagrama arată cum sunt mutate datele în Amazon S3, cum sunt gestionate datele stocate în Amazon S3 și analiza datelor cu alte servicii. Procesul este împărțit în trei secțiuni.

Caracteristicile S3 includ capabilități de a adăuga etichete ce conțin metadate despre obiecte, de a muta și de a stoca date în clasele de stocare S3, de a configura și de a aplica controale de acces la date, de a securiza datele împotriva utilizatorilor neautorizați, de a rula analize de date mari, de a monitoriza datele la nivel de obiect și de comportament și de a vizualiza utilizarea stocării și tendințele de activitate într-o organizație. Obiectele pot fi accesate prin punctele de acces S3 sau direct prin numele de gazdă al grupului.

Structura simplă, neierarhică a Amazon S3 și diverselor funcții de gestionare ajută clienții de toate dimensiunile și industriile să își organizeze datele în moduri care sunt valoroase pentru afaceri și echipe. Toate obiectele sunt stocate în compartimente S3 și pot fi organizate cu nume partajate numite prefixe.

De asemenea, se pot adăuga până la 10 perechi cheie-valoare numite etichete de obiect S3 la fiecare obiect, care pot fi create, actualizate și șterse pe parcursul ciclului de viață al unui obiect.

Pentru a ține evidența obiectelor și a etichetelor, compartimentelor și prefixelor acestora, se poate utiliza un raport de inventar S3 care listează obiectele stocate într-un compartiment S3 sau cu un prefix specific, precum și metadatele și starea de criptare ale acestora. S3 Inventory poate fi configurat pentru a genera rapoarte zilnic sau săptămânal.

Cu numele grupelor S3, prefixele, etichetele obiectelor și Inventarul S3, există o gamă largă de moduri de a clasifica și raporta datele și, ulterior, se pot configura alte caracteristici S3 pentru a lua măsuri. Indiferent dacă se stochează mii de obiecte sau un miliard, S3 Batch Operations simplifică gestionarea datelor în Amazon S3 la orice scară.

Cu S3 Batch Operations, se pot copia obiecte între bucket-uri, se pot înlocui seturi de etichete de obiecte, se pot modifica controalele de acces și se pot restaura obiecte arhivate din clasele de stocare S3 Glacier Flexible Retrieval și S3 Glacier Deep Archive, cu o singură solicitare S3 API sau câțiva pași în Consola S3.

De asemenea, se poate utiliza S3 Batch Operations pentru a rula funcții AWS Lambda pe obiectele clientului pentru a rula logica aplicațiilor, cum ar fi procesarea datelor sau transcodarea fișierelor imagine. Se specifică o listă de obiecte întâi utilizând un raport de inventar S3 sau furnizând o listă personalizată, apoi se selectează operația dorită dintr-un meniu pre-populat. Când se efectuează o solicitare S3 Batch Operation, va apărea o notificare și un raport de finalizare a tuturor modificărilor efectuate.

Amazon S3 suportă, de asemenea, funcții care ajută la menținerea controlului versiunii datelor, la prevenirea ștergerilor accidentale și la replicarea datelor în aceeași regiune AWS sau într-o altă regiune. Cu S3 Versioning, se poate păstra, prelua și restaura fiecare versiune a unui obiect stocat în Amazon S3, ceea ce permite recuperarea după acțiunile neintenționate ale utilizatorului și eșecurile aplicației. Pentru a preveni ștergerile accidentale, se poate activa opțiunea Multi-Factor Authentication (MFA) Delete pe un bucket S3.

Dacă se încearcă ștergerea unui obiect stocat într-un bucket cu MFA activat pentru ștergere, vor fi necesare două forme de autentificare: credențialele contului AWS și concatenarea unui număr de serie valid, a unui spațiu și a codului din șase cifre afișat pe un dispozitiv de autentificare aprobat, cum ar fi o cheie hardware sau o cheie de securitate Universal 2nd Factor (U2F).

Cu S3 Replication, se pot replica obiecte (și metadatele și etichetele de obiect ale acestora) în unul sau mai multe bucket-uri destinație în aceeași regiuni AWS sau în diferite regiuni pentru o latență redusă, conformitate, securitate, recuperare în caz de dezastru și alte cazuri de utilizare.

Se poate configura S3 Cross-Region Replication (CRR) pentru a replica obiecte dintr-un bucket S3 sursă în unul sau mai multe bucket-uri destinație în diferite regiuni AWS. S3 Same-Region Replication (SRR) replică obiecte între bucket-uri din aceeași regiune AWS. În timp ce replicarea în direct, cum ar fi CRR și SRR, replică automat obiectele nou încărcate pe măsură ce sunt scrise într-un bucket, S3 Batch Replication permite replicarea obiectelor existente.

Se poate utiliza S3 Batch Replication pentru a umple un bucket nou creat cu obiecte existente, pentru a reîncerca obiecte care anterior nu se puteau replica, pentru a migra date între conturi sau pentru a adăuga bucket-uri noi în data lakes [17].

2.6 Amazon API Gateway

Amazon API Gateway este un serviciu complet gestionat care facilitează publicarea, menținerea, monitorizarea, securizarea și operarea API-urilor la orice scară pentru dezvoltatori. Este un serviciu de tip pay-as-you-go care se ocupă de toate sarcinile grele nediferențiate implicate în rularea sigură și fiabilă a API-urilor la scară.

Odată cu înmulțirea dispozitivelor mobile și expansiunea Internetului obiectelor (IoT), este din ce în ce mai obișnuit ca sistemele și datele să fie accesibile aplicațiilor prin intermediul API-urilor. Deoarece atât de multe aplicații folosesc aceste API-uri și comunitățile de dezvoltatori se bazează pe ele, o cantitate tot mai mare de timp și efort se dedică pentru dezvoltarea API-urilor și gestionarea API-urilor. Pentru a facilita utilizarea acestor API-uri, API Gateway poate genera SDK-uri client pentru mai multe limbi de programare, inclusiv JavaScript, iOS și Android [20].



Figura 2.3 Diagramă ce ilustrează modul în care funcționează Amazon API Gateway [19]

Această diagramă ilustrează modul în care API-urile care sunt create în Amazon API Gateway oferă o experiență de dezvoltator integrată și consecventă pentru dezvoltarea de aplicații AWS serverless. API Gateway se ocupă cu acceptarea și procesarea a până la sute de mii de

apeluri API în același timp. Aceste sarcini includ controlul traficului și al accesului, autorizarea, supravegherea și gestionarea versiunilor API.

API Gateway acționează ca o „poartă” pentru aplicații pentru a accesa date, logica din spatele aplicației sau funcționalități din serviciile de backend, cum ar fi sarcinile de lucru care rulează pe Amazon Elastic Compute Cloud (Amazon EC2), codul care rulează pe AWS Lambda, orice aplicație web sau aplicații de comunicare în timp real [21].

Cu API Gateway, se pot crea API-uri RESTful folosind fie API-uri HTTP, fie API-uri REST. API-urile HTTP sunt cea mai bună modalitate de a crea API-uri care nu necesită funcții de gestionare a API-urilor. API-urile HTTP sunt optimizate pentru încărcături de lucru serverless și backend-uri HTTP - oferă până la 71% economisire a costurilor și 60% reducere a latenței în comparație cu API-urile REST de la API Gateway. Pentru sarcinile de lucru care necesită funcționalități proxy API și funcții de gestionare a API-urilor într-o singură soluție, cum ar fi accesul la metodele API și cheile API, API Gateway oferă API-uri REST.

Cu API Gateway, se pot redirecționa solicitările către resurse private din VPC. Folosind API-uri HTTP, se pot crea API-uri pentru servicii în spatele ALB-urilor (Application Load Balancer) private, NLB-urilor (Network Load Balancer) private și serviciilor bazate pe IP înregistrate în AWS Cloud Map, cum ar fi sarcinile ECS.

API Gateway ajută la gestionarea traficului către sistemele backend, permitând setarea de reguli de limitare pe baza numărului de solicitări pe secundă pentru fiecare metodă HTTP din API-urile clientului. API Gateway gestionează orice nivel de trafic primit de un API, astfel încât dezvoltatorii nu trebuie să se concentreze deloc pe mențenanța infrastructurii. Dacă se utilizează API-uri REST, se poate configura o memorie cache cu chei personalizabile și durată de viață în secunde pentru datele API, pentru a evita accesarea serviciilor backend la fiecare solicitare.

Cu API Gateway, se poate crea rapid și ușor un API personalizat pentru codul clientului care rulează în AWS Lambda și apoi să fie apelat codul Lambda din API. API Gateway poate executa codul AWS Lambda în contul clientului, poate porni mașinile de stare AWS Step Functions sau poate efectua apele către AWS Elastic Beanstalk, Amazon EC2 sau servicii web în afara AWS cu endpoint-uri HTTP accesibile public. Folosind consola API Gateway, se poate defini API-ul REST și resursele și metodele asociate acestuia, se poate gestiona ciclul de viață al API-ului, se pot genera SDK-uri pentru client și se poate vizualiza statistici cu privire la API.

După ce un API este implementat și utilizat, API Gateway oferă un tablou de bord pentru a monitoriza vizual apelele către servicii. Consola API Gateway este integrată cu Amazon CloudWatch, astfel încât se pot obține valori de performanță din backend, cum ar fi apelele API, latență și frecvența erorilor. Deoarece API Gateway utilizează CloudWatch pentru a înregistra informații de monitorizare, se poate configura alarme personalizate pentru API-urile din API Gateway. De asemenea, API Gateway poate înregistra erorile de execuție API în CloudWatch Logs pentru a ușura depanarea.

Dacă se utilizează API-uri REST, API Gateway ajută la gestionarea ecosistemului dezvoltatorilor terți care accesează API-urile. Se poate crea chei API pe API Gateway, seta permisiuni de acces detaliate pentru fiecare cheie API și distribui dezvoltatorilor terți aceste chei pentru a accesa API-urile. Utilizarea cheilor API este complet optională și trebuie să fie activată la nivel de metodă.

Pentru API-urile REST, API Gateway poate genera SDK-uri client pentru o serie de platforme pe care se pot utiliza pentru a testa rapid noi API-uri din aplicațiile clientului și pentru a distribui SDK-uri dezvoltatorilor terți. SDK-urile generate gestionează cheile API și semnează solicitările folosind acreditările AWS. API Gateway poate genera SDK-uri client pentru Java, JavaScript, Java pentru Android, Objective-C sau Swift pentru iOS și Ruby.

În cazul API-urilor REST, API Gateway permite rularea a mai multe versiuni ale aceluiași API simultan, astfel încât aplicațiile să poată continua să apeleze versiunile API anterioare chiar și după ce cele mai recente versiuni sunt publicate. API Gateway permite gestionarea a mai multe etape de lansare pentru fiecare versiune API, cum ar fi alfa, beta și producție. Fiecare etapă API poate fi configurată pentru a interacționa cu diferite endpoint-uri din backend în funcție de configurarea API-ului. Etape și versiuni specifice ale unui API pot fi asociate cu un nume de domeniu personalizat și gestionate prin API Gateway. Gestionarea etapei și a versiunilor permite testarea cu ușurință a noilor versiuni API care îmbunătățesc sau adaugă funcționalități noi la versiunile anterioare API și asigură compatibilitatea inversă pe măsură ce utilizatorii tranzitează spre adoptarea celei mai recente versiuni [20].

2.7 AWS CloudFormation

AWS CloudFormation oferă o modalitate simplă de a modela o colecție de resurse asociate cu AWS și terțe părți, de a le furniza rapid și consecvent și de a le gestiona pe parcursul ciclului de viață, tratând infrastructura ca cod. Un șablon CloudFormation descrie resursele dorite și dependențele acestora, astfel încât se pot lansa și configura împreună ca o stivă. Se poate folosi un șablon pentru a crea, actualiza și șterge o stivă întreagă ca o singură unitate, ori de câte ori este nevoie, în loc de gestionarea resurselor individual. Se pot gestiona și furniza stive în mai multe conturi AWS și regiuni AWS.

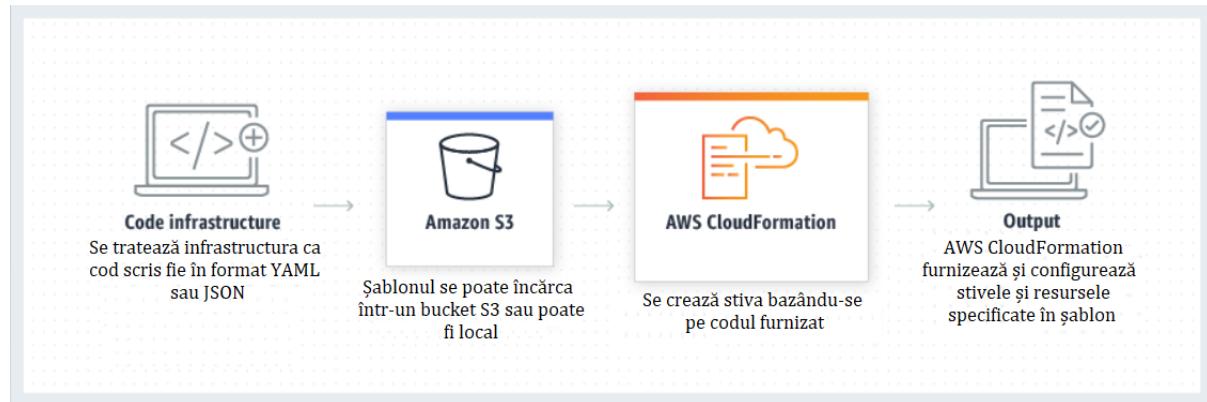


Figura 2.4 Diagramă ce ilustrează crearea unei stive în AWS CloudFormation [22]

Folosind AWS CloudFormation Registry, se poate modela și furniza resurse și module din terțe părți publicate de partenerii AWS Partner Network (APN) și comunitatea de dezvoltatori. Exemple de resurse din terțe părți sunt instrumentele de monitorizare, productivitatea echipei, managementul incidentelor și controlul versiunilor, împreună cu resursele partenerilor APN.

Se pot construi proprii furnizori de resurse utilizând AWS CloudFormation CLI, un instrument open-source care simplifică procesul de dezvoltare, inclusiv testarea locală și capabilitățile de generare de cod.

CloudFormation StackSets permite furnizarea unui set comun de resurse AWS în mai multe conturi și regiuni, cu un singur şablon CloudFormation. StackSets se ocupă de aprovizionarea, actualizarea sau ștergerea automată și sigură a stivelor, indiferent unde se află acestea.

CloudFormation permite modelarea întregului mediu cloud în fișiere text. Se pot utiliza limbaje declarative open-source, cum ar fi JSON sau YAML, pentru a descrie ce resurse AWS se doresc să fie create și configurate. Dacă se preferă o proiectare vizuală, se poate utiliza AWS CloudFormation Designer.

Cu AWS Cloud Development Kit (AWS CDK), se poate defini mediul cloud folosind TypeScript, Python, Java și .NET. AWS CDK este un framework de dezvoltare software open-source care permite modelarea resurselor aplicațiilor cloud utilizând limbaje de programare cunoscute și apoi furnizarea infrastructurii folosind CloudFormation direct din IDE.

Se poate dezvolta rapid aplicații serverless cu AWS Serverless Application Model (SAM), un framework open-source care oferă o sintaxă scurtă pentru a declara funcții, API-uri, baze de date și mapări ale surselor evenimentelor. Cu doar câteva linii de cod pe resursă, se poate defini aplicația și se poate modela folosind YAML. În timpul implementării, SAM transformă și extinde sintaxa SAM în sintaxa CloudFormation.

CloudFormation automatizează furnizarea și actualizarea infrastructurii într-un mod sigur și controlat. Nu există pași sau controale manuale care pot duce la erori. Se poate utiliza declanșatoarele de derulare înapoi a operațiilor efectuate pentru a specifica alarmele CloudWatch pe care CloudFormation ar trebui să le monitorizeze în timpul procesului de creare și actualizare a stivei. Dacă vreuna dintre alarme este declanșată, CloudFormation derulează înapoi întreaga operațiune a stivei la o versiune anterioară.

Folosind ChangeSets, se pot previzualiza modificările propuse pe care CloudFormation intenționează să le facă asupra infrastructurii și resurselor aplicației înainte de execuție, astfel încât implementările să funcționeze așa cum au fost planificate. CloudFormation determină operațiunile potrivite de efectuat, furnizează resurse în cel mai eficient mod posibil și efectuează derularea automată înapoi în cazul în care se întâlnesc erori. Aceasta reduce starea infrastructurii și a resurselor aplicației la ultima stare bună cunoscută. Folosind Drift Detection, se pot urmări modificările aduse resurselor din afara CloudFormation, asigurându-se folosirea întotdeauna a celei mai recente imagini a infrastructurii clientului.

AWS CloudFormation gestionează automat dependențele dintre resursele clientului în timpul acțiunilor de gestionare a stivei. Nu este nevoie de specificarea ordinii în care resursele sunt create, actualizate sau șterse; CloudFormation determină secvența corectă de acțiuni de întreprins pentru fiecare resursă atunci când se efectuează operațiuni pe stivă [23].

2.8 Xamarin

Xamarin este o platformă open-source destinată dezvoltării aplicațiilor noi, actuale și performante pentru dispozitivele mobile și Windows cu .NET. Xamarin reprezintă o abstractizare ce asigură comunicarea între diferite platforme și codul de bază partajat.

Cu Xamarin, dezvoltatorii de aplicații mobile pot partaja aproximativ 90% din codul de bază pe platformele sănătoase. Astfel, toată logica din spatele aplicației poate fi scrisă într-un singur

limbaj de programare, reușind să obțină o performanță asemănătoare cu o aplicație nativă pe fiecare dintre platforme.

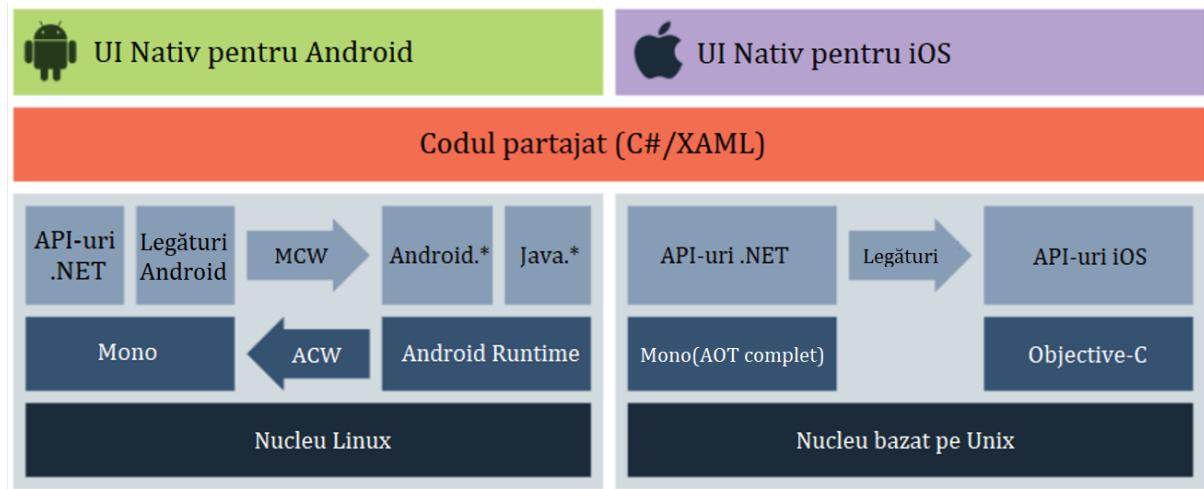


Figura 2.5 Arhitectura aplicățiilor Xamarin

Diagrama prezintă arhitectura generală a unei aplicații Xamarin multiplatformă. Xamarin permite crearea unei interfețe de utilizare nativă pe fiecare platformă și scrierea logicii din spatele aplicației în C# care este partajată pe platforme. În majoritatea cazurilor, 80% din codul aplicației poate fi partajat folosind Xamarin.

