

UNILINK-APLICAȚIE MOBILĂ PENTRU LICEE

Candidat: Anamaria-Victoria TUDUCE

Coordonator științific: Ș.I.dr.inf. Cristian ZIMBRU

Sesiunea: Iunie 2024

REZUMAT

În era digitală în care trăim, tehnologia a devenit un element central în transformarea sistemului educațional. Cu toate acestea, mulți liceeni se confruntă încă cu dificultăți în accesarea informațiilor relevante și în comunicarea eficientă în cadrul școlii lor. Aceste provocări pot afecta în mod semnificativ experiența lor academică și pot limita potențialul lor de învățare.

Scopul acestei platforme este să modernizeze și să simplifice procesele academice și administrative, facilitând accesul la resurse educaționale și îmbunătățind comunicarea între elevi, cadre didactice și personalul școlar. Implementarea unei astfel de platforme oferă numeroase beneficii.

Pentru a redresa aceste probleme, am implementat o platformă mobilă de campus virtual, adaptată specific nevoilor și cerințelor unui liceu. Elevii vor avea posibilitatea să acceseze rapid și ușor informațiile despre programele școlare, orarele cursurilor, materialele educaționale și alte resurse relevante, direct de pe dispozitivele lor mobile. De asemenea, platforma va facilita comunicarea între elevi și cadrele didactice, permițându-le să pună întrebări, să primească feedback și să colaboreze mai eficient în cadrul proiectelor și activităților școlare. Prin centralizarea și optimizarea resurselor educaționale și administrative, platforma mobilă de campus virtual va contribui la creșterea eficienței și la îmbunătățirea experienței educaționale pentru toți cei implicați în procesul de învățământ.

Prin aceasta lucrare de licență, imi propun dezvoltarea unei platforme mobile de campus virtual pentru liceeni, care reprezintă o soluție modernă și eficientă pentru abordarea provocărilor din domeniul educației. Această platformă nu numai că simplifică viața elevilor, dar și îmbunătățește calitatea educației și promovează o colaborare mai bună între toate părțile implicate în procesul de învățământ. Astfel, propunerea mea de proiect reprezintă un pas important către transformarea sistemului educațional într-unul mai adaptat cerințelor și tehnologiilor contemporane, menit să ofere o experiență educațională mai eficientă și mai relevantă pentru elevi în epoca digitală în care trăim.

1	INTRODUCERE	5
1.1	INFORMAȚII GENERALE	5
1.2	Tema	6
1.3	Obiectivele lucrării	7
1.4	Contextul lucrării	7
1.5	Motivație	8
2	APLICAȚII ASEMĂNĂTOARE	10
2.1	Campus Virtual UPT	10
2.2	Student UPT	2
2.3	eLearning UVT	1
2.4	UniCampus	1
2.5	Comparație aplicații	1
3	Fundamente Teoretice	1
3.1	Tehnologii folosite	1
3.1.1	Visual Studio Code	1
3.1.2	Flutter	3
3.1.3	Dart	5
3.1.4	Material Design	6
3.1.5	Firebase	7
3.1.6	API-uri și SDK-uri native Android	8
3.1.7	Android Studio	9
3.1.8	Emulator Android	10
4	Proiectarea și Implementarea	11
4.1	Arhitectura	11
4.2	Arhitectura bazei de date	12
4.3	Interfața aplicației	14
5	Bibliografie	1

1 INTRODUCERE

1.1 INFORMAȚII GENERALE

Această lucrare de licență explorează dezvoltarea unei aplicații mobile destinată unui spațiu online academic, cu scopul de a îmbunătăți comunicarea și colaborarea între membrii unui liceu. Proiectul a avut ca punct de plecare analiza cerințelor și specificațiilor necesare pentru a satisface nevoile diferitelor grupuri de utilizatori: elevi, profesori și administratori.

În etapa de proiectare, s-a pus accent pe crearea unei arhitecturi intuitive și a unui design prietenos, astfel încât utilizatorii să poată naviga și interacționa cu aplicația în mod eficient și plăcut. Funcționalitățile principale incluse în aplicație acoperă sistemul de mesagerie, secțiuni pentru anunțuri, evenimente, materiale didactice și note, adaptate nevoilor mediului academic.

Implementarea aplicației a fost realizată cu tehnologii moderne, asigurându-se structurarea codului sursă conform principiilor dezvoltării software pentru a garanta performanța și scalabilitatea soluției.

În zilele noastre pare normal să avem smartphone-ul nostru plin de aplicații, dar acum 30 de ani nu a fost. Aplicațiile mobile fac parte din zilele noastre și au făcut din mobil un element indispensabil în viața noastră. O reflecție a acestui lucru este că suntem atât de atrași de aplicațiile mobile încât, așa cum este indicat în Ditrencia, „aproape 70% din timpul pe care îl dedicăm zilnic mobilului, folosim una dintre aplicații”.

Primele aplicații mobile au apărut la sfârșitul anilor '90. Unele dintre acestea au fost aplicațiile de contacte, ordinea de zi, editorii de tonuri de apel. Acestea au fost și sunt aplicații care acopereau funcții de bază, dar care facilitau gestionarea zilelor noastre de zi cu zi.

Aplicațiile mobile au evoluat și au apărut primele jocuri populare, cum ar fi *Tetris*, instalat pentru prima dată pe un telefon mobil în 1994, în Danemarca; sau celebrul joc Nokia, *Snake*, pe care l-a jucat toată lumea, indiferent de generația din care a făcut parte.[1]

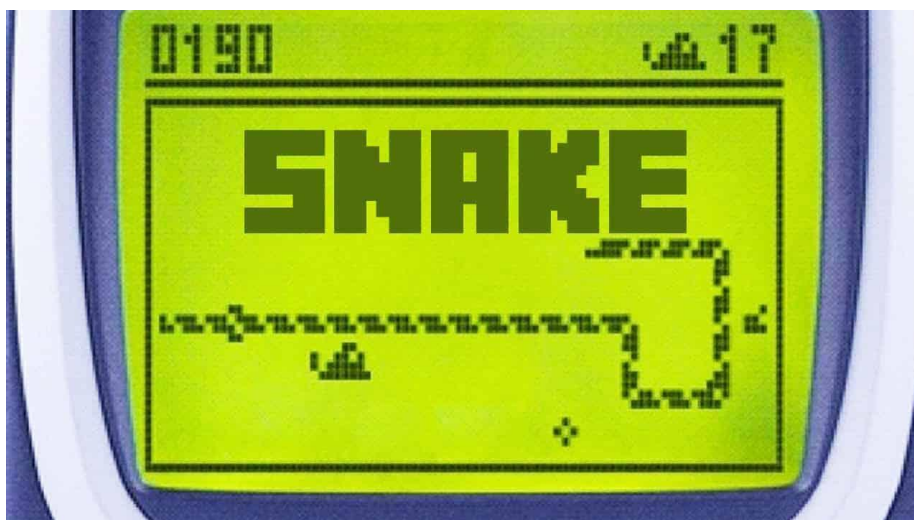


Figura 1.1.1 Snake-Interfața jocului pe telefonul mobil Nokia [2]

Prima aplicație de pe un telefon mobil din lume este considerată a fi "Snake" (Șarpele), dezvoltată de Taneli Aranto pentru Nokia și lansată în 1997. Jocul "Snake" a devenit extrem de popular și a fost inclus pe mai multe modele de telefoane Nokia, devenind unul dintre cele mai cunoscute și jucate jocuri pe telefoanele mobile din anii '90. Deși "Snake" este adesea menționat ca fiind prima aplicație de pe un telefon mobil, termenul de "aplicație" în contextul telefoanelor mobile s-a dezvoltat și evoluat odată cu avansul tehnologic.[3]

Platformele de social media au devenit fundamentale în experiența online a majorității utilizatorilor de internet. De la Facebook, WhatsApp și Instagram, la TikTok și YouTube, aceste platforme oferă utilizatorilor posibilitatea de a se conecta, a împărtăși conținut și a interacționa într-un mod digital.