Xamarin este dezvoltat pe .NET, care se ocupă automat de sarcini precum alocarea memoriei, eliberarea memoriei și interoperabilitatea cu platformele subiacente.

Xamarin combină abilitățile platformelor native, adăugând în același timp funcții care includ: legături complete pentru SDK-urile de bază, folosirea directă a bibliotecilor Objective-C, Java, C și C++, cât și terțe părți, îmbunătățiri datorate scrierii codului în C#, suport puternic XML, un IDE modern, suport multiplatformă și.a.

Aplicațiile Xamarin.Android sunt compilate în limbajul intermediar (IL), urmând să fie compilat Just-in-Time (JIT) într-un ansamblu nativ la lansarea aplicației. Aplicațiile se rulează în Mono, alături de Android Runtime (ART). Mono face referire la spațiile de nume Android.* și Java.* prin Managed Callable Wrappers (MCW) și furnizează Android Callable Wrappers (ACW) către ART, fiind astfel permis ca cele două medii de execuție să apeleze cod dintr-un mediu în altul.

Aplicațiile Xamarin.iOS sunt complet Ahead-of-Time (AOT) compilate din C# în codul de asamblare ARM nativ. Xamarin folosește selectoare pentru a expune Objective-C la C# gestionat și registratori pentru a expune codul C# gestionat la Objective-C. Selectoare și registratori în mod colectiv sunt numiți legături și permit ca Objective-C și C# să comunice.

Xamarin.Essentials este o bibliotecă care oferă servicii multiplatformă pentru funcțiile native ale dispozitivului. Ca și Xamarin însuși, Xamarin.Essentials simplifică procesul de accesare a funcționalității native. Câteva exemple de funcționalități oferite de Xamarin.Essentials includ [24]:

- Informații despre dispozitiv
- Sistemul de fișiere
- Accelerometru
- Apelare telefonică

- Text-to-speech
- Blocarea ecranului

2.9 Xamarin.Forms

Xamarin.Forms este un framework pentru interfețe utilizator open-source. Xamarin.Forms permite crearea de aplicații Xamarin.Android, Xamarin.iOS și Windows folosind același cod partajat.

Xamarin.Forms permite dezvoltatorilor crearea de interfețe utilizator în XAML (Extensible Application Markup Language) scrise în C#. Interfețele sunt redate pe platforme ca și cum ar fi controale native.

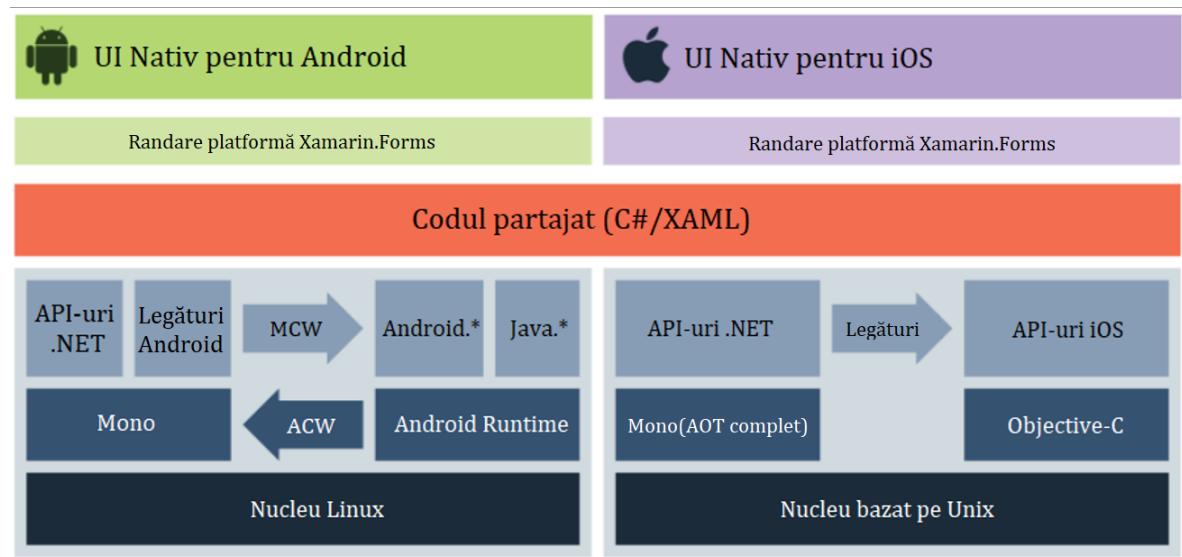


Figura 2.6 Arhitectura aplicațiilor Xamarin.Forms

În timpul execuției, Xamarin.Forms folosește randarea platformelor pentru a afișa elementele interfeței de utilizare multiplatformă în controale native pe dispozitivele Android, iOS și UWP.

Aplicațiile Xamarin.Forms sunt constituite dintr-o bibliotecă .NET Standard partajată și proiecte specifice unei platforme individuale. Biblioteca partajată conține vizualizările XAML sau C# și orice logică din spatele aplicației. Proiectele platformei conțin orice logică sau pachete specifice platformei pe care le solicită aplicația.

Xamarin.Forms folosește platforma Xamarin pentru a rula aplicații .NET nativ pe platforme [25].

2.10 ASP.NET Core

ASP.NET Core este un framework multiplatformă, cu o performanță foarte bună, open-source destinat dezvoltării aplicațiilor de actualitate, în cloud, ce comunică pe Internet.

Cu ASP.NET Core, se pot:

- Crea aplicații și servicii web, aplicații pentru Internetul obiectelor (IoT) și backend-uri pentru aplicațiile mobile.
- Utilizează instrumentele de dezvoltare cunoscute pe Windows, macOS și Linux.
- Implementă aplicații ce pot fi găzduite în cloud sau local.
- Rulează aplicațiile pe .NET Core.

ASP.NET Core s-a bazat pe ASP.NET 4.x, reproiectat la nivel de arhitectură rezultând astfel un framework mai modular și simplu.

ASP.NET Core oferă următoarele beneficii:

- poveste unificată pentru construirea de interfețe utilizator web și a API-urilor web.
- proiectare pentru testabilitate.
- paginile Razor fac codificarea scenariilor axate pe pagină mai ușor de dezvoltat și cu o productivitate crescută.
- cu Blazor se poate utiliza C# și JavaScript împreună într-un browser, logica aplicației fiind partajată pe partea clientului și a serverului, toate scrise în C#.
- abilitatea de a dezvolta și rula pe mai multe platforme precum Linux, Windows și macOS.
- open-source și axat pe comunitate.
- integrarea framework-urilor pentru clienți moderne și a procesului de dezvoltare.
- suport pentru găzduirea serviciilor Remote Procedure Call (RPC) folosind gRPC.
- un sistem de configurare bazat pe mediu pregătit pentru cloud.
- injecții de dependențe încorporate.
- un pipeline de solicitări HTTP ușor, de înaltă performanță și modular.
- versiuni multiple.

ASP.NET Core se integrează perfect cu framework-uri și biblioteci populare pe partea clientului, inclusiv Blazor, Angular, React și Bootstrap.

ASP.NET Core 3.x sau o versiune ulterioară poate viza numai .NET Core.

Unele dintre avantajele folosirii .NET Core în comparație cu .NET Framework sunt [26]:

- multiplatformă
- performanță mai bună
- versiuni multiple
- API-uri noi
- open-source

Capitolul 3. Proiectare și implementare

3.1 Motivația

Am ales această temă gândindu-mă ce aplicație aş putea dezvolta în folosul facultății, ce ar putea fi folosită atât de studenți cât și de profesori. Evaluarea profesorilor ce trebuie să o facem la fiecare început de semestrul m-a inspirat să creez o aplicație ce se focusează tot pe actul didactic, dar nu pe conduită profesorilor și relația student-profesor, ci pe materialele didactice utilizate de cadrele didactice în timpul orelor de curs, seminar, laborator sau proiect.

În contextul pandemiei în special, am observat din experiența personală că la unele discipline am interacționat mai mult cu materialele didactice puse la dispoziție de profesori, interacțiunea cu cadrul didactic fiind câteodată foarte scurtă sau doar la finalul orei fără a primi prea multe indicații de la aceștia, fiind încurajați să citim lucrarea de laborator sau suportul de curs și să ne descurcăm singuri. De asemenea, dacă un student a lipsit de la un curs, singurul mod în care ar putea recupera acele ore și a putea dobândi cunoștințele împărtășite în acel interval este prin citirea suportului de curs, excluzând cererea notițelor de curs de la alții colegi sau discuția cu profesorul cu privire la subiectele discutate la curs deoarece cadrul didactic nu ar putea relua cele spuse pentru fiecare student care nu a participat la curs, în cele din urmă studentul care a absentat va fi îndrumat să citească suportul de curs și materialele auxiliare.

Din aceste scenarii se poate observa importanța resurselor educaționale în actul didactic, corectitudinea conținutului și accesibilitatea fiind doar câteva dintre criteriile importante ce ar trebui să fie îndeplinite în mod implicit pentru a asigura o educație bună.

După ce am stabilit tema pe care mi-am propus să o abordez, m-am gândit la modul în care ar putea să implementez un sistem de evaluare al actului didactic și am decis să dezvolt o aplicație mobilă. În urma căutărilor am considerat că o aplicație Xamarin.Forms îndeplinește nevoile sistemului de evaluare pe care mi-am propus să-l dezvolt, în special pentru că pot scrie același cod pentru dispozitive Android și iOS.

Aplicația ar trebui să funcționeze în felul următor: studentul se conectează cu un cont deja creat, asemănător cu contul de Moodle, și se afișează o pagină cu chestionarul ce îl are de completat și trimité chestionarul. Cadrul didactic se conectează tot cu un cont deja creat și i se va afișa o pagină de unde va putea selecta la ce disciplină ar dori să vadă un raport al evaluărilor transmise de studenți. Un administrator se conectează cu un cont deja creat și poate înregistra sau șterge utilizatori.

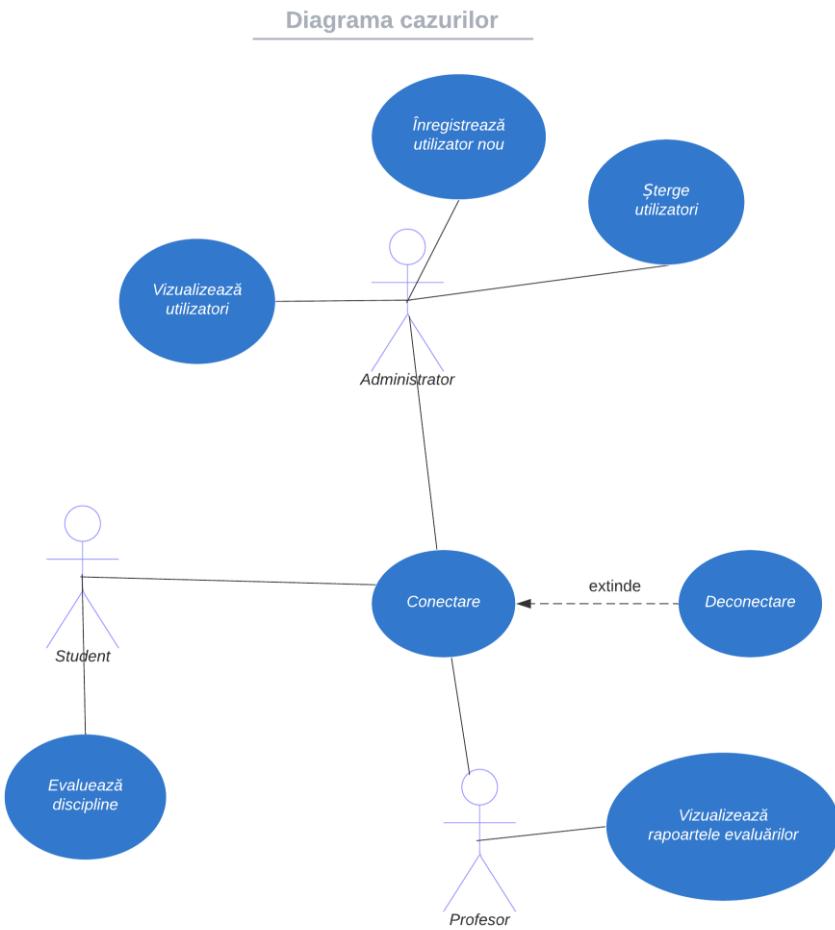


Figura 3.1 Diagrama cazurilor generală

3.2 Chestionarul

Chestionarul va arăta în felul următor: studentul va specifica ciclul de studii, anul de studiu, programul de studiu, disciplina, tipul activității, activitatea didactică sau activitățile didactice pe care dorește să le evalueze, activitățile didactice la care a fost prezent, întrebările și răspunsul la acestea.

Întrebările și răspunsurile ce pot fi date sunt următoarele:

1. Materialul didactic se regăsește în format: "Fizic", "Electronic", "Fizic, Electronic", "Nu se aplică".
2. Materialul didactic este gratuit: "Da", "Nu", "Nu se aplică".
3. Materialul didactic este accesibil(poate fi accesat ușor de pe un dispozitiv mobil/PC sau nu necesită mulți pași pentru a fi accesat): o scală de la 1 la 5 și opțiunea de „Nu se aplică” ce va reprezenta valoarea 6, valoarea 1 semnificând faptul că studentul nu este deloc de acord cu afirmația iar 5 semnifică faptul că studentul este de acord cu afirmația prezentată.
4. Materialul didactic oferă destule informații despre subiectul prezentat: idem.
5. Materialul didactic este organizat și ușor de navigat: idem.

6. Materialul didactic oferă destule explicații: idem.
7. Materialul didactic oferă o sursă de evaluare a performanțe melei: idem.
8. Materialul didactic de la această activitate este în concordanță cu cel de la alte tipuri de activități din cadrul acestei discipline: idem.
9. Gradul de dificultate al acestei activități a fost: idem.
10. Volumul de informații prezentat este adekvat: idem.
11. Experiența dobândită la această activitate mă ajută să mă dezvolt pe plan profesional: idem cu excepția faptului că opțiunea „Nu se aplică” nu există.
12. Materialul didactic nu conține multe greșeli gramaticale, este lizibil și nu conține informații eronate: idem.
13. Materialul didactic este disponibil în următoarele limbi: ”Română”, ”Engleză”, ”Nu se aplică”, ”Română, Engleză”, alte limbi, ”Română, *alte limbi*”, ”Engleză, *alte limbi*”, ”Română, Engleză, *alte limbi*”.
14. Ce îmbunătățiri ați propune pentru această activitate?: răspuns liber.
15. Ce aspecte din cadrul acestei activități v-au plăcut?: răspuns liber.

Evaluarea activității didactice se va lua în considerare doar dacă studentul a menționat că a fost prezent la acea activitate iar evaluarea se face doar o singură dată, fără posibilitatea de a modifica răspunsurile. Este obligatorie selectarea ciclului de studii, anului de studiu, programului de studiu, disciplina, tipul activității, activitățile pe care dorește să le evalueze, oferirea de răspunsuri la toate întrebările cu excepția întrebărilor cu răspuns liber. Nu este obligatorie specificarea activităților la care a fost prezent însă răspunsurile nu vor fi luate în considerare. Dacă nu a completat toate câmpurile obligatorii acesta va fi atenționat iar răspunsurile nu vor fi trimise până nu vor fi completate. Dacă a evaluat deja o anumită activitate didactică acesta va fi atenționat că a evaluat deja acea activitate. Dacă dorește să evalueze o activitate didactică la care nu a fost prezent studentul va fi atenționat că evaluarea nu a fost trimisă.

Pentru întrebările cu răspuns liber va exista câte un câmp pentru fiecare activitate pe care studentul dorește să o evalueze. Pentru selectarea activităților didactice pe care ar dori să le evalueze studentul va exista o listă cu numere de la 1 până la numărul total de activități didactice la tipul de activitate didactică ales de student (curs/laborator/seminar/proiect) la disciplina selectată. După selectarea activităților, studentul va putea selecta la care dintre acestea a fost prezent. Se recomandă ca studentul să fie sincer și să evalueze doar activitățile didactice la care a fost prezent, altfel răspunsurile nu vor fi salvate. În cazul în care a vrut să evalueze o activitate didactică la care nu a menționat că a fost prezent și trimite răspunsurile, acestea nu se vor salva, însă dacă evaluează iar acea activitate didactică și menționează că a fost prezent, răspunsurile vor fi trimise cu succes.

În funcție de numărul activității, adică dacă activitatea didactică se desfășoară în timpul semestrului sau la final de semestru, se vor afișa toate întrebările în cazul din urmă și doar întrebările 4, 5, 6, 7, 9, 10, 14 și 15 pentru primul caz.

3.3 Interfețele aplicației

Interfața paginii de administare va arăta în modul următor: va exista o secțiune de înregistrare a unui utilizator nou și unde va fi obligatorie specificarea numelui de utilizator, email-ul, parola și rolul (student sau profesor). Dacă nu au fost completate toate câmpurile sau doar unele dintre

acestea administratorul va fi atenționat și i se vor da indicații cum ar trebui să le completeze. Numele de utilizator se va considera valid dacă respectă formatul <nume>.<nume><număr>, doar litere mici și fără diacritice. Email-ul va fi valid doar dacă respectă modelul unui email valid tipic. Parola va fi considerată validă dacă începe cu literă mare, are cel puțin 6 caractere, conține cel puțin o cifră iar caracterele sunt doar litere sau cifre. Rolul va fi specificat prin alegerea dintr-o listă cu două elemente, student și profesor. Nu se vor putea crea alte conturi de administrator și nici nu se poate șterge unicul cont de administrator. Ștergerea utilizatorilor se va face prin selectarea dintr-o listă a numelui de utilizator a studentului sau profesorului ce se dorește a fi șters, existând posibilitatea de a selecta mai mulți utilizatori. Lista utilizatorilor se va reîmprospăta prin apăsarea unui buton.

Interfața ce va fi vizualizată de profesori va arăta în modul următor: cadrul didactic va trebui să specifice ciclul de studii, programul de studiu, disciplina și tipul activității pentru a putea vedea raportul evaluărilor făcute de studenți la o disciplină. Aceștia vor putea vedea raportul evaluărilor la orice disciplină din cadrul facultății. Se vor afișa mai întâi răspunsurile la întrebările cu răspuns liber iar apoi câte o diagramă cu linii pentru fiecare întrebare unde evaluarea se face printr-o scală numerică. Astfel se vor putea analiza tendințele studenților.