Internetul Lucrurilor, cunoscut și sub numele de IoT (Internet of Things), se referă la interconectarea obiectelor fizice cu internetul, facilitând comunicarea și interacțiunea acestora între ele. Exemple concrete din viața de zi cu zi includ dispozitivele inteligente pentru locuințe, vehiculele conectate și tehnologia purtabilă, toate bazându-se pe principiile IoT.

Această lucrare își propune să contribuie la dezvoltarea cunoștințelor și competențelor în managementul academic prin crearea unei aplicații mobile care să simplifice gestionarea și organizarea activităților academice. Aplicația va oferi o soluție digitală eficientă pentru coordonarea și comunicarea între elevi, profesori și administratori, adaptată nevoilor specifice ale mediului academic.

1.2 Tema

În contextul evoluției rapide a tehnologiei și a importanței pe care aceasta o are în societatea modernă, integrarea soluțiilor tehnologice în domeniul educațional devine din ce în ce mai relevantă. Tema acestei lucrări vizează dezvoltarea și implementarea unei aplicații mobile specializate în managementul academic, cu scopul de a optimiza și moderniza procesele specifice mediului educațional.

Aplicația mobilă propusă își propune să faciliteze gestionarea activităților academice, comunicarea și colaborarea între elevi, profesori și părinți. Prin intermediul acestei platforme digitale, utilizatorii vor avea acces la informații actualizate despre ore, note, anunțuri și evenimente școlare, precum și la un sistem de mesagerie intern pentru comunicarea eficientă între membrii comunității educaționale.

Pe lângă funcționalitățile de bază, aplicația va oferi și instrumente suplimentare pentru organizarea materialelor didactice, urmărirea progresului academic al elevilor și gestionarea resurselor educaționale. Interfața intuitivă și design-ul prietenos al aplicației vor facilita utilizarea și adoptarea acesteia de către diferitele categorii de utilizatori. Prin această lucrare, se urmărește nu doar dezvoltarea unei soluții tehnologice, ci și evaluarea impactului pe care o astfel de aplicație îl poate avea asupra eficienței și calității proceselor educaționale. De asemenea, se va analiza și feedback-ul primit de la utilizatori, identificând astfel posibile îmbunătățiri și optimizări pentru viitoarele versiuni ale aplicației.

Designul site-ului este simplist și modern, folosind culori plăcute și fonturi lizibile, ceea ce facilitează navigarea și înțelegerea funcționalităților aplicației. Imaginile sunt clare și ilustrează fiecare opțiune în detaliu.

În concluzie, tema propusă pentru lucrare reflectă o abordare integrată a tehnologiei în domeniul educațional, punând accentul pe beneficiile și oportunitățile pe care le poate aduce utilizarea aplicațiilor mobile în managementul academic.

1.3 Obiectivele lucrării

Obiectivele acestei lucrări sunt clare și bine definite pentru a asigura succesul și relevanța proiectului de dezvoltare a platformei mobile de campus virtual pentru liceeni. Prin abordarea lor în mod sistematic, se urmărește atingerea unor rezultate tangibile și semnificative în modernizarea și optimizarea experienței educaționale.

Obiectivul principal al acestei lucrări de licență este de a asigura funcționalitatea și fiabilitatea platformei. Prin dezvoltarea unei aplicații mobile robuste și ușor de utilizat, adaptată specific nevoilor și preferințelor liceenilor, se creează baza pentru o experiență educațională modernă și eficientă. Integrarea unui design intuitiv și ergonomic este esențială pentru a facilita accesul la informații și navigarea pe platformă, contribuind la o interacțiune fluidă și plăcută pentru utilizatori.

Un alt obiectiv central este să aprofundăm cunoștințele în limbajul Dart, framework-ul Flutter, mediul Android Studio și platforma Firebase. Pentru că acest demers va permite dezvoltatorului să proiecteze și să dezvolte aplicații mobile avansate și fiabile cu Flutter, să integreze servicii complexe prin Firebase și să beneficieze de eficiența mediului Android Studio în procesul de dezvoltare.

Un alt obiectiv este să dezvoltăm o aplicație care să evalueze nevoile și preferințele pieței pentru platformele mobile. Aceasta implică cercetarea cerințelor utilizatorilor și identificarea funcționalităților solicitate, permițându-ne să creăm aplicații adaptate și inovatoare care să răspundă eficient acestor cerințe.

Un alt obiectiv este asigurarea securității, performanței și utilizabilității în experiența utilizatorului. Aceste elemente sunt cruciale pentru îmbunătățirea continuă a platformei, având în vedere importanța utilizatorilor.

Un obiectiv final este asigurarea evoluției continue a aplicației prin implementarea unei opțiuni de feedback, unde utilizatorii pot sugera noi funcționalități. Scopul este identificarea nevoilor pieței și dezvoltarea unei strategii pe termen lung pentru a transforma această platformă mobilă într-un competitor puternic față de alte platforme existente.

1.4 Contextul lucrării

Aplicația mea mobilă de campus virtual se află în centrul unei transformări digitale, adaptându-se la nevoile în continuă schimbare ale elevilor și profesorilor din mediul academic contemporan. Într-o societate în care tehnologia devine din ce în ce mai esențială în educație, campusul virtual vine ca o soluție modernă și eficientă pentru gestionarea

cursurilor, comunicarea între membrii comunității academice și facilitarea procesului de învățare.

Cu o interfață intuitivă și ușor de navigat, aplicația oferă acces rapid la materialele cursurilor, note, anunțuri și evenimente, toate centralizate într-un singur loc. Fie că ești elev în căutare de resurse pentru un proiect sau profesor interesat să încarce materiale noi, platforma este concepută pentru a oferi o experiență fluidă și adaptată nevoilor fiecărui utilizator.

Un aspect esențial în dezvoltarea aplicației a fost securitatea și performanța. Într-un mediu în care confidențialitatea și integritatea datelor sunt prioritare, am implementat măsuri robuste de securitate pentru a proteja informațiile utilizatorilor. De asemenea, performanța aplicației este optimizată pentru a asigura o experiență rapidă și fără întreruperi, indiferent de dispozitivul folosit. Un alt element cheie este implicarea comunității. Prin includerea unui formular de feedback, utilizatorii au posibilitatea de a sugera noi funcționalități sau îmbunătățiri care pot face aplicația și mai utilă și relevantă pentru nevoile lor. Acest canal de comunicare deschis ne permite să identificăm rapid nevoile pieței și să adaptăm platforma în consecință, contribuind la consolidarea poziției noastre ca un competitor de top în domeniul aplicațiilor de campus virtual.

În concluzie, aplicația noastră de campus virtual este o soluție comprehensivă și inovatoare care vine în întâmpinarea cerințelor academice moderne. Prin combinarea tehnologiei avansate, securității robuste și interacțiunii continue cu comunitatea noastră, ne propunem să oferim o platformă care nu doar îndeplinește, ci și depășește așteptările utilizatorilor noștri.

1.5 Motivație

Am decis să abordez această temă de dezvoltare a unei aplicații mobile pentru campus virtual din mai multe motive. În primul rând, fascinația mea pentru limbajul de programare, dart, și programarea mobilă a fost un factor decisiv. Îmi place să explorez tehnologiile noi și să văd cum pot fi aplicate în mod practic pentru a rezolva probleme reale și a îmbunătăți experiența utilizatorilor.

Observând tendințele actuale în educație și tehnologie, am remarcat o discrepanță în ceea ce privește accesul la platforme de campus virtual. Deși universitățile au adoptat în mare măsură aceste sisteme pentru a gestiona cursurile și comunica cu studenții, nivelul de învățământ inferior, precum gimnazii sau licee, pare să fie în urmă în acest sens. Acest lucru a deschis o oportunitate evidentă de a aduce beneficiile tehnologiei moderne și în aceste instituții, facilitând astfel procesul de învățare și colaborare între elevi și profesori.

Prin dezvoltarea acestei aplicații, îmi propun să aduc un instrument care să fie ușor de utilizat, eficient și adaptat specific nevoilor educaționale ale nivelului de învățământ inferior. Înțelegând particularitățile acestui segment, pot proiecta funcționalități care să sprijine procesul educativ, să încurajeze interacțiunea și să faciliteze accesul la resursele necesare pentru o educație de calitate.