Pagina de conectare va reprezenta două câmpuri pentru introducerea adresei de email și parolei utilizatorului și un buton pentru conectare.

3.4 Aplicația Xamarin.Forms

Am creat o aplicație Xamarin.Forms cu o pagină goală. Am setat nivelul minim al API-ului pe care va putea rula aplicația la nivelul 21 (Lollipop – Android 5.0) și nivelul țintă la nivelul 33 (Android 13.0). Am setat permisiunile necesare ale aplicației pentru a putea rula conform, specificând necesitatea de a putea folosi internetul (INTERNET) și starea rețelei (ACCESS_NETWORK_STATE). Am instalat pachetele NuGet Microcharts.Forms (0.9.5.9), Microsoft.NET.Build.Containers (7.0.302), NETStandardLibrary (2.0.3), Newtonsoft.Json (13.0.3), SkiaSharp (2.88.3), Xamarin.Essentials (1.7.5) și Xamarin.Forms (5.0.0.2578). Aceste pachete vor fi necesare pentru afișarea diagramelor, serializarea și deserializarea JSON și pentru alte funcționalități de bază din cadrul proiectului.

Am creat paginile ce vor fi afișate utilizatorului, adică patru pagini de conținut ce se află în folderul "Pagini". Numele acestor pagini sunt "Administrare", "Conectare", "EvaluareDiscipline" și "RaportEvaluari". Sunt fișiere XAML ce au în spatele lor cod C#.

3.5 Interfață administrator

Pagina de administrare "Administrare" este compusă dintr-o grilă de o coloană și o linie ce are în interior un ScrollView, care la rândul său are un StackLayout unde vor fi plasate elementele necesare. Acest StackLayout este compus dintr-o etichetă cu textul "Înregistrare utilizator", trei câmpuri text pentru introducerea numelui utilizatorului, email-ul și parola, un selector Picker de unde se va selecta rolul utilizatorului, un buton pentru înregistrare utilizator, o etichetă cu textul "Ștergere utilizatori", un ScrollView ce are în interior un CollectionView ce va afișa lista utilizatorilor din baza de date, prin legarea datelor într-o etichetă, fiind selectat

modul selectare multiplă și un buton pentru ștergea utilizatorilor. De asemenea, fiecare pagină de conținut va avea un element în bara de instrumente pentru deconectare și redirecționarea utilizatorului către pagina de conectare.

```
0 references
private async void Deconectare(object sender, EventArgs e)
{
    await Navigation.PopAsync();
    await Navigation.PushAsync(new Conectare());
}
```

Figura 3.2 Funcția de deconectare

Înregistrarea se face prin preluarea datelor din câmpurile text și selector, validarea acestora și printr-o solicitare API se trimit datele și se salvează în baza de date dacă sunt valide. Dacă nu sunt valide, se va afișa o alertă ce va informa administratorul cum trebuie să arate un nume de utilizator, email și parolă valide. Se va afișa o alertă și dacă utilizatorul a fost înregistrat cu succes sau dacă nu a fost înregistrat cu succes.

Ștergerea utilizatorilor se realizează prin preluarea numelor de utilizator selectate din CollectionView și solicitări API repetitive cu numele selectate. Se va afișa o alertă dacă utilizatorul a fost șters cu succes sau dacă nu a fost șters, în funcție de codul HTTP primit ca răspuns.

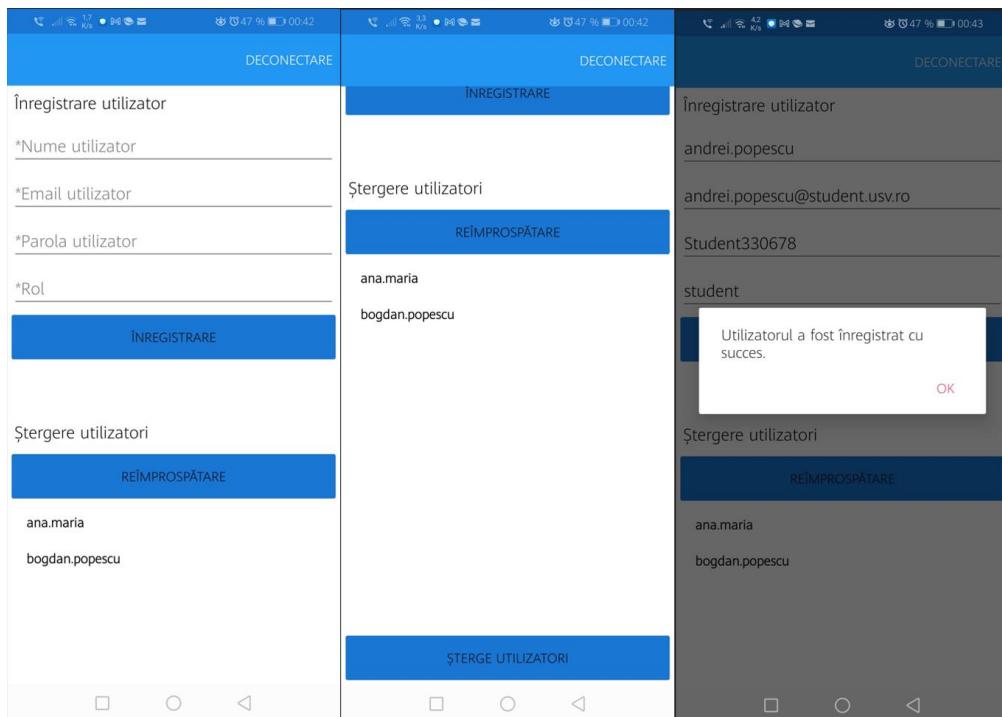


Figura 3.3 Interfață pagina de administrare (1)

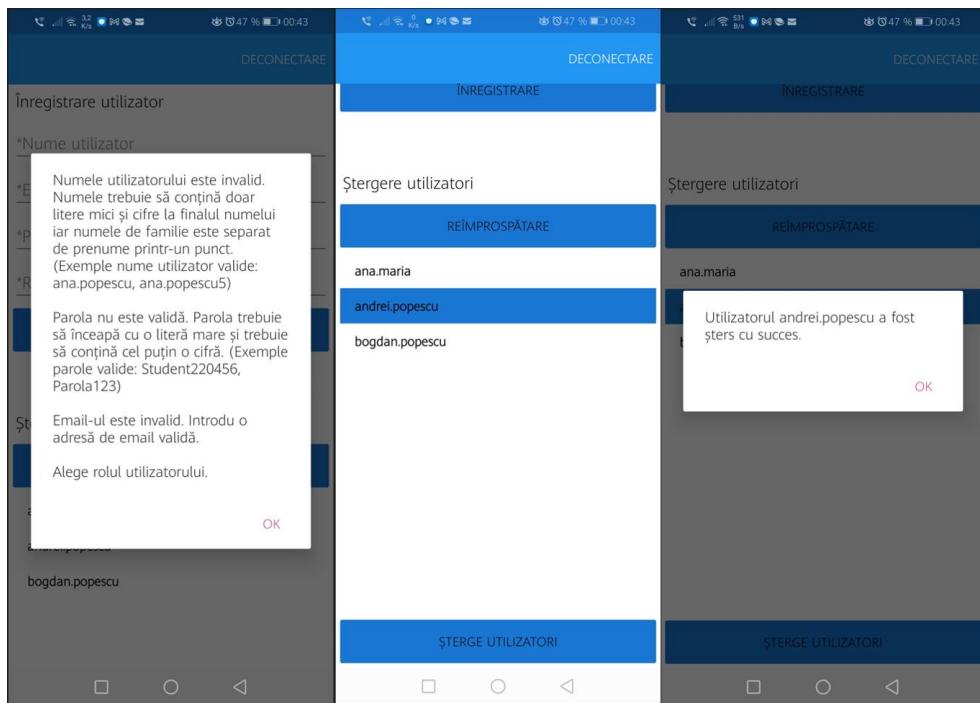


Figura 3.4 Interfață pagina de administrare (2)

3.6 Interfață conectare utilizator

Pagina de conectare este compusă din două câmpuri text pentru introducerea adresei de email și parolei, un buton pentru conectare și o imagine cu sigla universității.



Figura 3.5 Interfață pagina de conectare

Conectarea se realizează prin preluarea valorilor din aceste două câmpuri și trimiterea unui solicitări către API. Dacă credențialele introduse sunt corecte, răspunsul la solicitare va consta

dintron token de acces JWT și un token de reîmprospătare, identificatorul utilizatorului, rolul, data expirării celor două token-uri și numele utilizatorului. Token-ul de acces, de reîmprospătare și identificatorul vor fi salvate cu ajutorul API-ului Xamarin.Essentials.SecureStorage, ce va stoca valorile în Keychain (iOS) și KeyStore (Android). Aceste valori vor fi folosite de fiecare dată când se va face o solicitare API ce necesită un token de acces pentru autorizare. În funcție de rolul returnat, utilizatorul va fi redirecționat către una din celealte trei pagini: "Administrare" (administrator), "EvaluareDiscipline" (student), "RaportEvaluari" (profesor). Dacă credențialele nu sunt corecte, se va afișa o alertă pentru a atenționa utilizatorul de acest lucru. Dacă utilizatorul cu credențialele introduse nu există, API-ul va returna ca răspuns un obiect de tip Autentificare ce va avea proprietățile cu valoari nule.

```

var email = campEmail.Text;
var parola = campParola.Text;

Logare logare = new Logare { email = email, parola = parola };

var autentificare = await administrareServicii.Logare(logare);

if (autentificare.id_utilizator == null)
{
    await DisplayAlert("", "Parola sau numele de utilizator nu sunt corecte.", "Ok");
}
else
{
    await Xamarin.Essentials.SecureStorage.SetAsync("token", autentificare.token);
    await Xamarin.Essentials.SecureStorage.SetAsync("tokenReimprospatare", autentificare.token_reimprospatare);
    await Xamarin.Essentials.SecureStorage.SetAsync("idUtilizator", autentificare.id_utilizator);

    if (autentificare.rol == "student")
    {
        await Navigation.PopAsync();
        await Navigation.PushAsync(new EvaluareDiscipline());
    }

    if (autentificare.rol == "profesor")
    {
        await Navigation.PopAsync();
        await Navigation.PushAsync(new RaportEvaluari());
    }

    if (autentificare.rol == "administrator")
    {
        await Navigation.PopAsync();
        await Navigation.PushAsync(new Administrare());
    }
}

```

Figura 3.6 Funcția de conectare a utilizatorului

3.7 Interfața student

Pagina ce va conține chestionarul este formată din: o grilă cu două linii și o coloană, pe prima linie fiind chestionarul iar pe a doua linie butonul de trimitere a răspunsurilor, un ScrollView pe prima linie ce conține selectorii pentru ciclul de studii, anul de studiu, programul de studiu, disciplina, tipul activității, un ScrollView pentru lista activităților, un alt ScrollView ce va fi populat cu activitățile pe care studentul dorește să le evaluateze de unde va putea selecta activitățile didactice la care a fost prezent, etichete pentru fiecare întrebare din chestionar, câmpuri text pentru întrebările cu răspuns liber ce vor fi afișate dinamic și grile cu mai multe coloane pentru afișarea butoanelor radio sau căsuțelor de bifat (unde e cazul).

Implicit se vor afișa doar întrebările 4, 5, 6, 7, 9, 10, 14 și 15, iar dacă se selectează ultima activitate din semestru vor fi afișate toate întrebările.

Ciclurile de studii și tipurile de activități didactice vor fi preluate la inițializare printr-o solicitare API autorizată. Anii de studiu vor fi preluati printr-o solicitare API autorizată în funcție de ciclul de studii selectat. Programele de studii vor fi preluate printr-o solicitare API autorizată în funcție de ciclul de studii selectat. Disciplinele vor fi preluate printr-o solicitare API autorizată în funcție de programul de studiu selectat. Tipurile de activitate didactică vor fi preluate tot printr-o solicitare API autorizată și afișate în selector. La selectarea tipului activității didactice, se va popula lista activităților de unde va alege studentul pe cele care dorește să le evaluateze cu o listă de numere de la 1 până la numărul total de activități ce există la disciplina și tipul de activitate selectate. Când se selectează activitățile ce se doresc să fie evaluate, se populează și lista pentru selectarea activităților la care a fost prezent studentul cu aceste activități (numărul activităților, adică tot o listă de numere). De asemenea, se adaugă și câmpuri text la întrebările cu răspuns liber pentru fiecare activitate selectată.

Pentru întrebarea 1 vor fi trei căsuțe de bifat cu etichetele "Fizic", "Electronic", "Nu se aplică". Dacă se bifează "Fizic" iar apoi se bifează "Nu se aplică", căsuța cu eticheta "Fizic" va fi debifată și căsuța "Nu se aplică" bifată. Dacă se bifează ambele căsuțe "Fizic" și "Electronic", ambele vor rămâne bifate. Dacă se bifează mai întâi "Nu se aplică" și apoi se bifează "Fizic" sau "Electronic", căsuța "Nu se aplică" va fi debifată iar căsuța "Fizic" sau "Electronic" va fi bifată.

Pentru întrebarea 2 vor fi trei butoane radio "Da", "Nu" și "Nu se aplică". Doar unul din butoane poate fi selectat și nu poate fi deselectat.

Pentru întrebările 3, 4, 5, 6, 7, 8, 9, 10 și 12 vor exista șase butoane radio așezate pe orizontală, primul buton având eticheta „Nu sunt de acord”, al cincelea „Sunt de acord” și al șaselea „Nu se aplică”. Excepție face întrebarea 9 unde etichetele sunt „Foarte scăzut”, „Foarte ridicat” și „Nu se aplică”. Doar unul din butoane poate fi selectat și nu poate fi deselectat.

Pentru întrebarea 11 vor exista cinci butoane radio așezate pe orizontală, primul buton având eticheta „Nu sunt de acord” și al cincelea „Sunt de acord”. Doar unul din butoane poate fi selectat și nu poate fi deselectat.

Pentru întrebarea 13 vor fi trei căsuțe de bifat cu etichetele "Română", "Engleză" și "Nu se aplică" și un câmp text pentru a introduce alte limbi. Dacă se selectează "Română" iar apoi se selectează "Nu se aplică", căsuța "Română" va fi debifată iar căsuța "Nu se aplică" bifată. De asemenea, dacă s-a introdus o altă limbă și apoi se selectează căsuța "Nu se aplică", textul introdus se va șterge.

Pentru întrebările 14 și 15 se vor afișa câmpuri text în mod dinamic în funcție de activitățile selectate.

DECONECTARE	DECONECTARE	DECONECTARE
Câmpurile cu * sunt obligatorii	11 12 13 14	<input type="checkbox"/> Fizic <input checked="" type="checkbox"/> Electronic <input type="checkbox"/> Nu se aplică
Licență	• *Am fost prezent(ă) la următoarele activități: 1 2	• *Materialul didactic este gratuit: <input checked="" type="radio"/> Da <input type="radio"/> Nu <input type="checkbox"/> Nu se aplică
Calculatoare	14	• *Materialul didactic este accesibil(poate fi accesat ușor de pe un dispozitiv mobil/PC sau nu necesită mulți pași pentru a fi accesat): <input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică
Proiectare logică	12	• *Materialul didactic se regăsește în format: <input type="checkbox"/> Fizic <input checked="" type="checkbox"/> Electronic <input type="checkbox"/> Nu se aplică
Curs	11 12 13 14	• *Materialul didactic oferă destule informații despre subiectul prezentat: <input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică
● *Selectează activitatea/activitățile pe care dorești să le evaluezi:		
	TRIMITE	TRIMITE
	<input type="checkbox"/> <input type="radio"/> <input type="radio"/>	<input type="checkbox"/> <input type="radio"/> <input type="radio"/>

Figura 3.7 Interfață chestionar (1)

DECONECTARE	DECONECTARE	DECONECTARE
de pe un dispozitiv mobil/PC sau nu necesită mulți pași pentru a fi accesat): <input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică	<input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică	<input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică
• *Materialul didactic oferă destule informații despre subiectul prezentat: <input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică	• *Materialul didactic oferă destule explicații: <input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică	• *Materialul didactic de la această activitate este în concordanță cu cel de la alte tipuri de activități din cadrul acestei discipline: <input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică
• *Materialul didactic este organizat și usor de navigat: <input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică	• *Materialul didactic oferă o sursă de evaluare a performanțe mele: <input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică	• *Gradul de dificultate al acestei activități a fost: <input type="radio"/> Foarte scăzut <input checked="" type="radio"/> Foarte ridicat <input type="checkbox"/> Nu se aplică
• *Materialul didactic oferă destule	• *Materialul didactic de la această activitate este în concordanță cu cel de la alte tipuri de activități din cadrul acestei discipline: <input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică	• *Volumul de informații prezentat este adevarat: <input type="radio"/> Nu sunt de acord <input checked="" type="radio"/> Sunt de acord <input type="checkbox"/> Nu se aplică
TRIMITE	TRIMITE	TRIMITE
<input type="checkbox"/> <input type="radio"/> <input type="radio"/>	<input type="checkbox"/> <input type="radio"/> <input type="radio"/>	<input type="checkbox"/> <input type="radio"/> <input type="radio"/>

Figura 3.8 Interfață chestionar (2)

The figure consists of three side-by-side screenshots of a mobile application interface for a survey. Each screenshot shows a different type of question:

- Question 1 (Left):** A five-point Likert scale from "Foarte scăzut" (Very low) to "Nu se aplică" (Not applicable). Below it is a note: "• *Volumul de informații prezentat este adekvat:" followed by a five-point Likert scale from "Nu sunt de acord" (I do not agree) to "Sunt de acord" (I agree).
- Question 2 (Middle):** A five-point Likert scale from "Nu sunt de acord" (I do not agree) to "Sunt de acord" (I agree). Below it is a note: "• *Materialul didactic nu conține multe greșeli gramaticale, este lizibil și nu conține informații eronate:" followed by a five-point Likert scale from "Nu sunt de acord" (I do not agree) to "Sunt de acord" (I agree).
- Question 3 (Right):** A multiple-choice question with checkboxes for "Romana" (checked), "Engleza" (unchecked), "Nu se aplică" (unchecked), and "Alte limbi" (unchecked). Below it is a note: "• Ce îmbunătățiri ati propune pentru această activitate?" followed by a text input field containing "Nimic".

Each screenshot also includes a "TRIMITE" (Send) button at the bottom.

Figura 3.9 Interfață chestionar (3)

This screenshot shows a mobile survey interface with a modal dialog box overlaid on the main content. The dialog contains the following text:

*Doar evaluările la care ați fost prezent vor fi luate în considerare.
Evaluarea activității cu numărul 2 nu a fost trimisă.
Evaluarea activității cu numărul 14 a fost trimisă cu succes.
Evaluarea activității cu numărul 12 a fost trimisă cu succes.

At the bottom right of the dialog is an "OK" button.

The background of the survey interface shows the following elements:

- Top Bar:** DECONECTARE button, battery level (47%), and time (00:50).
- Section 1:** A list of languages with checkboxes: Romana (checked), Engleza (unchecked), Nu se aplică (unchecked), and Alte limbi (unchecked). Below this is a note: "• *Gradul de dificultate al acestei activitați a fost:" followed by a five-point Likert scale from "Foarte scăzut" (Very low) to "Nu se aplică" (Not applicable).
- Section 2:** A note: "• Ce îmbunătățiri ati propune pentru această activitate?" followed by a text input field containing "Lorem ipsum lorem ipsum".
- Bottom:** "TRIMITE" button and navigation icons.

Figura 3.10 Interfață chestionar (4)

Trimiterea răspunsurilor se face în felul următor: se validează faptul că la toate secțiunile obligatorii s-au introdus date și se va face o solicitare API autorizată ce va apela o procedură stocată din baza de date care va introduce răspunsurile introduse de student. Doar răspunsurile pentru activitățile la care a fost prezent studentul se vor salva în baza de date. Se va afișa o alertă în care va fi atenționat dacă răspunsurile sale au fost salvate sau nu sau dacă a evaluat deja acea activitate didactică. În baza de date sunt două proceduri stocate, unul pentru cazul când se evaluează o activitate din timpul semestrului și alta pentru finalul semestrului.

```

if (lista_activitati[i] == final)
{
    ChestionarCompletn DTO chestionarCompletn DTO = new ChestionarCompletn DTO();

    chestionarCompletn DTO.id_student = await Xamarin.Essentials.SecureStorage.GetAsync("idUtilizator");
    chestionarCompletn DTO.id_disciplina = disciplina.id_disciplina;
    chestionarCompletn DTO.id_an_de_studiu = anul_de_studiu.id_an_de_studiu;
    chestionarCompletn DTO.id_program_de_studiu = programul_de_studiu.id_programe_de_studiu;
    chestionarCompletn DTO.id_tip_activitate = tip_activitate.id_tip_activitate;
    chestionarCompletn DTO.numar_activitate = lista_activitati[i];
    chestionarCompletn DTO.text_raspuns1 = lista_prezente.Contains(lista_prezente_total[i]) ? "Da" : "Nu";
    chestionarCompletn DTO.text_raspuns2 = raspuns_intrebare2;
    chestionarCompletn DTO.text_raspuns3 = raspuns_intrebare3;
    chestionarCompletn DTO.text_raspuns4 = raspuns_intrebare4.ToString();
    chestionarCompletn DTO.text_raspuns5 = raspuns_intrebare5.ToString();
    chestionarCompletn DTO.text_raspuns6 = raspuns_intrebare6.ToString();
    chestionarCompletn DTO.text_raspuns7 = raspuns_intrebare7.ToString();
    chestionarCompletn DTO.text_raspuns8 = raspuns_intrebare8.ToString();
    chestionarCompletn DTO.text_raspuns9 = raspuns_intrebare9.ToString();
    chestionarCompletn DTO.text_raspuns10 = raspuns_intrebare10.ToString();
    chestionarCompletn DTO.text_raspuns11 = raspuns_intrebare11.ToString();
    chestionarCompletn DTO.text_raspuns12 = raspuns_intrebare12.ToString();
    chestionarCompletn DTO.text_raspuns13 = raspuns_intrebare13.ToString();
    chestionarCompletn DTO.text_raspuns14 = raspuns_intrebare14;
    chestionarCompletn DTO.text_raspuns15 = raspunsuri_intrebarea15[i];
    chestionarCompletn DTO.text_raspuns16 = raspunsuri_intrebarea16[i];

    if (chestionarCompletn DTO.text_raspuns1 == "Da")
    {
        statusCod[i] = await evaluareDisciplineServicii.trimitChestionarCompletn(chestionarCompletn DTO,
            await Xamarin.Essentials.SecureStorage.GetAsync("token"), await Xamarin.Essentials.SecureStorage.GetAsync("tokenReimprospatare"));
    }
}
else

```

Figura 3.11 Trimitere răspunsuri în urma evaluării unei activități didactice de la finalul semestru

```

else
{
    ChestionarPartialDTO chestionarPartialDTO = new ChestionarPartialDTO();

    chestionarPartialDTO.id_student = await Xamarin.Essentials.SecureStorage.GetAsync("idUtilizator");
    chestionarPartialDTO.id_disciplina = disciplina.id_disciplina;
    chestionarPartialDTO.id_an_de_studiu = anul_de_studiu.id_an_de_studiu;
    chestionarPartialDTO.id_program_de_studiu = programul_de_studiu.id_programe_de_studiu;
    chestionarPartialDTO.id_tip_activitate = tip_activitate.id_tip_activitate;
    chestionarPartialDTO.numar_activitate = lista_activitati[i];
    chestionarPartialDTO.text_raspuns1 = lista_prezente.Contains(lista_prezente_total[i]) ? "Da" : "Nu";
    chestionarPartialDTO.text_raspuns5 = raspuns_intrebare5.ToString();
    chestionarPartialDTO.text_raspuns6 = raspuns_intrebare6.ToString();
    chestionarPartialDTO.text_raspuns7 = raspuns_intrebare7.ToString();
    chestionarPartialDTO.text_raspuns8 = raspuns_intrebare8.ToString();
    chestionarPartialDTO.text_raspuns10 = raspuns_intrebare10.ToString();
    chestionarPartialDTO.text_raspuns11 = raspuns_intrebare11.ToString();
    chestionarPartialDTO.text_raspuns15 = raspunsuri_intrebarea15[i];
    chestionarPartialDTO.text_raspuns16 = raspunsuri_intrebarea16[i];

    if (chestionarPartialDTO.text_raspuns1 == "Da")
    {
        statusCod[i] = await evaluareDisciplineServicii.trimitChestionarPartial(chestionarPartialDTO,
            await Xamarin.Essentials.SecureStorage.GetAsync("token"), await Xamarin.Essentials.SecureStorage.GetAsync("tokenReimprospatare"));
    }
}

```

Figura 3.12 Trimitere răspunsuri în urma evaluării unei activități didactice din timpul semestrului

3.8 Interfața profesor

Interfața paginii ce va afișa raportul evaluărilor studenților va conține patru selectori pentru specificarea ciclului de studii, programului de studiu, disciplina și tipul activității didactice. Răspunsurile studenților la întrebările cu răspuns liber vor fi afișate într-o etichetă într-un ScrollView pentru a nu fie nevoie să derulăm foarte mult dacă sunt foarte multe răspunsuri pentru a vedea diagramele cu linii ce vor afișa tendințele pentru întrebările 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 și 13.

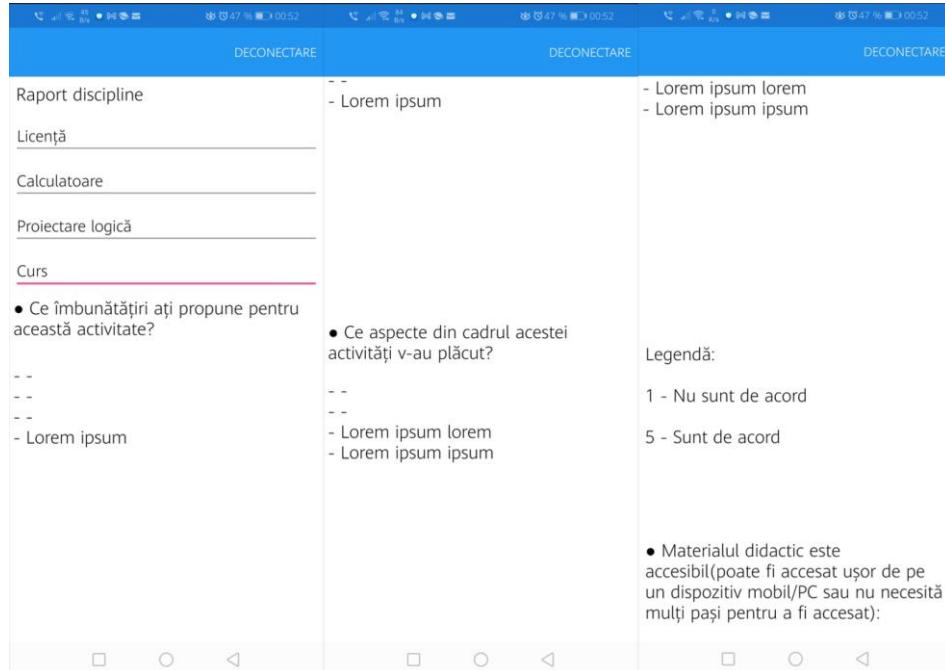


Figura 3.13 Interfață pagina de rapoarte (1)

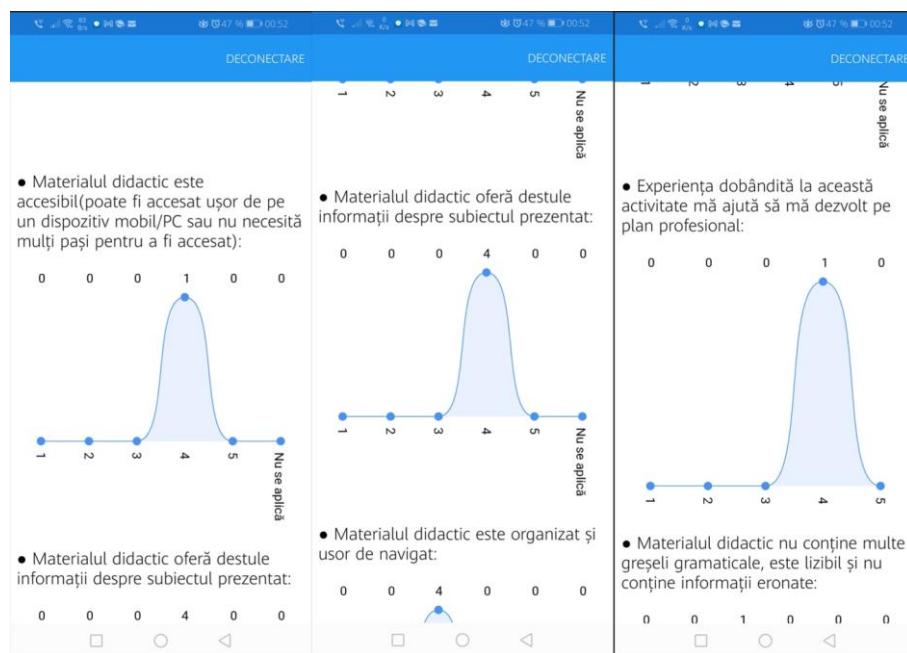


Figura 3.14 Interfață pagina de rapoarte (2)

Preluarea raportului se face printr-o solicitare API autorizată ce va apela o procedură stocată din baza de date care va returna seturi multiple de date, fiecare rezultat reprezentând o listă cu toate răspunsurile date de studenți la fiecare întrebare la disciplina și tipul activității didactice selectate de cadrul didactic. Răspunsurile apoi vor fi afișate într-o etichetă.

Diagramele cu linii le-am realizat cu biblioteca Microcharts și SkiaSharp.

```

for (int l = 0; l < 10; l++)
{
    string intrebare = "intrebarea" + (l + 4);
    foreach (RaportText raspuns in raport_complet[intrebare])
    {
        raspunsuri_intrebari_4_13[l, Convert.ToInt32(raspuns.text_raspuns) - 1]++;
    }

    List<Entry> date_diagrama = new List<Entry>();
    for (int j = 0; j < 6; j++)
    {
        if (j == 5 && l != 8)
        {
            date_diagrama.Add(new Entry(raspunsuri_intrebari_4_13[l, j]))
            {
                Color = SKColor.Parse("#4992EF"),
                Label = "Nu se aplică",
                ValueLabel = raspunsuri_intrebari_4_13[l, j].ToString()
            });
        }
        else
        {
            if (j < 5)
            {
                date_diagrama.Add(new Entry(raspunsuri_intrebari_4_13[l, j]))
                {
                    Color = SKColor.Parse("#4992EF"),
                    Label = (j + 1).ToString(),
                    ValueLabel = raspunsuri_intrebari_4_13[l, j].ToString()
                });
            }
        }
    }
    date.Add(date_diagrama);
}

```

Figura 3.15 Creare diagrame cu linii

Diagramele se creează relativ simplu: se creează un nou obiect de tip Entry și se specifică culoarea ce o va avea linia și punctele, eticheta și valoarea, pentru fiecare valoare de pe axa orizontală.

- Materialul didactic oferă destule explicații:

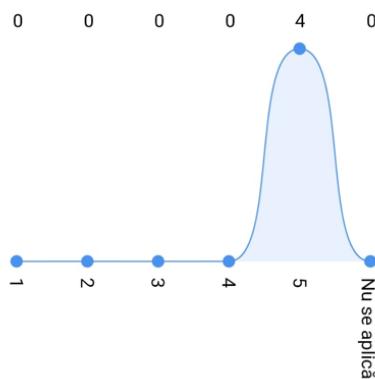


Figura 3.16 Diagramă cu linii

```

diagramaIntrebarea4.Chart = new LineChart { Entries = date.ElementAt(0), LabelTextSize = 45f,
    ValueLabelOrientation = Orientation.Horizontal, BackgroundColor = SKColors.Transparent,
    LabelColor = SKColors.Black, PointSize = 30f };

```

Figura 3.17 Personalizare diagramă cu linii

Diagramele cu linii oferite de aceste biblioteci se pot personaliza, însă opțiunile sunt destul de limitate.

3.9 Servicii API

Serviciile API folosite în acest proiect sunt structurate în trei clase: "AdministrareServicii", "EvaluareDiscipline" și "RaportEvaluariServicii".

```
2 references
public async Task<Autentificare> Logare(Logare logare)
{
    string _url = "https://otla7tanf3.execute-api.eu-central-1.amazonaws.com/Prod/api/Utilizatori/Logare";
    var json = JsonConvert.SerializeObject(logare);
    var continut = new StringContent(json, Encoding.UTF8, "application/json");
    HttpClient client = new HttpClient();
    HttpResponseMessage raspuns = await client.PostAsync(_url, continut);
    var detalii = new Autentificare();
    if (raspuns.StatusCode == System.Net.HttpStatusCode.OK)
    {
        var rezultat = await raspuns.Content.ReadAsStringAsync();
        detalii = JsonConvert.DeserializeObject<Autentificare>(rezultat);
    }
    return detalii;
}
```

Figura 3.18 Serviciu API pentru conectare

Pentru a face o solicitare API am creat un obiect HttpClient client ce va trimite o solicitare de tip POST într-o operațiune asincronă către URI-ul (Uniform Resource Identifier) specificat în variabila _url și va conține un obiect de tip Logare serializat într-un sir JSON. Dacă codul de status HTTP primit ca răspuns este 200, vom serializa conținutul HTTP primit și vom returna un obiect de tip Autentificare ce va conține token-ul de acces, token-ul de reîmprospătare, data expirării token-urilor, identificatorul utilizatorului și rolul acestuia.

Dacă trebuie să fac o solicitare API autorizată se va specifica în header-ul solicitării token-ul de acces JWT.

```
client.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer", token);
```

Figura 3.19 Specificare token de acces la o solicitare API ce necesită autorizare

Dacă token-ul a expirat, facem o altă solicitare API și cerem un nou token de acces și de reîmprospătare. Cu aceste noi token-uri, reîncercăm să facem iar solicitarea precedentă. Dacă codul de status HTTP este 200, vom salva aceste token-uri aşa cum am făcut și în pagina de conectare.

```

2 references
public async Task<HttpStatusCode> StergeUtilizator(Utilizatori nume_utilizator, string token, string token_reimprospatare)
{
    string _url = "https://otla7tanf3.execute-api.eu-central-1.amazonaws.com/Prod/api/Utilizatori/StergeUtilizator";
    var json = JsonConvert.SerializeObject(nume_utilizator);

    var continut = new StringContent(json, Encoding.UTF8, "application/json");

    HttpClient client = new HttpClient();

    client.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer", token);

    HttpResponseMessage raspuns = await client.PostAsync(_url, continut);

    if (raspuns.StatusCode == HttpStatusCode.Unauthorized || raspuns.StatusCode == HttpStatusCode.NotFound)
    {
        Token token_cerere = new Token { token = token, token_reimprospatare = token_reimprospatare };

        Token token_nou = await reimprospatareToken(token_cerere);

        client.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer", token_nou.token);

        json = JsonConvert.SerializeObject(nume_utilizator);

        continut = new StringContent(json, Encoding.UTF8, "application/json");

        HttpResponseMessage raspuns_nou = await client.PostAsync(_url, continut);

        if (raspuns_nou.IsSuccessStatusCode)
        {
            await SecureStorage.SetAsync("token", token_nou.token);

            await SecureStorage.SetAsync("tokenReimprospatare", token_nou.token_reimprospatare);
        }
    }

    return raspuns_nou.StatusCode;
}

return raspuns.StatusCode;
}

```

Figura 3.20 Serviciu API pentru ștergerea utilizatorilor

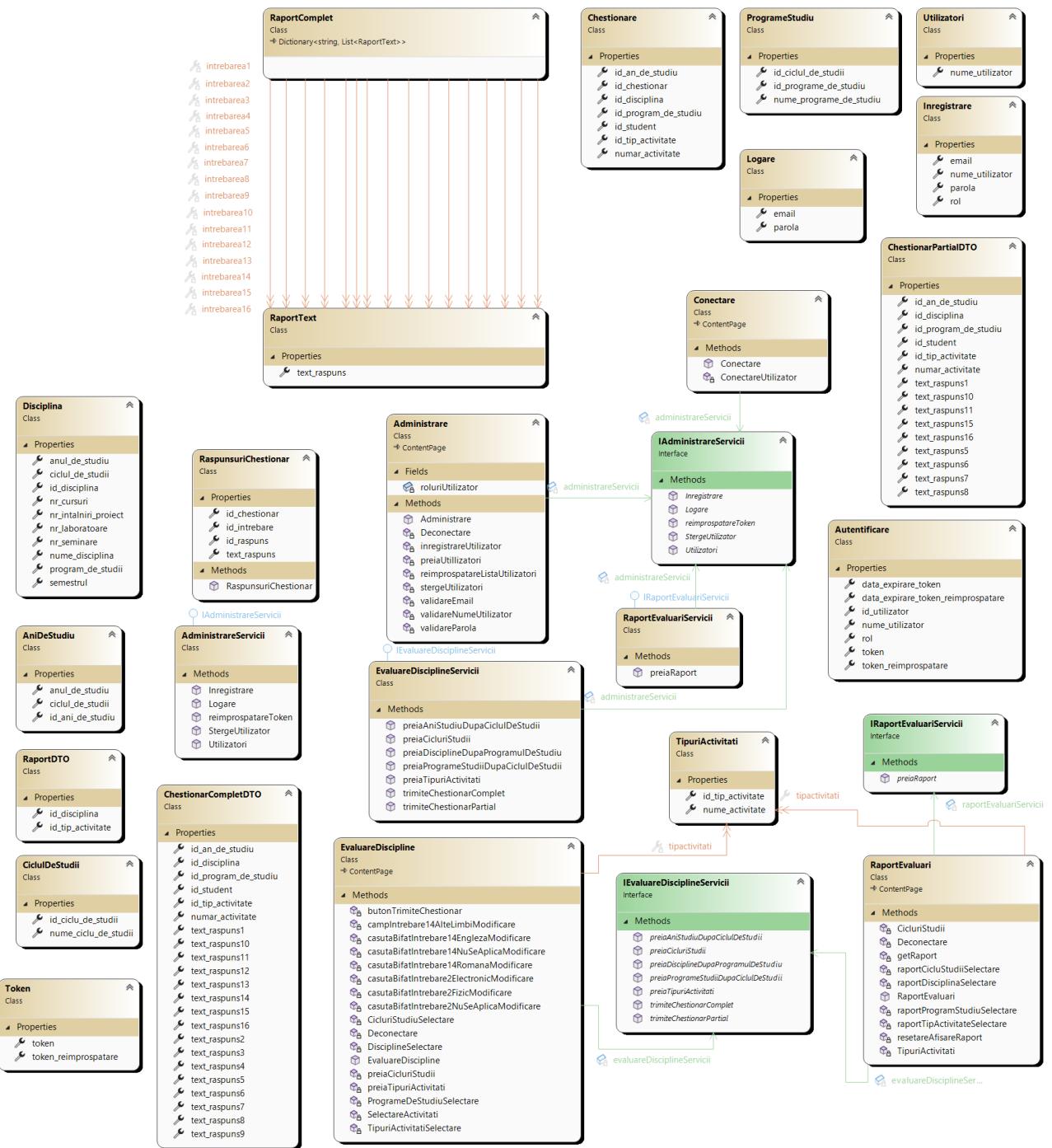


Figura 3.21 Diagrama claselor

În diagrama de mai sus sunt ilustrate clasele din acest proiect, diagramă creată cu Class Designer în Visual Studio.