De asemenea, prin aducerea tehnologiei în aceste instituții mai mici, îmi doresc să contribui la reducerea decalajului digital dintre diferite niveluri de educație. Astfel, toți elevii, indiferent de instituția la care studiază, ar trebui să aibă acces la aceleași resurse și oportunități de învățare.

În concluzie, alegerea mea de a dezvolta o aplicație de campus virtual pentru nivelul de învățământ inferior nu este doar o pasiune pentru programare și tehnologie, ci și o viziune de a aduce inovația în educație și de a contribui la formarea unei generații care să fie pregătită pentru lumea digitală în care trăim.

2 APLICAȚII ASEMĂNĂTOARE

2.1 Campus Virtual UPT

Această aplicație este prima sursă de inspirație datorită ușurinței de utilizare și din dorința de a extinde această resursă tehnologică și la nivelul instituțiilor de învățământ inferior.

Campusul Virtual al UPT este o platformă educațională online dedicată susținerii academice pentru toate facultățile Universității Politehnica Timișoara (UPT) și pentru învățământul la distanță. Această platformă oferă un mediu centralizat unde studenții și profesorii pot accesa materialele de curs, pot participa la discuții, pot încărca și descărca documente și pot colabora în proiecte și asigurații. Cu o interfață prietenoasă și funcționalități intuitive, Campusul Virtual al UPT facilitează procesul de învățare, oferind acces rapid și ușor la resursele educaționale necesare pentru un parcurs academic de succes.

Figura 2.1.1 Pagina de autentificare pentru Campus Virtual UPT[4]

În Figura 2.1.1. se poate observa interfața aplicației Campus Virtual UPT, unde utilizatorul își poate introduce adresa de email, primită de la facultate și parola. Acest

proces de autentificare este unul simplu, ușor de înțeles de către toate tipurile de utilizatori.

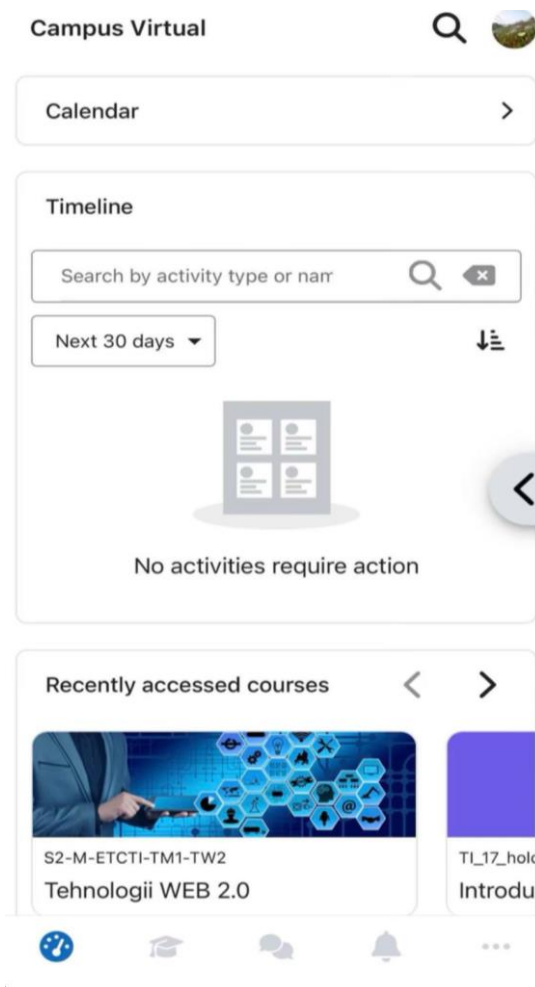


Figura 2.1.2 Pagina principală a aplicației[4]

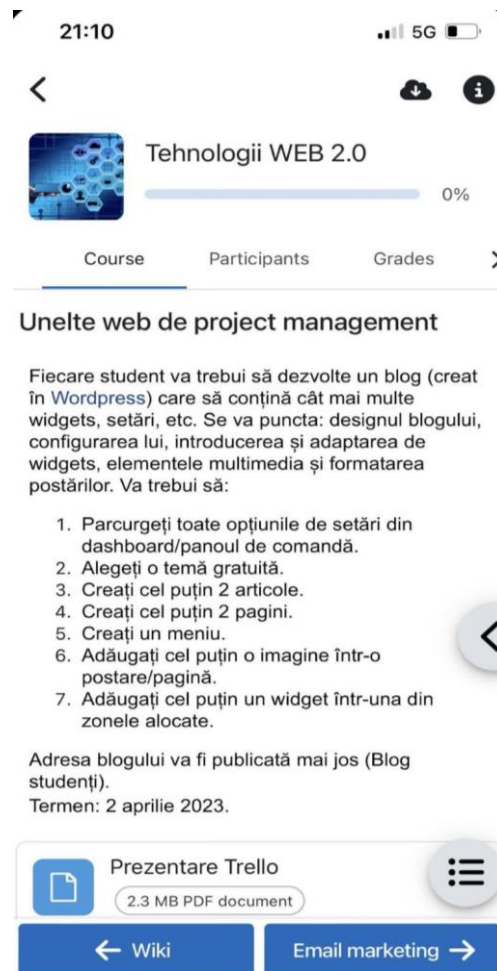


Figura 2.1.3 Pagina unui curs din aplicației[4]

Aplicația este structurată pentru a oferi o experiență intuitivă și eficientă utilizatorilor. În partea de jos, dispune de o bară de meniu care permite acces rapid la diferite secțiuni și funcționalități ale platformei, precum cursuri, resurse, chat și altele. În partea de sus, utilizatorii pot vedea o poză de profil și au la dispoziție o opțiune de căutare inteligentă, care ajută la găsirea rapidă a cursurilor, materialelor sau informațiilor de care au nevoie. Aceste caracteristici contribuie la crearea unei experiențe personalizate și la optimizarea utilizării aplicației în contextul educațional.

2.2 Student UPT

Aplicația "Student UPT" folosește aceeași metodă de autentificare ca și campusul virtual, pentru a asigura coerența și securitatea datelor studenților. Cu toate acestea, spre deosebire de campusul virtual, aplicația "Student UPT" vine cu un set de funcționalități specifice nevoilor administrative și financiare ale studenților.

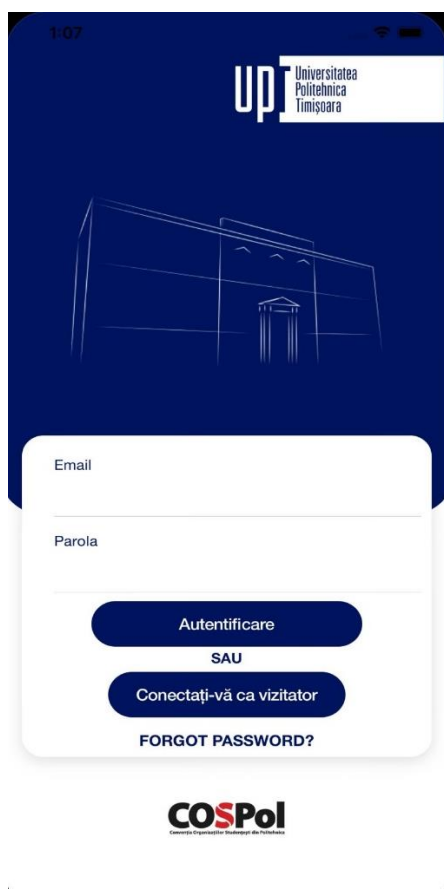


Figura 2.2.1 Pagina autentificare pentru Student UPT[5]

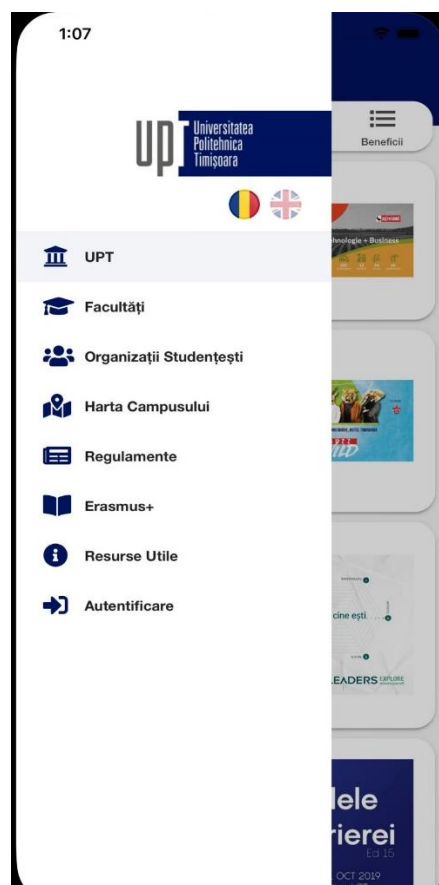


Figura 2.2.2 Pagina principală pentru Student UPT[5]

După cum se poate remarca în Figura 2.1.4 și Figura 2.1.5 studenții pot accesa rapid și ușor informații legate de situația lor școlară, inclusiv note, absențe și orar. De asemenea, aplicația oferă o secțiune dedicată plăților, unde studenții pot verifica taxele datorate, pot efectua plăți online sau pot verifica istoricul plăților. În plus, aplicația include și o secțiune pentru contractele de studii, unde studenții pot vizualiza și semna contractele, pot verifica stadiul documentelor sau pot comunica cu departamentul administrativ pentru eventuale clarificări sau modificări.