3.10 Baza de date

Pentru baza de date am decis să folosesc servicii cloud pentru a putea face solicitări API de oriunde aş fi, pe Internet. Am ales să folosesc serviciile oferite de Amazon deoarece am

gratuitate timp de 12 luni să creez o bază de date MySQL în cloud activă 750 de ore pe lună. A necesitat însă crearea unui cont, introducerea datelor cardului bancar și taxa de un dolar.

Am întâmpinat probleme la crearea contului, mai precis la verificarea numărului de telefon. Dacă nu verificam numărul de telefon într-un interval cât mai scurt de la solicitarea verificării nu mai puteam să verific numărul de telefon ulterior. Fără verificarea numărului de telefon nu pot fi folosite serviciile oferite de Amazon.

Am creat o bază de date cu Amazon RDS având următoarele configurații:

- motor bază de date MySQL Community
- clasa db.t2.micro
- se află în regiunea AWS eu-central-1a (Frankfurt)
- versiune motor instantă 8.0.28
- un UCP virtualizat
- 1 GB RAM
- Memorie de stocare de 10 GiB (SSD uz general)
- fără criptare a datelor
- fără autoscalare a memoriei de stocare
- fără copii de rezerve automate
- este publică
- protocol IPv4
- motor de stocare InnoDB

Acste configurații sunt satisfăcătoare pentru cerințele acestui sistem de evaluare.

Pentru a accesa baza de date folosesc aplicația MySQL Workbench 8.0 CE și mă conectez folosind numele gazdei generat după crearea bazei de date și credențialele setate în timpul creării bazei de date.

3.11 Tabele

Am creat tabelele ”AniDeStudiu”, ”Chestionare”, ”CiclulDeStudii”, ”Discipline”, ”Intrebari”, ”ProgrameDeStudiu”, ”RaspunsuriChestionar”, ”TipuriActivitati”, ”TokenReimprospatare”.

Tabelele ”_EFMigrationsHistory”, ”AtributeRoluri”, ”AtributeUtilizatori”, ”LogareUtilizator”, ”Roluri”, ”RoluriUtilizatori”, ”TokenUtilizatori”, ”Utilizatori” au fost create automat folosind API-ul ASP.NET Core Identity pentru gestionarea utilizatorilor și rolurilor din web API-ul ASP.NET Core.

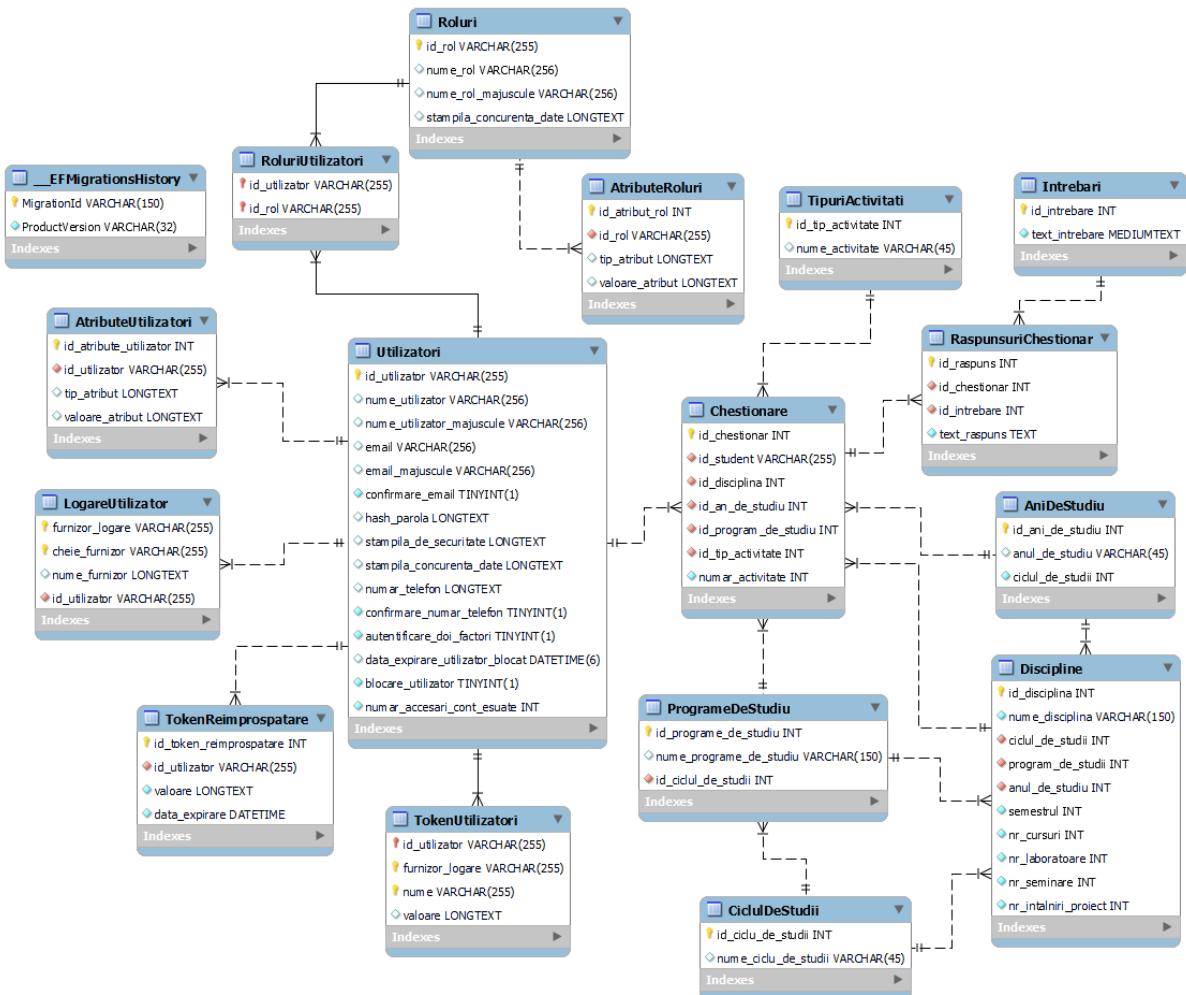


Figura 3.22 Schema bazei de date

În tabelul "AniDeStudiu" vor fi stocați anii de studiu de la licență și masterat. În tabelul "Chestionare" sunt stocate date despre evaluare precum studentul care a făcut evaluarea, disciplina care a evaluat-o, anul de studiu în care se află, programul de studiu, tipul activității didactice la care a participat și numărul activității didactice (adică săptămâna). În tabelul "CiclulDeStudii" vor fi stocate ciclurile de studii, adică licență și masterat. În tabelul "Discipline" sunt date preluate din fișele disciplinelor din cadrul facultății precum numele disciplinei, ciclul de studii, programul de studiu, anul de studiu, semestrul, numărul de cursuri, laboratoare, seminare și întâlniri de proiect. În tabelul "Intrebări" sunt stocate întrebările din chestionar. În tabelul "ProgrameDeStudiu" sunt stocate date despre programele de studiu din cadrul facultății precum numele programului de studiu și ciclul de studii. În tabelul "RaspunsuriChestionar" sunt stocate răspunsurile despre evaluare, întrebarea la care s-a răspuns și răspunsul la întrebare. În tabelul "TipuriActivitati" sunt stocate cele patru tipuri de activități didactice: curs, seminar, laborator, proiect. În tabelul "TokenReimprospatare" sunt stocate date despre token-urile de reîmprospătare folosite pentru autorizare precum valoarea acestora, data expirării și identificatorul utilizatorului.

Dintre tabelele generate doar tabelele "Roluri", "RoluriUtilizatori" și "Utilizatori" sunt folosite pentru gestionarea utilizatorilor și rolurilor. În tabelul "Roluri" sunt stocate rolurile ce le poate avea un utilizator în această aplicație: student, profesor sau administrator. În tabelul "RoluriUtilizatori" sunt stocați identificatorii utilizatorilor și rolurilor. În tabelul "Utilizatori" sunt stocate date despre utilizator precum numele utilizatorului, email-ul, confirmarea email-

ului, hash-ul pentru criptarea parolei, numărul de telefon, confirmarea numărului de telefon, stampila de securitate, stampila pentru concurență datelor, dacă autentificarea în doi factori este activată, când expiră termenul de blocare a utilizatorului dacă este cazul, dacă utilizatorul a fost blocat sau nu și numărul de accesări eşuate ale contului.

3.12 Proceduri stocate

Am creat trei proceduri stocate pentru a salva răspunsurile date la evaluările disciplinelor pentru cazul când activitatea didactică evaluată este în timpul semestrului și cazul în care este la finalul semestrului.

```
CREATE DEFINER='adminDiscipline'@'%' PROCEDURE `chestionar_complet`(`IN idStudent VARCHAR(255), IN idDisciplina INT, IN idAnDeStudiu INT, IN idProgramDeStudiu INT, IN idTipActivitate INT, IN numarActivitate INT, IN textRaspuns1 TEXT, IN textRaspuns2 TEXT, IN textRaspuns3 TEXT, IN textRaspuns4 TEXT, IN textRaspuns5 TEXT, IN textRaspuns6 TEXT, IN textRaspuns7 TEXT, IN textRaspuns8 TEXT, IN textRaspuns9 TEXT, IN textRaspuns10 TEXT, IN textRaspuns11 TEXT, IN textRaspuns12 TEXT, IN textRaspuns13 TEXT, IN textRaspuns14 TEXT, IN textRaspuns15 TEXT, IN textRaspuns16 TEXT)
BEGIN
DECLARE idChestionar INT;
INSERT INTO chestionarDiscipline.Chestionare (id_student, id_disciplina, id_an_de_studiu, id_program_de_studiu, id_tip_activitate, numar_activitate) values (idStudent, idDisciplina, idAnDeStudiu, idProgramDeStudiu, idTipActivitate, numarActivitate);
SELECT id_chestionar INTO idChestionar FROM chestionarDiscipline.Chestionare WHERE id_student = idStudent AND id_disciplina = idDisciplina AND id_an_de_studiu = idAnDeStudiu AND id_program_de_studiu = idProgramDeStudiu AND id_tip_activitate = idTipActivitate AND numar_activitate = numarActivitate;
INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns) VALUES (idChestionar, 1, textRaspuns1);
INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns) VALUES (idChestionar, 2, textRaspuns2);
INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns) VALUES (idChestionar, 3, textRaspuns3);
INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns) VALUES (idChestionar, 4, textRaspuns4);
INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns) VALUES (idChestionar, 5, textRaspuns5);
INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns) VALUES (idChestionar, 6, textRaspuns6);
INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns) VALUES (idChestionar, 7, textRaspuns7);
INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns) VALUES (idChestionar, 8, textRaspuns8);
INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns) VALUES (idChestionar, 9, textRaspuns9);
INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns) VALUES (idChestionar, 10, textRaspuns10);
```

Figura 3.23 Procedură stocată pentru salvarea răspunsurilor în urma evaluării unei activități didactice la sfârșitul semestrului (1)

```

INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns)
VALUES (idChestionar, 11, textRaspuns11);

INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns)
VALUES (idChestionar, 12, textRaspuns12);

INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns)
VALUES (idChestionar, 13, textRaspuns13);

INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns)
VALUES (idChestionar, 14, textRaspuns14);

INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns)
VALUES (idChestionar, 15, textRaspuns15);

INSERT INTO chestionarDiscipline.RaspunsuriChestionar (id_chestionar, id_intrebare, text_raspuns)
VALUES (idChestionar, 16, textRaspuns16);

END

```

Figura 3.24 Procedură stocată pentru salvarea răspunsurilor în urma evaluării unei activități didactice la sfârșitul semestrului (2)

Procedura stocată pentru salvarea răspunsurilor în urma evaluării unei activități didactice din timpul semestrului este asemănătoare, diferența fiind că introduce răspunsurile la mai puține întrebări.

Procedura stocată pentru crearea raportului la o anumită disciplină și la un anumit tip de activitate didactică presupune listarea tuturor răspunsurilor date de studenți la fiecare întrebare din chestionar la acea disciplină și la acel tip de activitate didactică.

3.12 AWS Serverless ASP.NET Core Web API

Pentru crearea serviciilor API am considerat că ar fi potrivit un web API în ASP.NET Core pentru a face legătura dintre aplicația mobilă și baza de date. Și de data aceasta m-am folosit de serviciile gratuite oferite de Amazon, mai precis AWS Lambda, AWS API Gateway și Amazon S3.

3.13 Configurări

Pentru a creația o aplicație serverless ce va fi publicată în AWS Lambda, a fost nevoie de instalarea extensiei "AWS Toolkit for Visual Studio 2022" pentru a putea folosi serviciile AWS direct din Visual Studio. De asemenea am avut nevoie de pachetele NuGet Amazon.Lambda.AspNetCoreServer (8.1.0), Dapper (2.0.138), Microsoft.AspNetCore.Authentication.JwtBearer (6.0.18), Microsoft.AspNetCore.Identity.EntityFrameworkCore (6.0.18), Microsoft.EntityFrameworkCore (7.0.7), Microsoft.EntityFrameworkCore.Design (7.0.7), Microsoft.EntityFrameworkCore.SqlServer (7.0.7), Microsoft.VisualStudio.Web.CodeGeneration.Design (6.0.14), Pomelo.EntityFrameworkCore.MySql (7.0.0), Swashbuckle.AspNetCore (6.5.0), Swashbuckle.AspNetCore.Swagger (6.5.0). Swashbuckle.AspNetCore.SwaggerGen (6.5.0). Swashbuckle.AspNetCore.SwaggerUI (6.5.0) pentru conexiunea cu baza de date, generarea de token-uri de acces JWT, folosirea comenzilor din consolă pentru migrarea și actualizarea bazei de date și utilizarea interfeței Swagger pentru a testa metodele.

Am creat o aplicație AWS Serverless (.NET Core – C#) ASP.NET Core Web API ce va fi publicată într-un bucket S3 în regiunea Frankfurt.

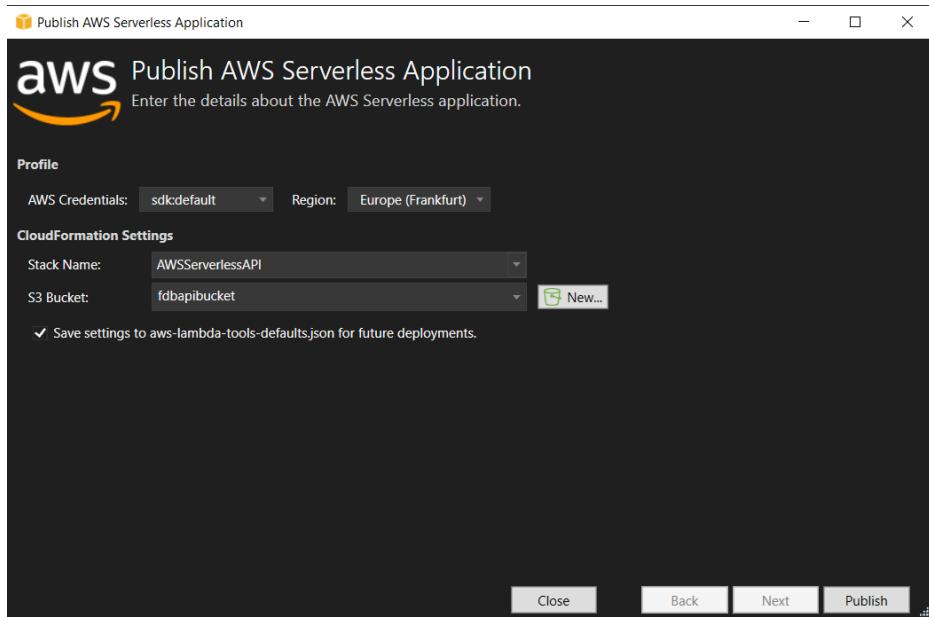


Figura 3.25 Publicare web API folosind AWS Lambda

```
{
  "Information": [
    "This file provides default values for the deployment wizard inside Visual Studio and the AWS Lambda commands added to the .NET Core CLI.",
    "To learn more about the Lambda commands with the .NET Core CLI execute the following command at the command line in the project root directory: 'dotnet lambda help'.",
    "All the command line options for the Lambda command can be specified in this file."
  ],
  "profile": "default",
  "region": "eu-central-1",
  "configuration": "Release",
  "s3-prefix": "AWS_ServerlessFeedbackDiscipline/",
  "template": "serverless.template",
  "template-parameters": "",
  "stack-name": "AWS_ServerlessAPI",
  "s3-bucket": "fdbapibucket"
}
```

Figura 3.26 Configurare aplicație pentru a putea fi publicată pe AWS Lambda

Pentru a putea gestiona utilizatorii și rolurile acestora mai ușor, m-am folosit de API-ul ASP.NET Core Identity pentru a crea tabelele automat în baza de date prin migrații.