2.3 eLearning UVT

Aplicația "eLearning UVT", similară celor menționate anterior, furnizează acces mobil la platforma de învățare online oficială a Universității de Vest din Timișoara.. Această aplicație servește ca un canal de comunicare între studenți și cadrele didactice, consolidând relația dintre ei. Ea se constituie ca un instrument suplimentar în cadrul tehnologiei e-learning deja implementate la nivel universitar, oferind flexibilitate și accesibilitate în procesul de predare, învățare și evaluare.

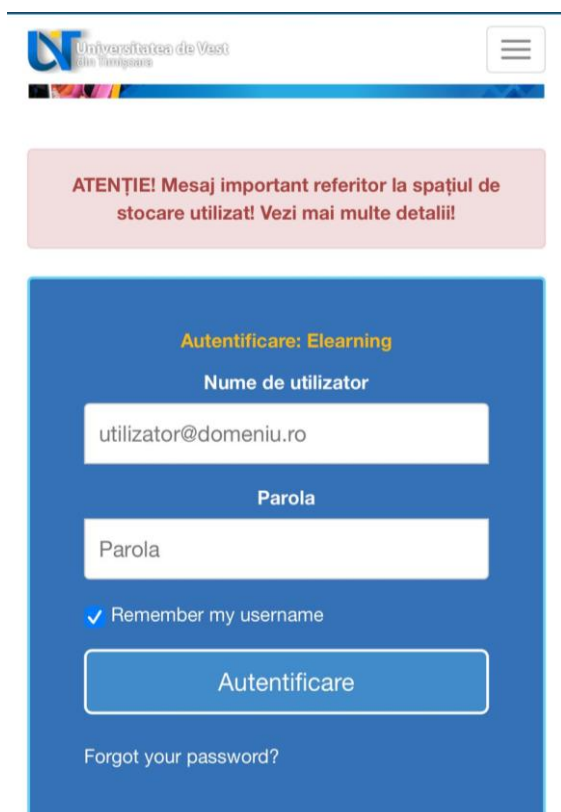


Figura 2.3.1 Pagina autentificare pentru eLearning UVT[6]

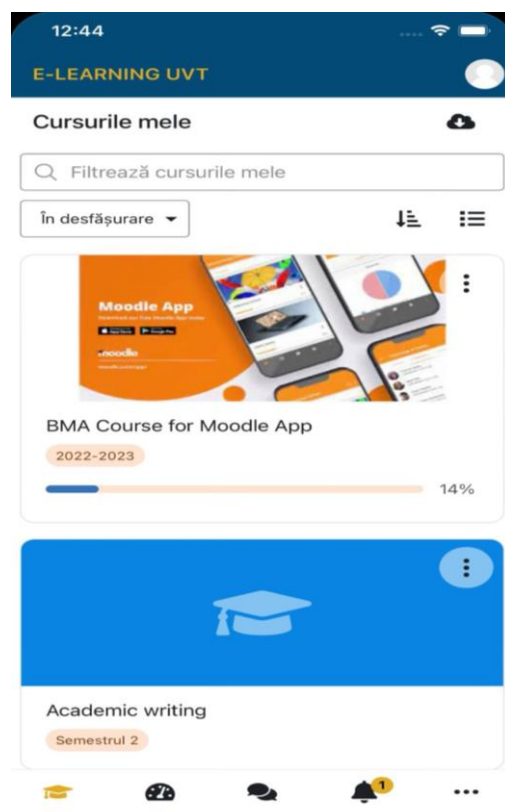


Figura 2.3.2 Pagina principală pentru Student UPT[7]

2.4 UniCampus

UniCampus, o inițiativă a Centrului de eLearning (CeL) al Universității Politehnica Timișoara, are ca obiectiv consolidarea recunoașterii universităților românești și extinderea influenței lor în viața socială și educațională din România prin promovarea unei educații academice de calitate și accesibile. Scopul acestui proiect este să dezvolte primul curs MOOC (Massive Open Online Course) din România, oferind o platformă virtuală online pentru cursuri gratuite și deschise tuturor.

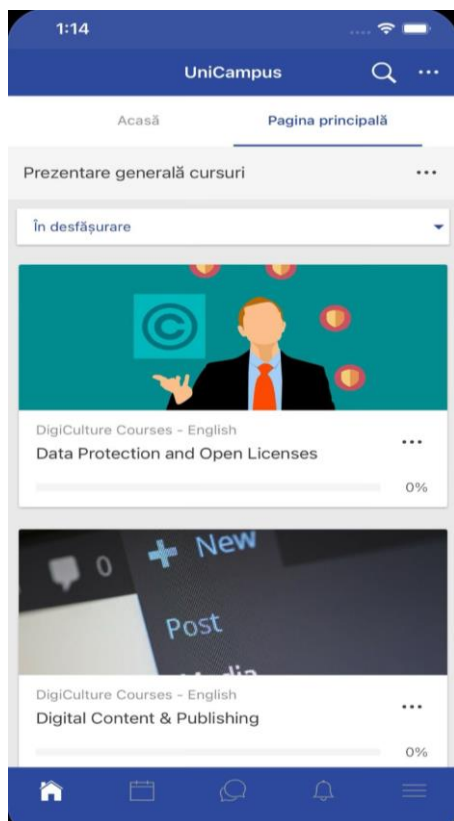


Figura 2.4.1 Pagina principală pentru UniCampus[7]

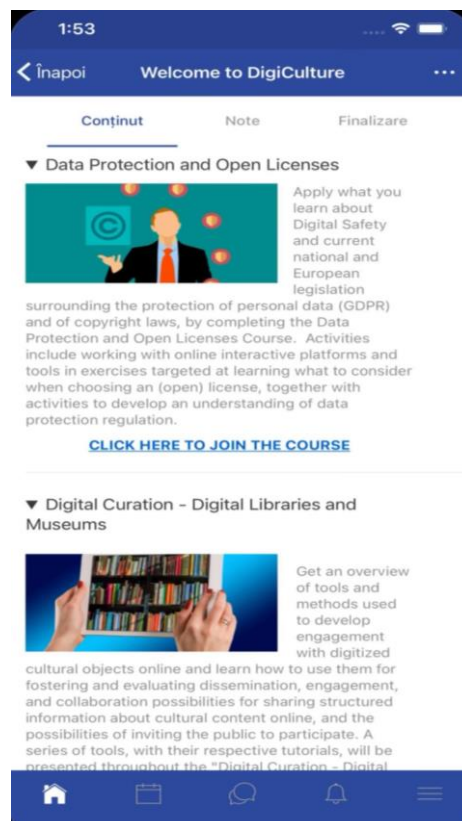


Figura 2.4.2 Pagina unui curs din aplicației[7]

2.5 Comparație aplicații

Autentificare prin email și parolă, bară de meniu pentru navigare rapidă, plată online și citirea codului QR sunt funcționalități comune în aplicațiile de campus virtual. Acestea optimizează accesul la resurse și servicii, facilitând gestionarea studiilor și comunicarea în mediul academic.

Funcție	Aplicație	Campus Virtual	Student UPT	eLearning UVT	UniCampus
Autentificare cu email și parolă		DA	DA	DA	DA
Bară de meniu		DA	DA	DA	DA
Modalitate online de plată		NU	DA	NU	NU
Citire cod QR		NU	DA	NU	NU

Tabel 2.5.1 Tabel de comparații

Toate aplicațiile menționate, fie că este vorba de "Campusul Virtual al UPT", "aplicatia student upt" sau "eLearning UVT", au la bază o structură similară în ceea ce privește autentificarea utilizatorilor și interfața generală. Acestea pun accent pe accesibilitatea și flexibilitatea învățării, oferind utilizatorilor posibilitatea de a accesa resursele educaționale direct de pe dispozitivele mobile. Ele au fost dezvoltate cu gândul la a sprijini comunitatea academică, facilitând comunicarea între studenți și cadrele didactice și consolidând procesul de învățământ prin tehnologie e-learning.

Pe lângă asemănările de bază, fiecare aplicație vine cu un set distinct de funcționalități care răspund nevoilor specifice ale comunităților universitare cărora li se adresează. Indiferent de diferențele de design și funcționalități, scopul lor comun este acela de a îmbunătăți experiența educațională și de a promova o învățare modernă și adaptată nevoilor actuale.

3 Fundamente Teoretice

3.1 Tehnologii folosite

3.1.1 Visual Studio Code

Această aplicație a fost dezvoltată utilizând Visual Studio Code (VS Code), un editor de cod sursă creat de Microsoft. Acest editor este gratuit și compatibil cu diverse platforme, inclusiv Windows, macOS și Linux. VS Code este recunoscut ca unul dintre cele mai populare și apreciate editoare de cod datorită gamei sale extinse de funcționalități.

VS Code oferă o interfață utilizator intuitivă care simplifică navigarea în editor. Utilizatorii pot adăuga extensii pentru a îmbogăți funcționalitățile editorului. De asemenea, este echipat cu diverse instrumente de dezvoltare, inclusiv un debugger, care ajută la eficientizarea procesului de codare. Interfața este personalizabilă, oferind o varietate de teme care contribuie la o vizualizare mai clară și mai plăcută a codului.

Acest editor dispune de funcționalitatea de autocompletare inteligentă, un instrument extrem de util pentru scrierea rapidă și eficientă a codului. De asemenea, suportă depanarea și analiza codului, facilitând identificarea și rezolvarea erorilor. Editorul include și un terminal integrat, permițând utilizatorilor să execute comenzi direct în cadrul editorului, fără necesitatea de a deschide o fereastră de terminal separată.

Interfața utilizator a VS Code este simplă și personalizabilă, oferind acces rapid la barele de instrumente. Editorul este dotat cu un panou lateral care ușurează navigarea prin fișiere, accesul la codul sursă și gestionarea extensiilor instalate. Include, de asemenea, funcționalități avansate precum numerotarea liniilor, extrem de utilă pentru localizarea erorilor, evidențierea sintaxei și indentarea automată, care contribuie semnificativ la claritatea și lizibilitatea codului.

VS Code oferă o gamă vastă de extensii care pot fi instalate pentru a îmbunătăți și eficientiza fluxul de lucru, facilitând lucrul eficient cu diverse limbaje de programare și tehnologii. Printre cele mai populare extensii se numără IntelliSense pentru autocompletare, formatarea automată a codului, linter pentru verificarea codului, debuggere integrate pentru diferite limbaje, precum și integrări cu platforme de cloud. Aceste extensii sunt disponibile în magazinul oficial de extensii al VS Code și pot fi instalate cu ușurință direct din editor.

VS Code include suport nativ pentru sistemele de control al versiunilor, cu un accent deosebit pe Git. Utilizatorii pot accesa comenzi și unelte specifice pentru gestionarea versiunilor. Extensii cum ar fi GitLens îmbogățesc acest suport, oferind detalii suplimentare despre istoricul modificărilor codului și despre contribuțiile diferiților dezvoltatori.

În procesul de depanare, VS Code vine echipat cu suport integrat pentru diverse limbaje de programare. Utilizatorii pot configura breakpoint-uri, locuri specificate unde execuția codului se oprește temporar. Aceasta permite setarea mai multor puncte de oprire pentru a monitoriza variabilele și pentru a executa codul pas cu pas, facilitând identificarea și rezolvarea problemelor. [8]

VS Code se integrează eficient cu diverse servicii de cloud, incluzând Azure, AWS și Google Cloud. Utilizatorii pot beneficia de extensii specifice pentru cloud, care permit accesul la resursele cloud, administrarea mașinilor virtuale, gestionarea bazelor de date și multe alte funcționalități. De asemenea, serviciul Live Share este extrem de util pentru

colaborarea în timp real. Acesta permite dezvoltatorilor să partajeze și să lucreze împreună la același cod simultan, ceea ce poate accelera dezvoltarea proiectelor și îmbunătăți colaborarea în echipă.

Acest editor de cod permite personalizarea temelor, culorilor, fonturilor și multor altor elemente, astfel încât fiecare utilizator să-și poată adapta editorul la propriul stil preferat, optimizând experiența de dezvoltare. De asemenea, comunitatea este foarte activă și contribuie constant la introducerea de noi funcționalități, pentru a satisface diversele nevoi ale dezvoltatorilor. [9]

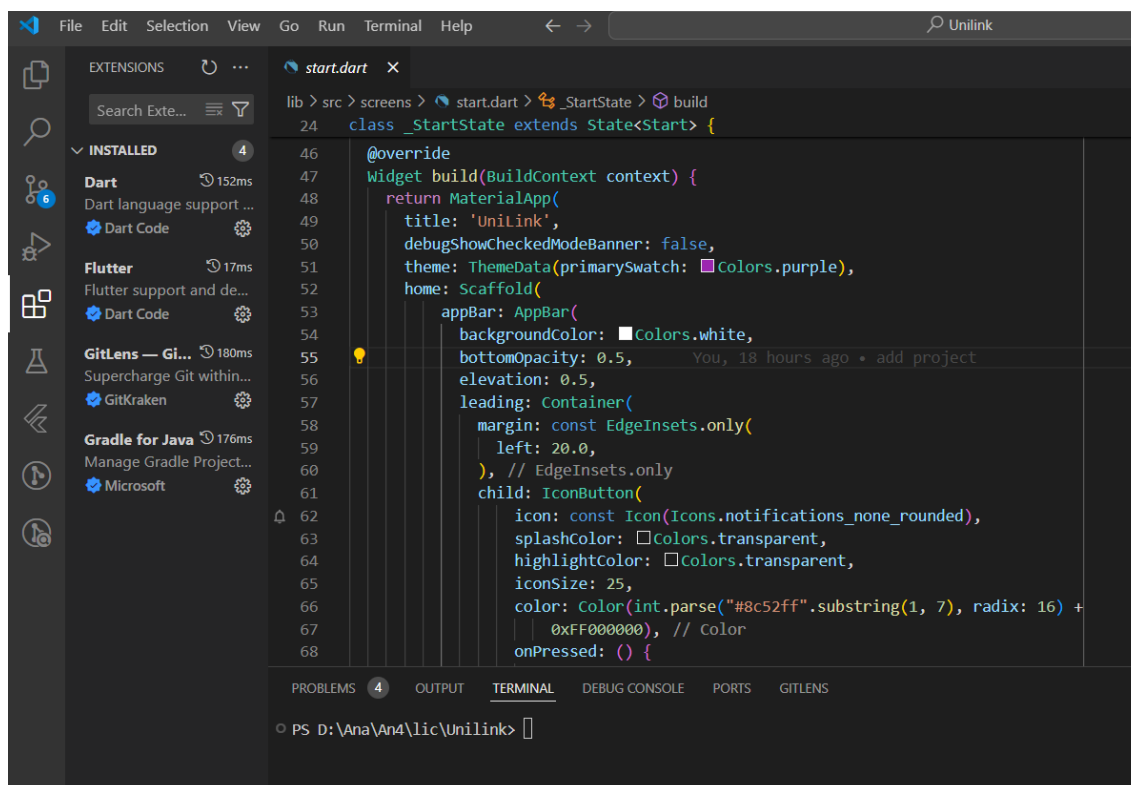


Figura. 3.1.1. Interfața Visual Studio Code, extensiile și terminalul

Figura 3.1.1 ilustrează interfața Visual Studio Code și extensiile sale, împreună cu fereastra terminalului.