Pentru a face legătura cu baza de date am creat o clasă ”FeedbackDisciplineContext” care extinde clasa IdentityDbContext<IdentityUser>. Am creat un DbSet pentru tabelele ”AniDeStudiu”, ”Chestionare”, ”CiclulDeStudii”, ”Discipline”, ”ProgrameDeStudiu”, ”TipuriActivitati” și ”TokenReimprospatare”. Astfel pot crea interogări cu baza de date folosind LINQ. În fișierul ”Startup.cs” am creat conexiunea cu baza de date astfel:

```
services.AddDbContext<FeedbackDisciplineContext>(optiuni =>
    optiuni.UseMySql(Configuration.GetConnectionString("FeedbackDisciplineContext"),
    ServerVersion.AutoDetect(Configuration.GetConnectionString("FeedbackDisciplineContext"))));
```

Figura 3.27 Conexiune baza de date

```
{
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "FeedbackDisciplineContext": "Server=chestionardiscipline.c7rzgix1fdxj.eu-central-1.rds.amazonaws.com;User ID=adminDiscipline;Password=12345678"
  }
}
```

Figura 3.28 Configurare conexiune baza de date în appsettings.json

În fișierul "appsettings.json" am stocat sirul de conectare necesar pentru a face legătura cu baza de date, și ce conține detalii despre credențialele folosite pentru conectarea la baza de date.

Pentru a putea folosi API-ul ASP.NET Core Identity la gestionarea utilizatorilor și rolurilor, am făcut următoarele configurații în fișierul "Startup.cs":

```
services.AddIdentity<IdentityUser, IdentityRole>(optiuni =>
{
    optiuni.SignIn.RequireConfirmedPhoneNumber = false;
    optiuni.SignIn.RequireConfirmedEmail = false;
    optiuni.SignIn.RequireConfirmedAccount = false;
    optiuni.Password.RequireDigit = true;
    optiuni.Password.RequiredLength = 6;
    optiuni.Password.RequireNonAlphanumeric = false;
    optiuni.Password.RequireUppercase = true;
    optiuni.User.RequireUniqueEmail = true;
}).AddEntityFrameworkStores<FeedbackDisciplineContext>().AddDefaultTokenProviders();
```

Figura 3.29 Configurații pentru gestionarea utilizatorilor și rolurilor

Acste configurații specifică faptul că la crearea unui utilizator nu va fi nevoie de confirmarea numărului de telefon, al email-ului sau al contului pentru ca acel cont de utilizator să fie activ. Parola de asemenea trebuie să aibă cel puțin o cifră, să aibă lungimea minimă de 6 caractere, să nu aibă neapărat caractere care nu sunt alfanumerice și să conțină o literă mare. Adresa de email trebuie să fie unică.

Pentru autorizare și autentificare este nevoie de următoarea configurație în fișierul "Startup.cs":

```
app.UseAuthentication();
app.UseAuthorization();
```

Figura 3.30 Permiterea autorizării și autentificării

```
services.AddSwaggerGen(c =>
{
    c.AddSecurityDefinition("Bearer", new Microsoft.OpenApi.Models.OpenApiSecurityScheme
    {
        Name = "Authorization",
        Type = Microsoft.OpenApi.Models.SecuritySchemeType.ApiKey,
        Scheme = "Bearer",
        BearerFormat = "JWT",
        In = Microsoft.OpenApi.Models.ParameterLocation.Header,
        Description = "Introdu token-ul JWT"
    });
    c.AddSecurityRequirement(new Microsoft.OpenApi.Models.OpenApiSecurityRequirement
    {
        {
            new OpenApiSecurityScheme
            {
                Reference = new OpenApiReference
                {
                    Type = ReferenceType.SecurityScheme,
                    Id = "Bearer"
                }
            },
            new string[]{}
        }
    });
});
```

Figura 3.31 Permite introducerea în header-ul solicitării a unui token JWT în interfața Swagger pentru autorizare

```

services.AddTransient<AWSServerlessFeedbackDiscipline.Servicii.JWTServicii>();

services.AddAuthentication(optiuni =>
{
    optiuni.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;
    optiuni.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
    optiuni.DefaultScheme = JwtBearerDefaults.AuthenticationScheme;
})

    .AddJwtBearer(optiuni =>
{
    optiuni.SaveToken = true;
    optiuni.TokenValidationParameters = new TokenValidationParameters()
    {
        ValidateIssuer = true,
        ValidateAudience = true,
        ValidateLifetime = true,
        ValidateIssuerSigningKey = true,
        ValidAudience = Configuration["Jwt:Audienta"],
        ValidIssuer = Configuration["Jwt:Emisent"],
        ClockSkew = TimeSpan.Zero,
        IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(Configuration["Jwt:Cheie"]))
    };
});

```

Figura 3.32 Permite autentificarea folosind un token JWT și validarea acestuia

Alte configurări:

```
app.UseCors(options => options.WithOrigins("*").AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader());
```

Figura 3.33 Permite interacțiunea resurselor din domenii diferite

3.14 Autorizare cu token JWT

Pentru a crea un token JWT, am creat clasa ”JWTServicii” unde am o metodă ce returnează un obiect de tip AutentificareRaspuns ce conține valoarea token-ului, data expirării și identificatorul utilizatorului. Token-ul este semnat digital folosind un algoritm de securitate HMACSHA256 pentru a împiedica modificarea acestuia de către client sau un atacator.

```

2 references
public AutentificareRaspuns CreazaToken(IdentityUser utilizator, IEnumerable<Claim> atribute)
{
    var expirare = DateTime.UtcNow.AddMinutes(30);

    var cheie = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(_configuratie["Jwt:Cheie"]));

    var token = new JwtSecurityToken(issuer: _configuratie["Jwt:Emisent"], audience: _configuratie["Jwt:Audienta"],
        expires: expirare, claims: atribute, signingCredentials: new SigningCredentials(cheie, SecurityAlgorithms.HmacSha256));

    var tokenHandler = new JwtSecurityTokenHandler();

    return new AutentificareRaspuns
    {
        token = tokenHandler.WriteToken(token),
        data_expirare_token = expirare,
        id_utilizator = utilizator.Id
    };
}

```

Figura 3.34 Creare token JWT

Pentru a crea un token de reîmprospătare ce va fi primit de client atunci când se conectează în aplicația mobilă, se generează aleator un sir în baza 64.

Pentru a genera un nou token de reîmprospătare se verifică dacă token-ul deja existent și cu cel expirat sunt identice.

```

2 references
public string CreazaTokenReimprospatare()
{
    var numar = new byte[32];
    using(var n = RandomNumberGenerator.Create())
    {
        n.GetBytes(numar);
        return Convert.ToBase64String(numar);
    }
}

1 reference
public ClaimsPrincipal TokenExpiratInformatii(string token)
{
    var tokenValidare = new TokenValidationParameters
    {
        ValidateAudience = false,
        ValidateIssuer = false,
        ValidateIssuerSigningKey = true,
        IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(_configuratie["Jwt:Cheie"])),
        ValidateLifetime = false
    };

    var tokenHandler = new JwtSecurityTokenHandler();
    SecurityToken tokenSecuritate;
    var principal = tokenHandler.ValidateToken(token, tokenValidare, out tokenSecuritate);
    var tokenSecuritateJwt = tokenSecuritate as JwtSecurityToken;

    if(tokenSecuritateJwt == null || !tokenSecuritateJwt.Header.Alg.Equals(SecurityAlgorithms.HmacSha256, StringComparison.InvariantCultureIgnoreCase))
    {
        throw new SecurityTokenException("Token invalid");
    }

    return principal;
}

```

Figura 3.35 Verificare token reîmprospătare

Am creat un controller ”UtilizatoriController” ce se va ocupa de gestionarea utilizatorilor, adică vor exista metode ce vor conecta utilizatorul în aplicație, vor înregistra un nou utilizator în baza de date, vor prelua toți utilizatorii din baza de date cu excepția administratorului, vor șterge utilizatori și vor crea noi token-uri de reîmprospătare.

```

[HttpPost("Logare")]
0 references
public async Task<ActionResult<AutentificareRaspuns>> Logare(AutentificareCerere cerere)
{
    var utilizator = await _utilizatoriManager.FindByEmailAsync(cerere.email);

    if (utilizator == null)
    {
        return BadRequest("Numele de utilizator sau parola nu sunt corecte.");
    }

    var esteValidaParola = await _utilizatoriManager.CheckPasswordAsync(utilizator, cerere.parola);

    if (!esteValidaParola)
    {
        return BadRequest("Numele de utilizator sau parola nu sunt corecte.");
    }

    if(utilizator != null && esteValidaParola)
    {
        var roluri = await _utilizatoriManager.GetRolesAsync(utilizator);
        var atribute = new List<Claim> { new Claim(ClaimTypes.Name, utilizator.UserName), new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString()) };

        foreach(var rolUtilizator in roluri)
        {
            atribute.Add(new Claim(ClaimTypes.Role, rolUtilizator));
        }

        var token = _jwt.CreazaToken(utilizator, atribute);
        var tokenReimprospatare = _jwt.CreazaTokenReimprospatare();

        var informatiiToken = _context.TokenReimprospatare.FirstOrDefault(a => a.id_utilizator == utilizator.Id);

        if(informatiiToken == null)
        {
            _context.TokenReimprospatare.Add(new TokenReimprospatare { id_utilizator = utilizator.Id, valoare = tokenReimprospatare,
            : data_expirare = DateTime.Now.AddDays(7)} );
        }
        else
    }
}

```

Figura 3.36 Metodă logare utilizator ce verifică dacă credențialele trimise de către client sunt corecte și returnează un obiect TokenReimprospatare ce conține token-ul de acces nou generat, token-ul de reîmprospătare, identificatorul utilizatorului, numele de utilizator, data expirării token-urilor și rolul utilizatorului, în cazul în care credențialele sunt valide (1)

```

        data_expirare = DateTime.Now.AddDays(7),
    }
    else
    {
        informatiiToken.valoare = tokenReimprospatare;
        informatiiToken.data_expirare = DateTime.Now.AddDays(7);
    }

    _context.SaveChanges();

    return Ok(new AutentificareRaspuns { id_utilizator = utilizator.Id, nume_utilizator = utilizator.UserName,
        token = token.token, data_expirare_token = token.data_expirare_token, token_reimprospatare = tokenReimprospatare,
        data_expirare_token_reimprospatare = DateTime.Now.AddDays(7), rol = roluri.First() });
}

return Ok(new AutentificareRaspuns { nume_utilizator = null});

```

Figura 3.37 Metodă logare utilizator ce verifică dacă credențialele trimise de către client sunt corecte și returnează un obiect TokenReimprospatare ce conține token-ul de acces nou generat, token-ul de reîmprospătare, identificatorul utilizatorului, numele de utilizator, data expirării token-urilor și rolul utilizatorului, în cazul în care credențialele sunt valide (2)

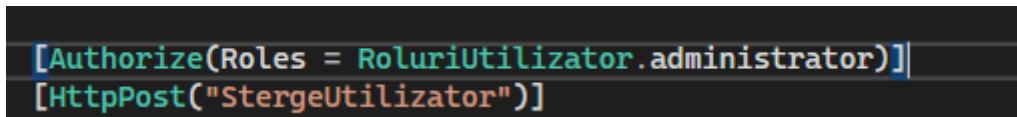


Figura 3.38 Autorizare după rol

```

[HttpPost("Token")]
public async Task<IActionResult> TokenReimprospatare(CerereTokenReimprospatare cerereToken)
{
    if(cerereToken == null)
    {
        return BadRequest("Cerere invalidă.");
    }

    string token = cerereToken.token;
    string tokenReimprospatare = cerereToken.token_reimprospatare;
    var principal = _jwt.TokenExpiratInformatii(token);
    var numeUtilizator = principal.Identity.Name;
    IdentityUser idUtilizator = await _utilizatoriManager.FindByNameAsync(numeUtilizator);
    var utilizator = _context.TokenReimprospatare.SingleOrDefault(u => u.id_utilizator == idUtilizator.Id);

    if(utilizator == null || utilizator.valoare != tokenReimprospatare || utilizator.data_expirare <= DateTime.Now)
    {
        return BadRequest("Cerere invalidă");
    }

    var tokenNou = _jwt.CreazaToken(idUtilizator, principal.Claims);
    var tokenReimprospatareNou = _jwt.CreazaTokenReimprospatare();
    utilizator.valoare = tokenReimprospatareNou;

    await _context.SaveChangesAsync();

    return Ok(new CerereTokenReimprospatare()
    {
        token = tokenNou.token,
        token_reimprospatare = tokenReimprospatareNou
    });
}

```

Figura 3.39 Metodă ce returnează un nou token de reîmprospătare și de acces clientului dacă token-ul trimis de acesta și cel existent în baza de date coincid

Token-ul de acces expiră după 30 de minute de la crearea sa. Token-ul de reîmprospătare are o durată de viață mai lungă, expirând după 7 zile.

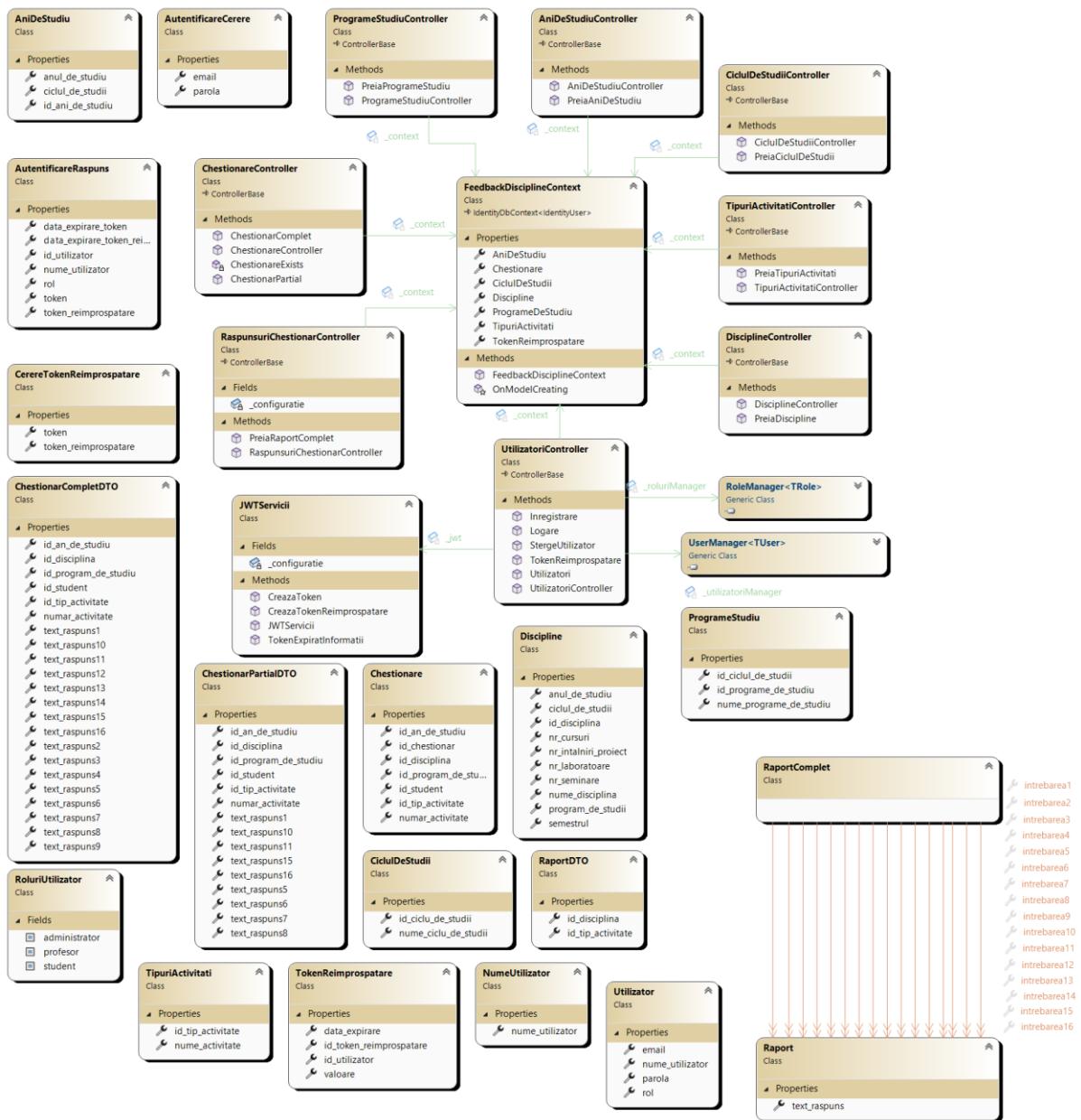


Figura 3.40 Diagrama claselor API

```
// GET: api/AnideStudiu
[Authorize(Roles = RoluriUtilizator.student + "," + RoluriUtilizator.profesor)]
[HttpGet("{ciclul_de_studii:int}")]
0 references
public async Task<ActionResult<IEnumerable<AnideStudiu>>> PreiaAnideStudiu([FromRoute] int ciclul_de_studii)
{
    var cicluDeStudii = await _context.CiclulDeStudii.FindAsync(ciclul_de_studii);

    if (cicluDeStudii == null)
    {
        return NotFound();
    }

    return await _context.AnideStudiu.Where(c => c.ciclul_de_studii == ciclul_de_studii).ToListAsync();
}
```

Figura 3.41 Metodă ce preia ciclurile de studii din baza de date și returnează o listă de obiecte AnideStudiu

```

[Authorize(Roles = "RoluriUtilizator.student")]
[HttpPost("evaluarePartiala")]
0 references
public async Task<IActionResult> CestionarPartial(CestionarPartialDTO cestionarPartialDTO)
{
    if(!CestionareExists(cestionarPartialDTO.id_student, cestionarPartialDTO.id_disciplina,
        cestionarPartialDTO.id_tip_activitate, cestionarPartialDTO.numar_activitatem))
    {
        await _context.Database.
            ExecuteSqlRawAsync("CALL cestionar_partial({0},{1},{2},{3},{4},{5},{6},{7},{8},{9},{10},{11},{12},{13},{14})",
            cestionarPartialDTO.id_student,
            cestionarPartialDTO.id_disciplina,
            cestionarPartialDTO.id_an_de_studiu,
            cestionarPartialDTO.id_program_de_studiu,
            cestionarPartialDTO.id_tip_activitate,
            cestionarPartialDTO.numar_activitatem,
            cestionarPartialDTO.text_raspuns1,
            cestionarPartialDTO.text_raspuns5,
            cestionarPartialDTO.text_raspuns6,
            cestionarPartialDTO.text_raspuns7,
            cestionarPartialDTO.text_raspuns8,
            cestionarPartialDTO.text_raspuns10,
            cestionarPartialDTO.text_raspuns11,
            cestionarPartialDTO.text_raspuns15,
            cestionarPartialDTO.text_raspuns16);
        await _context.SaveChangesAsync();
    }
    else
    {
        return Conflict();
    }
    return NoContent();
}

```

Figura 3.42 Metodă ce salvează în baza de date răspunsurile la întrebări oferite de studenți în cazul unei activități didactice din timpul semestrului și apelează o procedură stocată pentru a face asta, într-o operațiune asincronă

Capitolul 4. Manual de utilizare

La lansarea aplicației, utilizatorului i se va afișa pagina de conectare unde va trebui să își introducă credențialele, adică adresa de email și parola. Aceste credențiale vor fi furnizate de către facultate, asemănător cu conturile de Moodle.