3.1.2 Flutter

Flutter este un kit de dezvoltare software pentru interfețe utilizator, deschis și creat de Google. Acesta permite dezvoltatorilor să creeze aplicații multi-platformă dintr-o singură bază de cod, funcționale pe web, Fuchsia, Android, iOS, Linux, macOS și Windows. Prezentat inițial în 2015 și lansat în mai 2017, Flutter este utilizat atât intern de Google în aplicații precum Google Pay și Google Earth, cât și de alte companii mari precum ByteDance și Alibaba. Flutter combină un limbaj UI cu un motor de randare propriu. Când o aplicație Flutter este compilată, aceasta include atât codul UI, cât și motorul de randare, totalizând aproximativ 4 MB. Acesta este un contrast față de alte cadre UI care se bazează pe motoare de randare separate și livrează doar codul UI, cum ar fi aplicațiile native Android sau aplicațiile web HTML/JavaScript.

Controlul complet al Flutter asupra procesului său de randare simplifică suportul pentru multiple platforme, necesitând doar ca platforma să permită rularea codului nativ. Inițial cunoscut sub numele de "Sky" și destinat sistemului de operare Android, Flutter a fost prezentat în 2015 cu obiectivul de a atinge 120 de cadre pe secundă.

Componenta fundamentală a unei aplicații Flutter este "widgetul", care poate conține alte widgeturi. Flutter se distinge de alte soluții multi-platformă, precum React Native și Xamarin, prin faptul că redă widgeturile direct pe pixeli, fără a utiliza componente native ale platformei. Flutter include widgeturi fără stare și widgeturi cu stare. Widgeturile fără stare se actualizează doar când intrările lor se schimbă, în timp ce widgeturile cu stare pot actualiza o stare internă și se pot redesena folosind metoda `setState()`.

Aplicațiile Flutter sunt scrise în limbajul Dart. Pentru web, codul este transformat în JavaScript sau WebAssembly, în timp ce pentru alte platforme se folosește compilarea anticipată. Flutter include și un manager de pachete Dart Pub, permitând publicarea și utilizarea pachetelor și pluginurilor personalizate.

Flutter suportă două back-end-uri pentru motorul său de randare: biblioteca de grafică Skia de la Google și motorul Flutter numit Impeller, activat implicit pe iOS și în beta pe Android. În plus, include biblioteca Foundation scrisă în Dart, care oferă funcții esențiale pentru construirea aplicațiilor.

Cadru de dezvoltare Flutter oferă două seturi de widgeturi care respectă limbaje de design specifice: widgeturile Material Design, pentru limbajul de design Google, și widgeturile Cupertino, pentru ghidurile Apple de interfață umană, permițând dezvoltatorilor să utilizeze orice set de widgeturi pe oricare platformă. Pachetele terțe pot ajusta automat designul aplicației la sistemul de operare utilizat.[10]

Flutter SDK reprezintă o platformă esențială pentru dezvoltarea de aplicații mobile, web și desktop, oferind un cadru de lucru extrem de versatil și eficient. Avantajul principal al Flutter este capacitatea sa de a compila codul direct în cod nativ pentru platformele vizate, asigurând performanțe ridicate și o experiență fluidă pentru utilizatori.

Arhitectura bazată pe widget-uri este nucleul Flutter. Acesta tratează totul ca un widget - de la elemente simple ale interfeței, cum ar fi textul și butoanele, până la structurile

complexe de layout. Această abordare modulară facilitează crearea de interfețe personalizabile și reutilizabile.

Motorul de randare al Flutter, scris în C++, desenează widget-urile direct pe canvas, permițând Flutter să opereze independent de platformele native și să controleze precis fiecare pixel afișat, fără limitările impuse de sistemele de randare specifice fiecărei platforme.

Hot Reload este una dintre funcțiile cel mai mult apreciate în Flutter, permițând dezvoltatorilor să vadă modificările în cod în timp real, fără a pierde starea aplicației. Aceasta caracteristică accelerează dezvoltarea și îmbunătățește productivitatea.

Suportul multi-platformă este un alt punct forte al Flutter, care permite dezvoltarea aplicațiilor pentru Android, iOS, Windows, macOS, Linux și web folosind aceeași bază de cod, reducând astfel efortul și costurile asociate cu menținerea aplicațiilor pe multiple platforme.

Performanța apropiată de cea nativă, obținută prin compilarea în cod nativ și optimizările de performanță, asigură că aplicațiile Flutter funcționează la viteze comparabile cu cele native, evitând problemele de performanță ale altor framework-uri cross-platform care utilizează bridge-uri JavaScript.

Suportul extensiv pentru animații și tranziții este inclus în SDK, oferind unelte puternice pentru crearea de animații complexe și tranziții suave, esențiale pentru o experiență de utilizator captivantă și imersivă.

Dezvoltarea comunității și ecosistemul de plugin-uri sunt vitale pentru Flutter, cu o comunitate activă de dezvoltatori și un ecosistem vast de plugin-uri care extind funcționalitățile framework-ului, de la integrarea cu alte platforme și servicii, la funcționalități avansate de UI și UX.

În concluzie, Flutter SDK este un instrument valoros pentru dezvoltatorii care aspiră să creeze aplicații moderne, eficiente și estetic atrăgătoare, cu un suport extins pentru multiple platforme, toate acestea cu un efort relativ redus.[11]. În Figura 3.1.3 este prezentată pictograma caracteristică Flutter.



Figura 3.1.2 Pictograma Flutter

3.1.3 Dart

Dart este un limbaj de programare dezvoltat de Lars Bak și Kasper Lund și introdus de Google. Este un limbaj orientat obiect, bazat pe clase, cu gestionare automată a memoriei și o sintaxă în stil C, conceput pentru dezvoltarea de aplicații web, mobile, pentru server și desktop. Dart suportă funcționalități avansate cum ar fi interfețe, mixin-uri, clase abstracte, generice reificate și inferența de tip.

Lansat inițial în octombrie 2011 la conferința GOTO din Aarhus, Danemarca, Dart a primit inițial o primire mixtă, fiind criticat pentru potențiala sa de a fragmenta web-ul. Prima versiune stabilă, Dart 1.0, a fost lansată în noiembrie 2013. În 2015, cu lansarea Dart 1.9, focusul s-a schimbat spre compilarea codului în JavaScript, întrucât planurile de a include o VM Dart în Chrome au fost abandonate. Versiunea Dart 2.0, lansată în august 2018, a adus modificări semnificative limbajului, inclusiv un sistem de tipuri îmbunătățit. Dart 2.6 a introdus dart2native, care permitea compilarea nativă pe platforme desktop, extinzând astfel capabilitățile limbajului.

Mai recent, Dart 3.0, lansat în mai 2023, a consolidat sistemul de tipuri pentru a include siguranța nulului și a adăugat noi funcții precum înregistrările, modelele și modificatorii de clasă. Începând cu Dart 3.22, limbajul a fost capabil să se compileze în WebAssembly, crescând portabilitatea și flexibilitatea sa.[12]

Dart este utilizat în mod esențial pentru dezvoltarea în Flutter, fiind optimizat pentru construirea de interfețe fluide și reactive pe dispozitive mobile. Oferă o sintaxă clară și concisă, combinând caracteristicile limbajelor orientate pe obiect cu instrumente moderne de programare funcțională, ceea ce îl face ideal pentru dezvoltarea rapidă și eficientă a aplicațiilor complexe. Compilatorul avansat al lui Dart oferă două moduri de compilare: „la timp” și „înaintea timpului”. Prima opțiune este crucială în timpul dezvoltării pentru reîncărcarea la cald, care permite modificări instantanee fără a pierde starea aplicației, în timp ce AOT este utilizat pentru producție, generând cod nativ optimizat pentru fiecare platformă. Aceasta asigură că aplicațiile Flutter rulează la viteze comparabile cu cele ale aplicațiilor native. Acesta implementează programarea asincronă cu `async` și `await`, esențială pentru gestionarea eficientă a operațiunilor dependente de timp, cum ar fi solicitările de rețea. [13] Dart este valoros nu doar pentru flexibilitatea în dezvoltare și adaptabilitatea la diverse platforme, dar și pentru productivitatea sporită și suportul extins pentru diagnosticarea problemelor. Astfel, Dart se afirmă ca un limbaj de programare robust, performant și estetic plăcut, ideal pentru o gamă largă de aplicații software. În Figura 3.1.3 este reprezentată pictograma specifică Dart.



Figura 3.1.3 Pictograma Dart

3.1.4 Material Design

Material Design este un limbaj de design vizual dezvoltat de Google, introdus în 2014, cu scopul de a îmbunătăți coerența interfețelor utilizatorilor pe diverse platforme, în special pe dispozitive mobile. Acest set de principii de design se concentrează pe utilizarea spațiului, a tranzițiilor dinamice, a efectelor de adâncime cum ar fi umbrele și iluminarea realistă, și pe o paletă de culori vii pentru a crea interfețe plăcute și intuitive. Acesta este conceput să funcționeze pe o varietate largă de dispozitive și dimensiuni de ecran, de la ceasuri inteligente și telefoane până la tablete și ecrane mari de desktop. Prin utilizarea unui sistem bazat pe unități adaptative și puncte de ancorare flexibile, Material Design asigură o experiență de utilizator fluidă și accesibilă, indiferent de dispozitiv sau context.[14]

În contextul Flutter, Material Design este integrat profund în framework, oferind dezvoltatorilor un set bogat de widget-uri și stiluri care aderă la acest limbaj de design. Acesta include componente precum butoane, carduri, bare de navigație, și dialoguri, toate cu animații predefinite și un comportament tactil optimizat pentru interacțiuni intuitive. Implementarea Material Design în Flutter nu se limitează doar la aspectul estetic; ea încorporează și principii de design pentru accesibilitate, asigurându-se că aplicațiile sunt utilizabile de către o gamă largă de utilizatori, inclusiv cei cu dizabilități. De exemplu, contrastul adecvat al culorilor, dimensiunile accesibile ale elementelor tactabile și etichetele descriptive sunt toate luate în considerare, îmbunătățind accesibilitatea globală a aplicațiilor.

Material Design în Flutter este, de asemenea, extrem de personalizabil, permițând designerilor și dezvoltatorilor să modifice temele pentru a se potrivi branding-ului specific al unei aplicații sau preferințelor estetice. Aceasta include ajustarea schemelor de culori, a tipografiilor și a formelor componentelor UI. Datorită acestui nivel de flexibilitate, aplicațiile pot menține coerența cu Material Design, în timp ce se disting prin aspectul unic.[15]

3.1.5 Firebase

Firebase este un ansamblu de servicii de cloud computing pentru backend și platforme de dezvoltare a aplicațiilor oferite de Google. Oferă hosting pentru baze de date, servicii, autentificare și integrare pentru o varietate de aplicații, inclusiv pentru Android, iOS, JavaScript, Node.js, Java, Unity, PHP și C++. Firebase a evoluat din Envolv, o companie anterioară fondată de James Tamplin și Andrew Lee în 2011. Dezvoltatorii utilizau Envolv pentru a sincroniza datele aplicației, cum ar fi starea jocului, în timp real între utilizatori. Tamplin și Lee au decis să separe sistemul de chat de arhitectura în timp real care îl susținea. Au fondat Firebase ca o companie separată în 2011 și a fost lansată publicului în aprilie 2012.[16]

Firebase, o platformă dezvoltată de Google, este adesea integrată cu Flutter pentru a oferi o suită robustă de servicii backend care îmbunătățesc și simplifică dezvoltarea aplicațiilor mobile. Această combinație permite dezvoltatorilor să creeze soluții rapide și eficiente, reducând semnificativ timpul și efortul necesar pentru administrarea infrastructurii backend.

Autentificarea - Firebase Auth este un serviciu care suportă autentificarea bazată pe email și parolă, autentificare prin rețele sociale precum Facebook, Twitter și Google, și autentificare bazată pe telefon. Integrează-se ușor în aplicațiile Flutter, oferind o soluție sigură și scalabilă pentru gestionarea utilizatorilor.

Stocarea datelor - Firebase oferă două soluții principale de stocare a datelor adaptate nevoilor aplicațiilor moderne: Firestore și Firebase Realtime Database. Firestore este o bază de date NoSQL care permite stocarea și sincronizarea datelor între utilizatori în timp real, ideală pentru aplicații interactive. Firebase Realtime Database oferă funcționalități similare, cu o structură de date ceva mai simplă, facilitând accesul rapid și în timp real la date.

Notificările push - Firebase Cloud Messaging (FCM) este un serviciu gratuit care permite dezvoltatorilor să trimită notificări push pe dispozitivele Android și iOS. În Flutter, acesta poate fi utilizat pentru a îmbunătăți angajamentul utilizatorilor prin trimiterea de mesaje și alerte proactive.

Firebase Analytics oferă o integrare profundă cu aplicațiile Flutter, permițând dezvoltatorilor să colecteze și să analizeze comportamentul utilizatorilor. Aceasta informație este vitală pentru optimizarea experienței utilizatorilor și pentru luarea unor decizii informate bazate pe date.

Capacitatea de a implementa rapid funcționalități complexe fără a fi necesară o infrastructură backend extinsă este unul dintre avantajele utilizării Firebase în proiectele Flutter. Deoarece Firebase gestionează lucruri precum managementul bazei de date, securitatea și scalabilitatea, dezvoltatorii pot concentra atenția pe crearea experienței utilizatorului. SDK-urile bine documentate și suportul extensiv pentru Flutter sunt disponibile pe Firebase, ceea ce face integrarea simplă și directă. Cu puțin cod, dezvoltatorii pot adăuga și configura servicii Firebase în aplicațiile Flutter.

Prin urmare, integrarea Firebase în aplicațiile Flutter este o strategie eficientă pentru a construi aplicații mobile complexe, oferind o gamă largă de servicii backend care pot îmbunătăți semnificativ funcționalitatea, securitatea și experiența utilizatorului final.[17]

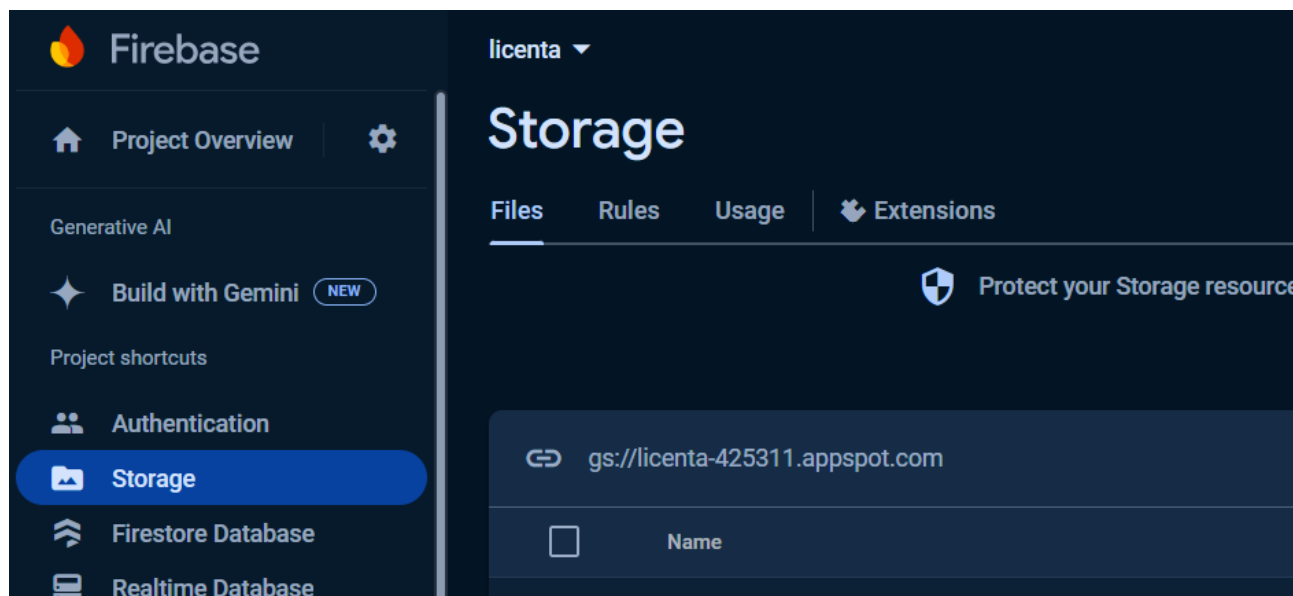


Figura 3.1.5. Firebase Storage

În Figura 3.1.5. se pot observa funcționalitățile Firebase pe care le-am implementat în cadrul aplicației mele.