După ce își introduce credențialele acesta va apăsa pe butonul "Conectare". Dacă credențialele nu sunt corecte, se va afișa o alertă care va informa utilizatorul că datele introduse nu sunt corecte.

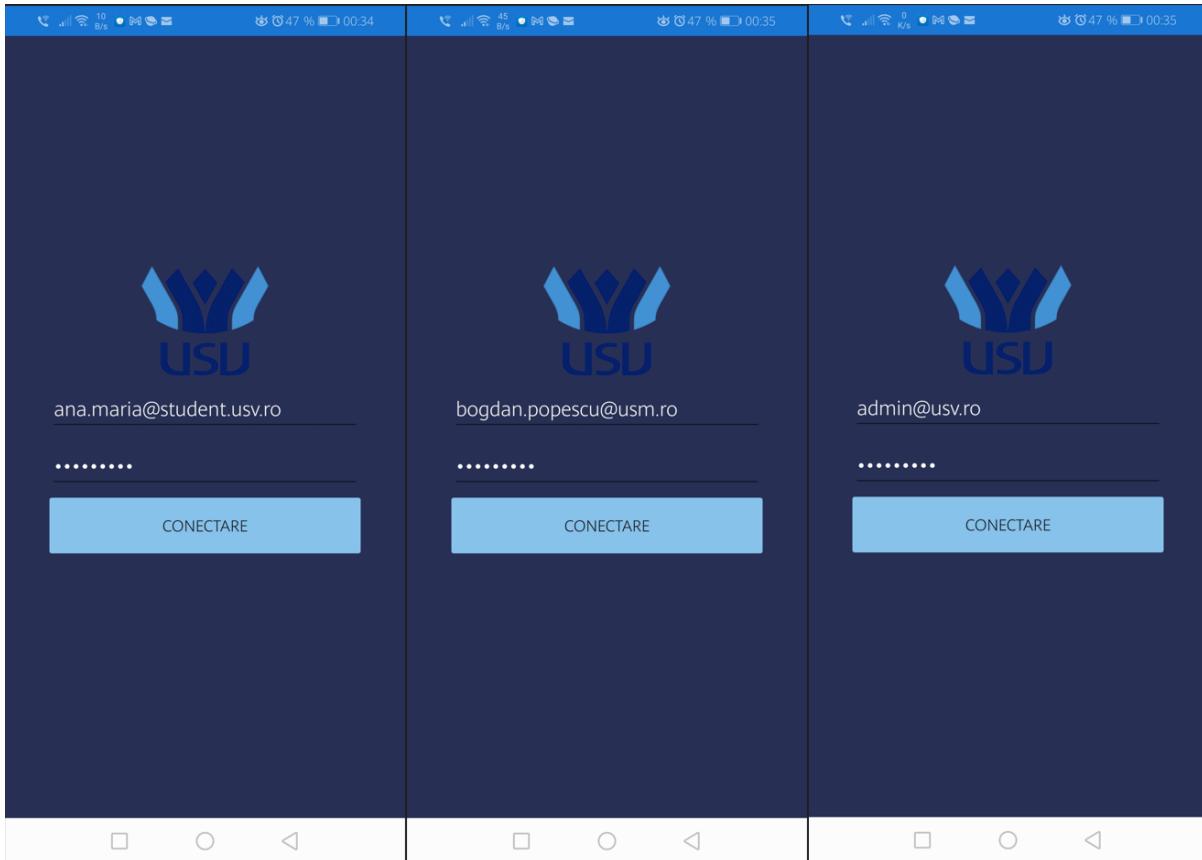


Figura 4.1 Interfață pagina de conectare și credențialele pentru un cont de student, profesor și administrator

În funcție de rolul utilizatorului, adică administrator, student sau profesor, utilizatorul va vedea o interfață diferită.

4.1 Perspectiva administratorului

În cazul administratorului, acesta va vedea următoarea interfață:

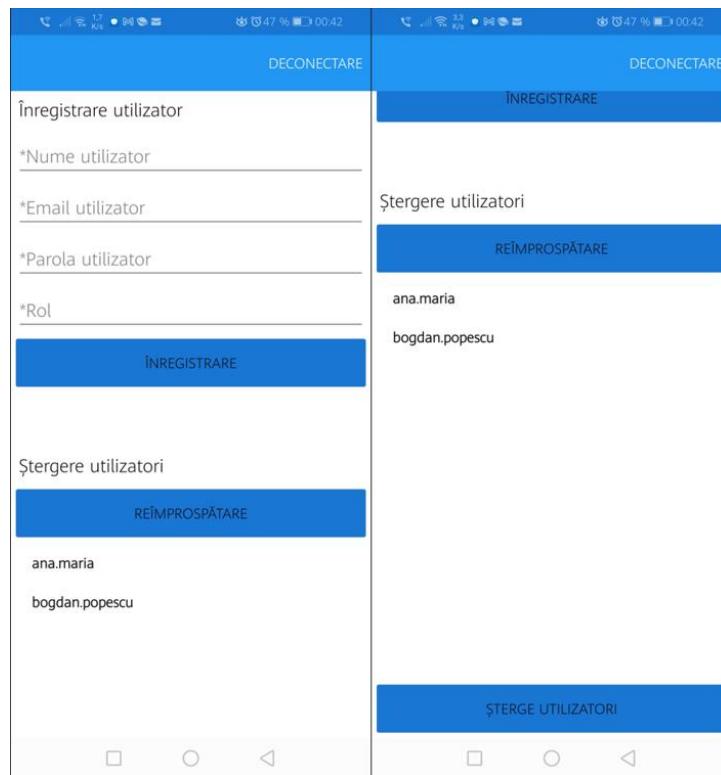


Figura 4.2 Interfața din perspectiva administratorului (1)

În secțiunea de înregistrare a utilizatorului acesta va putea introduce un utilizator nou specificând numele de utilizator, email-ul, parola și rolul acestuia. Pentru a introduce rolul acesta va selecta dintr-o listă "student" sau "profesor", la apăsarea pe câmpul respectiv.

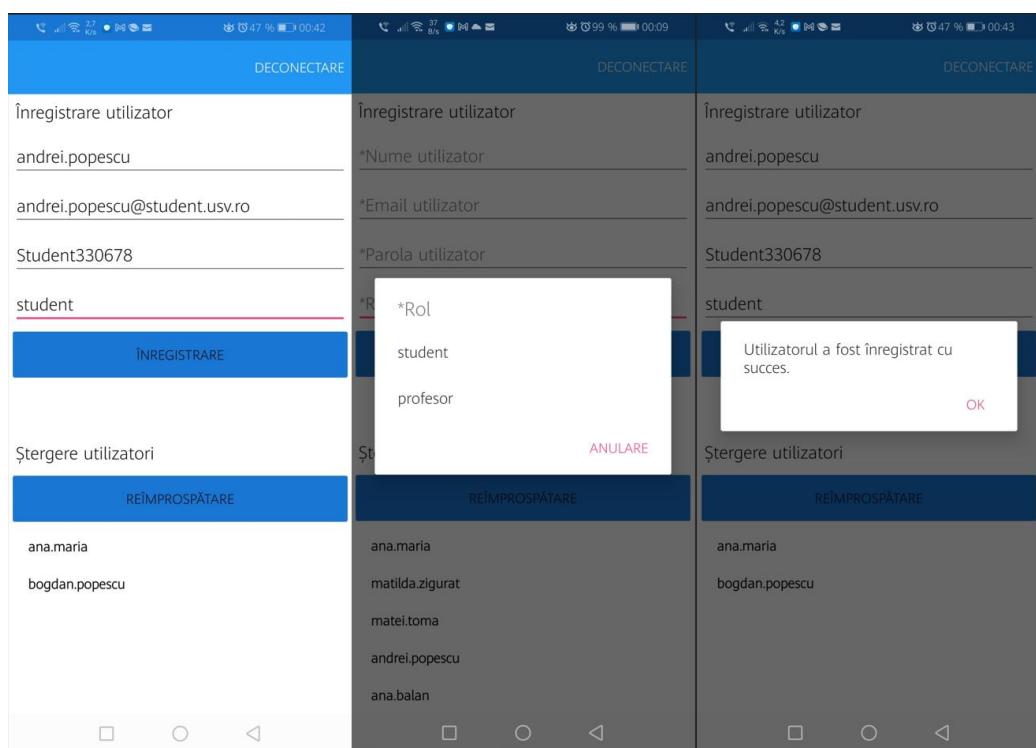


Figura 4.3 Interfața din perspectiva administratorului (2)

Dacă datele sunt corect introduse și respectă modelul, la apăsarea butonului "Înregistrare" va apărea o alertă ce îl va informa pe administrator că utilizatorul a fost înregistrat cu succes.

Numele de utilizator se va considera valid dacă respectă formatul <nume>.<nume><număr>, doar litere mici și fără diacritice. Email-ul va fi valid doar dacă respectă modelul unui email valid tipic. Parola va fi considerată validă dacă începe cu literă mare, are cel puțin 6 caractere, conține cel puțin o cifră iar caracterele sunt doar litere sau cifre. Dacă nu se respectă aceste cerințe și nu se selectează un rol, la apăsarea butonului va apărea o alertă cu un mesaj ce atenționează administratorul că datele introduse nu sunt valide și modelul care trebuie să îl respecte.

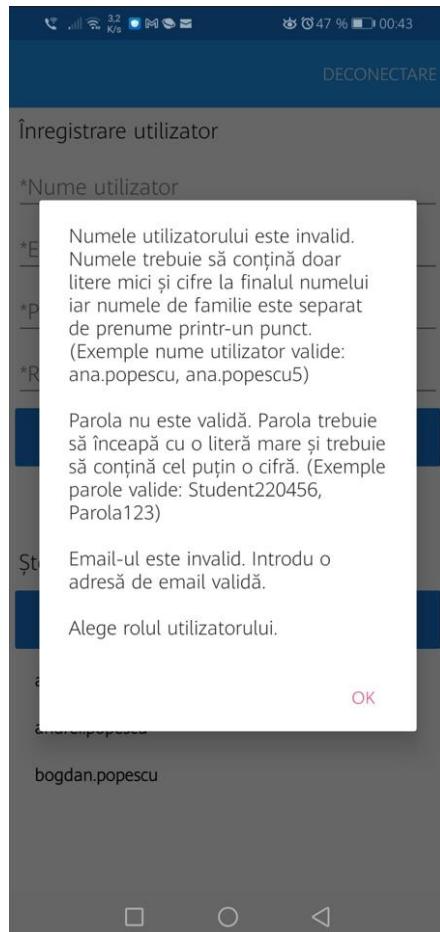


Figura 4.4 Interfața din perspectiva administratorului (3)

Pentru ștergerea utilizatorilor acesta va selecta din lista afișată utilizatorii pe care dorește să îi șteargă. La apăsarea butonului ”Șterge utilizatori” administratorul va fi informat dacă utilizatorii au fost ștersi cu succes sau nu și care dintre aceștia nu s-au șters sau care au fost ștersi.

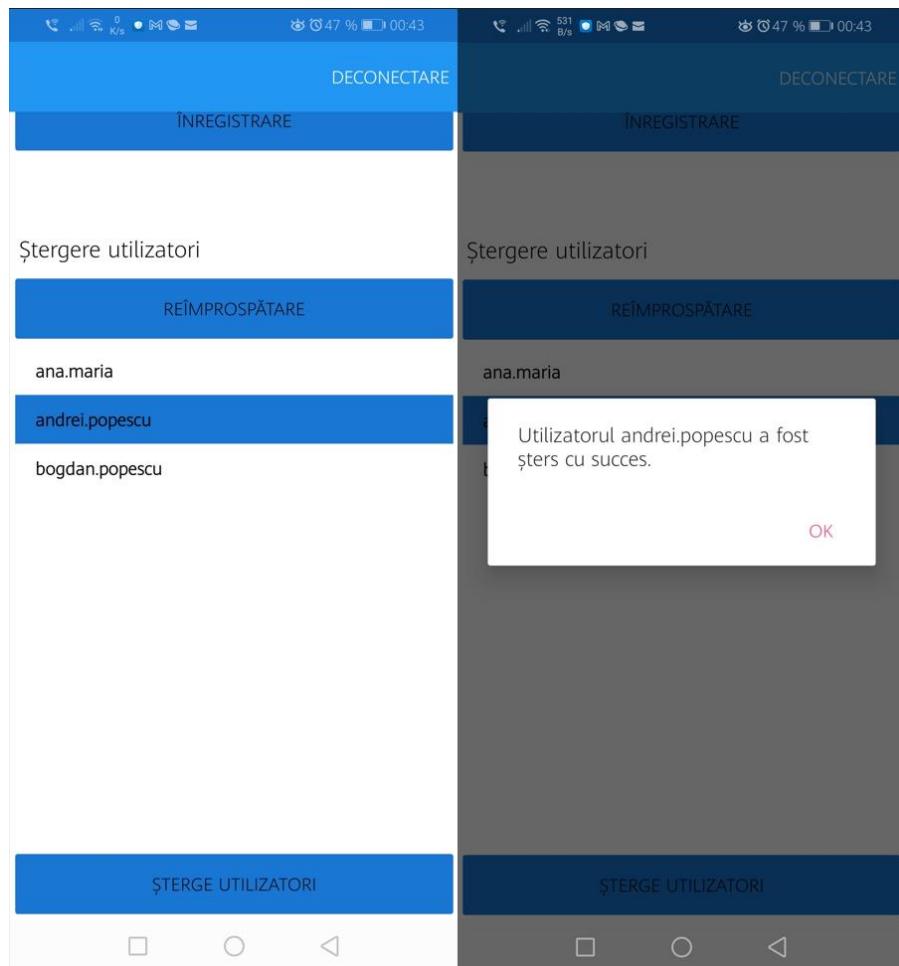


Figura 4.5 Interfața din perspectiva administratorului (4)

Pentru deconectare, utilizatorul va apăsa pe butonul de deconectare din dreapta sus, fiind redirecționat către pagina de conectare. Este la fel și pentru utilizatorii cu rolul de student sau profesor.

4.2 Perspectiva studentului

În cazul studentului, acesta va vedea următoarea interfață:

DECONECTARE		DECONECTARE		DECONECTARE		DECONECTARE	
Câmpurile cu * sunt obligatorii		*Am fost prezent(ă) la următoarele activități:		*Materialul didactic este organizat și usor de navigat:		*Gradul de dificultate al acestei activități a fost:	
*Ciclul de studii:							
*Anul de studiu:							
*Programul de studiu:							
*Disciplina:							
*Tipul activității:							
• *Selectează activitatea/activitățile pe care dorești să le evaluezi:							
• *Am fost prezent(ă) la următoarele activități:							
TRIMITE		TRIMITE		TRIMITE		TRIMITE	

Figura 4.6 Interfața din perspectiva studentului

Acesta va trebui să răspundă la niște întrebări cu răspuns închis și deschis și să menționeze ciclul de studii, anul de studiu în care se află, programul de studiu, disciplina pe care dorește să o evalueze, tipul activității, activitățile pe care dorește să le evalueze și activitățile la care a fost prezent. Dacă nu se completează toate secțiunile obligatorii, la apăsarea butonului trimite se va afișa o alertă în care va fi atenționat că nu a completat toate câmpurile obligatorii.

Pentru a introduce ciclul de studii, anul de studiu, programul de studiu, disciplina, tipul activității, activitățile pe care dorește să le evalueze și activitățile la care a fost prezent, studentul va selecta dintr-o listă datele.

Dacă nu selectează ciclul de studii, nu va putea introduce date pentru niciunul din celelalte câmpuri. Anii de studiu și programele de studiu vor fi afișate în funcție de ciclul de studii ales. Discipline vor fi afișate în funcție de ciclul de studii și programul de studii selectate. Lista activităților va apărea doar dacă disciplina selectată are ore de curs/seminar/laborator/proiect, în funcție de tipul activității selectat.

Dacă acesta selectează ultima activitate din semestrul (de exemplu activitatea cu numărul 14), se vor afișa mai multe întrebări la care va trebui să răspundă.

The screenshot shows a mobile application interface with four survey questions:

- Câmpurile cu * sunt obligatorii**: A dropdown menu with options 11, 12, 13, and 14. Option 12 is selected.
- Licență**: A dropdown menu with options 1, 12, 13, and 14. Option 14 is selected. Below it is a note: "• *Am fost prezent(ă) la următoarele activități:" followed by a list item "2".
- Calculatoare**: A dropdown menu with options 11, 12, 13, and 14. Option 12 is selected.
- Proiectare logică**: A dropdown menu with options 11, 12, 13, and 14. Option 12 is selected.
- Curs**: A dropdown menu with options 11, 12, 13, and 14. Option 12 is selected. Below it is a note: "• Selectează activitatea/activitățile pe care dorești să le evaluezi:" followed by a list item "11" and another list item "• *Materialul didactic se regăsește în format:" with options Fizic, Electronic, and Nu se aplică. Option Electronic is checked.
- *Selectează activitatea/activitățile pe care dorești să le evaluezi:** A dropdown menu with options 11, 12, 13, and 14. Option 12 is selected.
- *Am fost prezent(ă) la următoarele activități:** A dropdown menu with options 11, 12, 13, and 14. Option 12 is selected.
- *Materialul didactic este gratuit:** Radio buttons for Da (selected), Nu, and Nu se aplică.
- *Materialul didactic este accesibil (poate fi accesat ușor de pe un dispozitiv mobil/PC sau nu necesită mulți pași pentru a fi accesat):** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.
- *Materialul didactic oferă destule informații despre subiectul prezentat:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.
- *Materialul didactic este organizat și usor de navigat:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.
- *Materialul didactic oferă destule informații despre subiectul prezentat:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.

At the bottom of each section is a "TRIMITE" button.

Figura 4.7 Chestionar complet (1)

The screenshot shows a mobile application interface with five survey questions:

- Nu sunt de acord**, **Sunt de acord**, **Nu se aplică**
- *Materialul didactic oferă destule explicații:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.
- *Materialul didactic oferă o sursă de evaluare a performanțe mele:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.
- *Materialul didactic de la această activitate este în concordanță cu cel de la alte tipuri de activități din cadrul acestor discipline:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.
- *Gradul de dificultate al acestei activități a fost:** Radio buttons for Nu sunt de acord, Foarte scăzut, Foarte ridicat, and Nu se aplică.
- *Volumul de informații prezentat este adecvat:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.
- *Experiența dobândită la această activitate mă ajută să mă dezvolt pe plan profesional:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.
- *Volumul de informații prezentat este adecvat:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.
- *Materialul didactic nu conține multe greșeli gramaticale, este lizibil și nu conține informații eronate:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică.
- *Materialul didactic este disponibil în următoarele limbi:** Radio buttons for Nu sunt de acord, Sunt de acord, and Nu se aplică. Options include Romana (checked), Engleză, and Alte limbi.
- *Ce îmbunătățiri ați propune pentru această activitate?** Text input field: "Nimic".
- Alte limbi**: Text input field: "Lorem ipsum".
- *Ce aspecte din cadrul acestei activități v-au plăcut?** Text input field: "Lorem ipsum lorem ipsum".
- Alte limbi**: Text input field: "Lorem ipsum lorem ipsum".
- Alte limbi**: Text input field: "Lorem ipsum ipsum".

At the bottom of each section is a "TRIMITE" button.

Figura 4.8 Chestionar complet (2)

Ultimele două întrebări nu sunt obligatorii și nu este nevoie să completeze câmpurile text. Studentul va răspunde la întrebări prin bifarea căsuței (unde este cazul) sau selectarea butonului radio corespunzător variantei de răspuns care consideră că se potrivește. Pentru a trimite răspunsurile acesta va apăsa pe butonul "Trimite". Dacă nu a completat toate câmpurile obligatorii, se va afișa o alertă care îl va atenționa să le completeze și vor fi marcate cu roșu cele necomplete. Dacă le-a completat, se va afișa o alertă care îl va informa că răspunsurile s-au trimis cu succes dacă acesta a menționat că a fost prezent la acea activitate. Altfel, va fi informat că nu s-au trimis răspunsurile. De asemenea, este informat și că se vor trimite doar răspunsurile la activitățile la care a fost prezent. Dacă a mai evaluat deja o activitate, acesta va fi informat că a evaluat deja acea activitate. Se va specifica și numărul activității pentru toate cazurile.

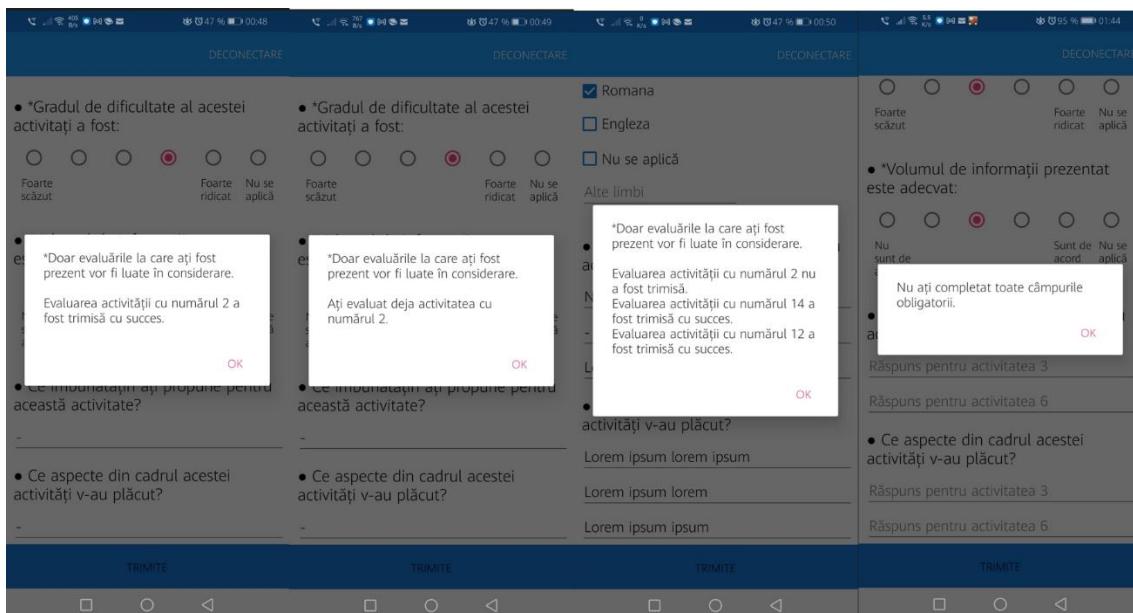


Figura 4.9 Trimitere răspunsuri

Pentru deconectare va apăsa pe butonul din dreapta sus.

4.3 Perspectiva profesorului

În cazul profesorului, acesta va vedea următoarea interfață:

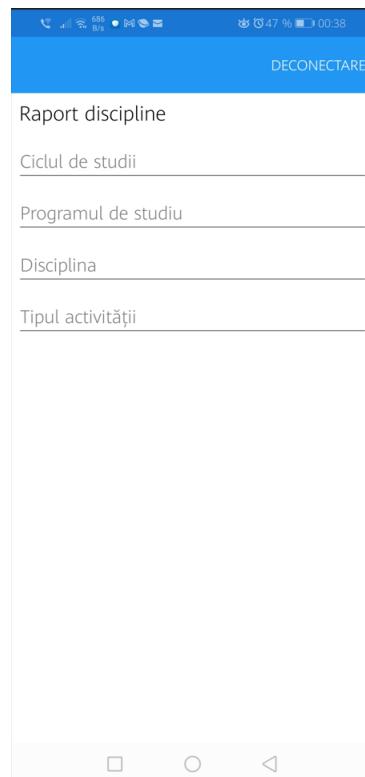


Figura 4.10 Interfața din perspectiva profesorului

Pentru a putea vizualiza raportul pentru o anumită disciplină, acesta va trebui să specifice ciclul de studii, programul de studii, disciplina și tipul activității. Aceste date se vor selecta dintr-o listă.

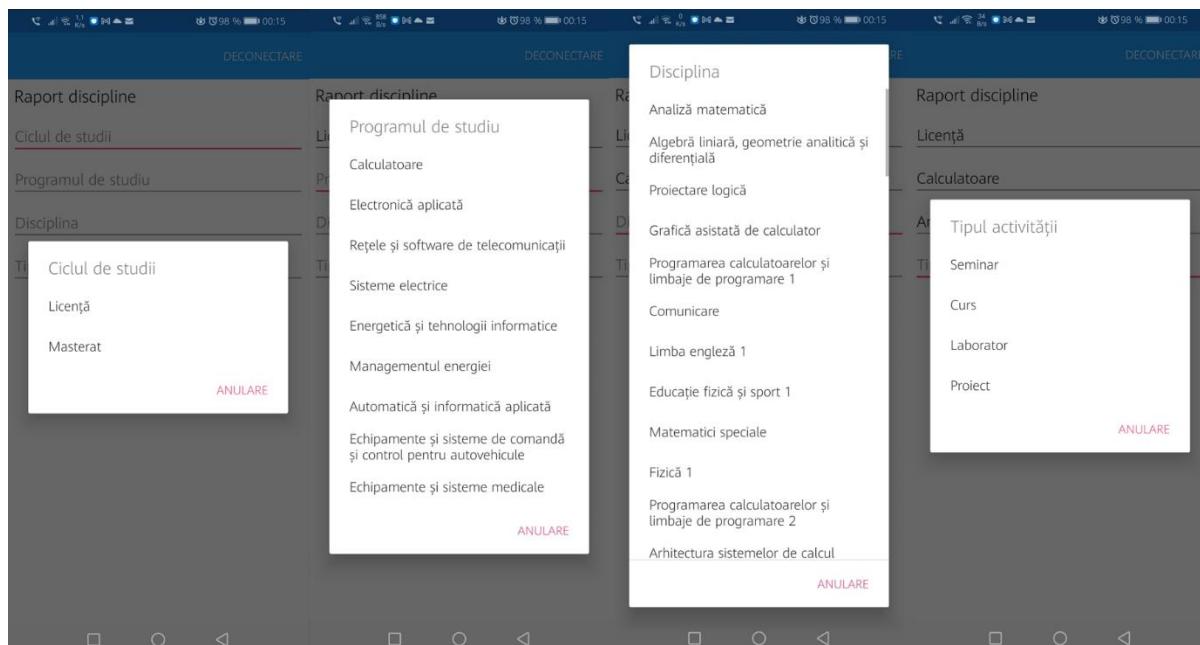


Figura 4.11 Introducere date

Dacă disciplina selectată nu are ore de curs/laborator/seminar/proiect, nu se va afișa nimic. Dacă are, se vor afișa răspunsurile date de studenți la întrebările cu răspuns liber. Dacă sunt multe, acestea se pot derula. Pentru celelalte întrebări se vor afișa diagrame cu linii ce vor afișa linia de tendință.

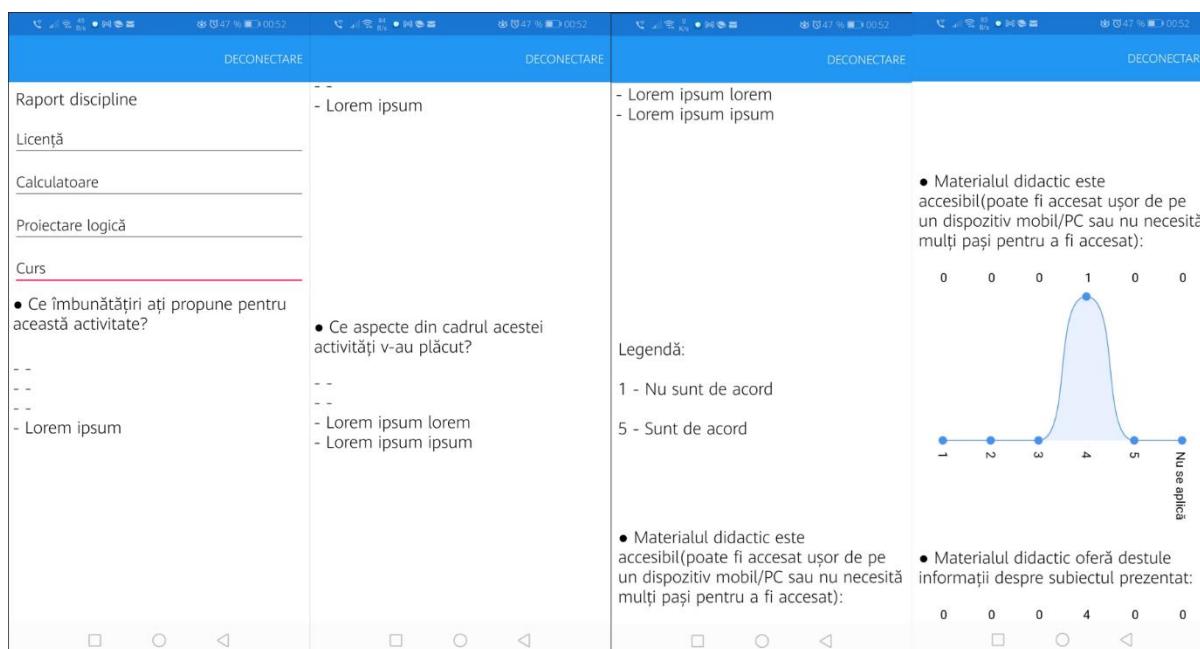


Figura 4.12 Afisare raport (1)



Figura 4.13 Afisare raport (2)

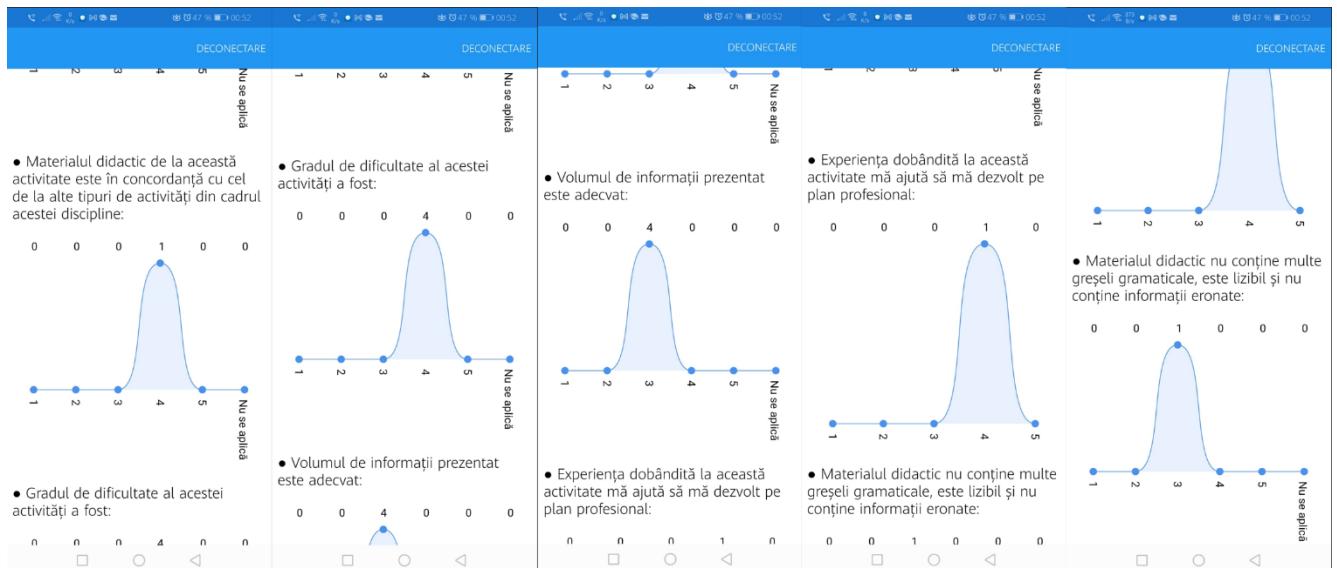


Figura 4.14 Afisare raport (3)

Pentru deconectare va apăsa pe butonul de deconectare din dreapta sus.

Capitolul 5. Concluzii și dezvoltări viitoare

Acest sistem de evaluare a reprezentat o provocare și sunt multe aspecte ce s-ar putea îmbunătății. De exemplu, răspunsurile la întrebări ar putea fi arhivate și salvate într-un fișier. Această arhivă ar fi utilă și atunci când trebuie șterse conturile ce nu mai sunt active deoarece studenții fie au absolvit sau s-au lăsat de facultate, fără a pierde acele evaluări.

S-ar putea implementa pe viitor funcționalități de adăugare a disciplinelor direct din interfața administratorului deoarece în momentul de față se poate face doar din baza de date.

Un alt aspect important ce ar putea fi îmbunătățit ar fi culorile utilizate și modul în care acestea sunt afișate în funcție de tema telefonului. Pe telefoanele ce au o versiune mai recentă de Android (10.0+) și au modul întunecat activat, butoanele radio nu se văd mai deloc, ceea ce ar presupune crearea unor butoane radio personalizate pentru a rezolva această problemă.

Afișarea raportului ar putea fi modificată pentru a avea un aspect mai plăcut și s-ar putea adăuga și moduri noi de reprezentare a datelor.

În prezent aplicația nu menține utilizatorul conectat după ce părăsește aplicația, fiind deconectat după scurt timp după ce a fost închisă. Acest lucru ar putea fi îmbunătățit prin verificarea token-ului de reîmprospătare.

Timpii de răspuns al aplicației reprezintă un alt aspect ce poate fi deranjant, mai ales la conectare și preluarea informațiilor. Aici ar putea exista mai multe motive pentru timpii de răspuns mari și probabil și multe soluții de optimizare.

O altă problemă întâmpinată a fost testarea aplicației pe un dispozitiv iOS, problemă ce nu am putut să o rezolv deoarece ar presupune necesitatea unui Mac sau plata unei taxe de înrolare în programul dezvoltatorilor Apple de 99 de dolari, astfel nu am putut include imagini cu interfața aplicației pe un dispozitiv iOS.

ACEstea sunt doar câteva dintre aspectele ce ar putea fi remediate pe viitor, cu excepția ultimei probleme.

În concluzie, aplicația reușește să salveze răspunsurile oferite de studenți la întrebările din chestionar și să afișeze un raport al acestor date, fiind scopul principal al acestui sistem.

Bibliografie

- [1] *Number of smartphone mobile network subscriptions worldwide from 2016 to 2022, with forecasts from 2023 to 2028*, <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>, Ericsson, 2022
- [2] *Global number of internet users 2012-2022, by operating system*, by operating system, <https://www.statista.com/statistics/543185/worldwide-internet-connected-operating-system-population/>, Petroc Taylor, 2023
- [3] *Annual number of global mobile app downloads 2016-2022* , <https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/>, L. Ceci, 2023
- [4] *Distribution of web traffic in Romania 2022, by mobile OS*, <https://www.statista.com/statistics/1214250/romania-share-of-web-traffic-by-mobile-operating-system/>, Statista Research Department, 2023
- [5] *What is cloud computing?*, <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing>, Microsoft, 2023
- [6] *Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach Nearly \$600 Billion in 2023*, <https://www.gartner.com/en/newsroom/press-releases/2022-10-31-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023#:~:text=Worldwide%20end%2Duser%20spending%20on,18.8%25%20growth%20forecast%20for%202022>, Meghan Rimol DeLisi, 2022
- [7] *Gartner's 2023 Top Strategic Technology Trends*, <https://www.gartner.com/en/information-technology/trends/top-technology-trends-ai-security-gb-pd>, Gartner, 2023
- [8] *Google Cloud Brand Pulse Survey*, https://docs.google.com/presentation/d/1P8app8FaqFaOlsM4bs6tRlvYsl1erP6rW7lliQJclLU/edit?pli=1&resourcekey=0-eDHgbzM9JNC0TvUrsGQg1A#slide=id.g200b2c3323b_3_5, Google Cloud, 2022
- [9] *The Digital Crunch Time: 2022 State of APIs and Applications*, <https://cloud.google.com/resources/state-of-apis-and-applications-report>, Google Cloud, 2022
- [10] *Peeking Behind the Curtains of Serverless Platforms*, <https://www.usenix.org/system/files/conference/atc18/atc18-wang-liang.pdf>, Liang Wang s.a., 2018
- [11] *AWS Lambda*, <https://aws.amazon.com/lambda/>, Amazon Web Services, 2023
- [12] *AWS Lambda Features*, <https://aws.amazon.com/lambda/features/?pg=ln&sec=hs>, Amazon Web Services, 2023
- [13] *What is AWS Lambda?*, <https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>, Amazon Web Services, 2023
- [14] *What is Amazon Relational Database Service (Amazon RDS)?*, <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>, Amazon Web Services, 2023

- [15] *DB instance classes*,
<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.DBInstanceClass.html>, Amazon Web Services, 2023
- [16] *AWS Free Tier*, https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all, Amazon Web Services, 2023
- [17] *Amazon S3 Features*, <https://aws.amazon.com/s3/features/>, Amazon Web Services, 2023
- [18] *Amazon S3*, <https://aws.amazon.com/s3/>, Amazon Web Services, 2023
- [19] *Amazon API Gateway*, <https://aws.amazon.com/api-gateway/>, Amazon Web Services, 2023
- [20] *Amazon API Gateway Features*, <https://aws.amazon.com/api-gateway/features/>, Amazon Web Services, 2023
- [21] *What is Amazon API Gateway?*,
<https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>, Amazon Web Services, 2023
- [22] *AWS CloudFormation*, <https://aws.amazon.com/cloudformation/>, Amazon Web Services, 2023
- [23] *AWS CloudFormation Features*,
<https://aws.amazon.com/cloudformation/features/?pg=ln&sec=hs>, Amazon Web Services, 2023
- [24] *What is Xamarin?*, <https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>, Justin Johnson §.a., 2022
- [25] *What is Xamarin.Forms?*, <https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin-forms>, Justin Johnson §.a., 2021
- [26] *Overview of ASP.NET Core*, <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0>, Daniel Roth §.a., 2022