3.1.6 API-uri și SDK-uri native Android

Flutter este un set de instrumente de dezvoltare a aplicațiilor extrem de versatil, care nu numai că vă permite să creați UI-uri complexe, de înaltă performanță, dar permite și integrarea profundă cu platforma de bază, în special Android, folosind API-urile și SDK-urile native ale acestui sistem de operare. Această caracteristică este importantă pentru accesarea anumitor caracteristici ale dispozitivului care nu sunt expuse direct prin Flutter.

Prin canalul platformei, Flutter permite dezvoltatorilor să ruleze cod nativ pe platformele Android și iOS. Acest mecanism de comunicare între codul Dart și platformele native este esențial pentru a utiliza caracteristici care nu sunt disponibile în biblioteca standard Flutter, cum ar fi accesul la senzori specifici, utilizarea caracteristicilor hardware avansate sau integrarea cu API-uri de sistem mai complexe.

Integrarea API-urilor native Android într-o aplicație Flutter se realizează prin definirea unui canal de comunicare prin care se trimit și se primesc mesaje între codul Dart și Java/Kotlin. Acesta permite aplicațiilor Flutter să execute funcții native cum ar fi: accesarea și gestionarea datelor de senzori, manipularea rețelei sau utilizarea Bluetooth,

interacțiunea cu serviciile de localizare, utilizarea camerelor dispozitivului pentru captura de imagini sau video.[18]

3.1.7 Android Studio

Android Studio, mediul de dezvoltare integrat (IDE) oficial pentru Android de la Google, este construit pe baza software-ului IntelliJ IDEA de la JetBrains. Acesta este destinat în mod specific dezvoltării de aplicații Android și poate fi descărcat pentru sistemele de operare Windows, macOS și Linux. Android Studio a venit ca înlocuitor al Eclipse Android Development Tools și este principala alegere pentru dezvoltarea de aplicații Android native. Cu toate că este licențiat sub licența Apache, include actualizări SDK sub licențe non-free, ceea ce înseamnă că nu este complet open source.

Android Studio oferă o interfață intuitivă împreună cu o serie de instrumente utile, cum ar fi un editor de cod inteligent, unelte pentru analiza performanței, emulatoare, și altele. Aceste caracteristici îl fac ideal pentru depanarea și dezvoltarea aplicațiilor, fiind foarte accesibil pentru programatorii începători. De asemenea, este compatibil cu toate platformele Android și integrează suport nativ pentru facilități cum ar fi Google Cloud Messaging și Google App Engine. Bazându-se pe IntelliJ IDEA, beneficiază de un editor de cod avansat care contribuie la confortul utilizatorilor. Include, de asemenea, o funcție de rulare instantanee care accelerează procesele de editare, compilare și executare.

Android Studio utilizează Gradle ca sistem de bază pentru compilare, cu funcții suplimentare specifice Android oferite de pluginul Android Gradle. Acest sistem de compilare permite utilizatorilor să personalizeze, configureze și extindă procesul de construire, creând multiple formate de distribuție și instalare pentru aceeași aplicație cu funcții diferite, folosind același proiect și module, și reutilizând codul și resursele.

Fișierele de configurare pentru asamblare în Android Studio sunt denumite `build.gradle.kts` atunci când se utilizează Kotlin sau `build.gradle` pentru Groovy. Acestea sunt fișiere text simple ce folosesc sintaxa Kotlin sau Groovy pentru a configura construcția cu elemente oferite de pluginul Android Gradle. Fiecare proiect are un fișier de configurare de nivel superior pentru întregul proiect și fișiere de configurare la nivel de modul pentru fiecare modul individual. La importarea unui proiect existent, Android Studio generează automat fișierele de configurare necesare.[18]

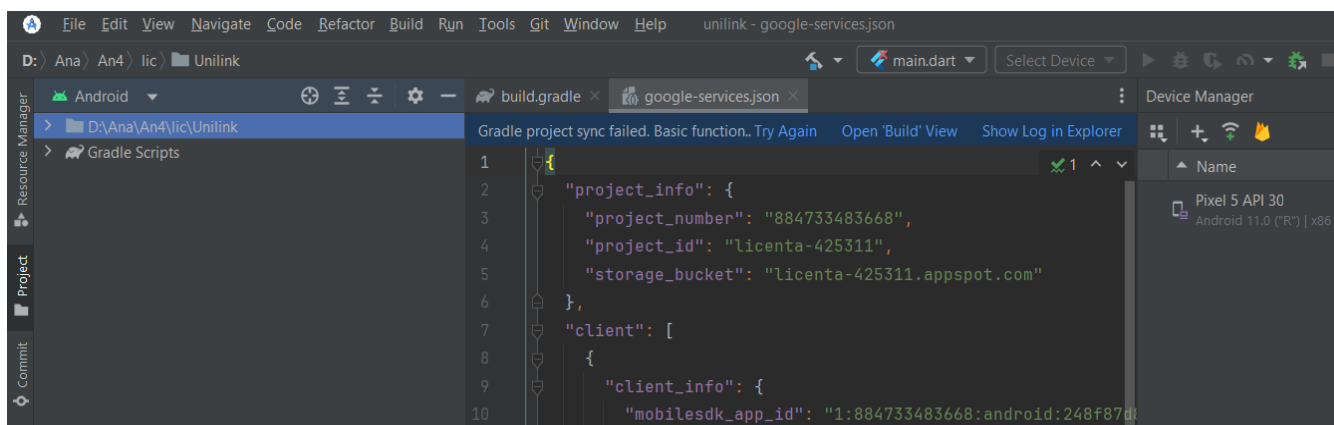


Figura 3.1.7 Interfața Android Studio

3.1.8 Emulator Android

Emulatorul Android este un instrument vital în procesul de dezvoltare a aplicațiilor pentru Android, oferind dezvoltatorilor posibilitatea de a simula o gamă diversă de dispozitive Android direct de pe propriul computer. Această simulare acoperă diferite configurații de hardware și versiuni ale sistemului de operare, facilitând testarea aplicațiilor într-o varietate de condiții ambientale fără a necesita accesul la multiple dispozitive fizice. Acesta dispune de o varietate largă de configurații pentru diferite dispozitive, cum ar fi telefoane, tablete, dispozitive portabile. Această diversitate permite dezvoltatorilor să verifice comportamentul aplicațiilor într-un spectru larg de situații, asigurând astfel o compatibilitate extinsă.

Emulatorul reproduce cu fidelitate funcționalitățile unui dispozitiv real, permitând testarea profundă a aplicațiilor. De la simularea apelurilor telefonice și mesajelor text, la specificarea locației GPS, simularea vitezelor de rețea, rotația dispozitivului și senzorii hardware - toate sunt posibile și oferă o perspectivă precisă asupra comportamentului aplicației în condiții reale.

Viteza ciclului de dezvoltare este îmbunătățită semnificativ, oferind un mediu eficient pentru testare și depanare rapidă. Transferul de date prin emulator se realizează adesea mai repede decât prin intermediul unui dispozitiv fizic conectat prin USB, permițând implementarea și verificarea modificărilor într-un mod aproape instant.[19]

Procesul de utilizare a emulatorului începe cu instalarea Android Studio, care include un set de unelte de dezvoltare și o bibliotecă de imagini de sistem pentru emulator. După instalare, dezvoltatorii pot configura o instanță de emulator pentru a replica dispozitivul țintă.

Interfața unui emulator Android, Pixel 5 vizibilă în Figura 3.1.8.

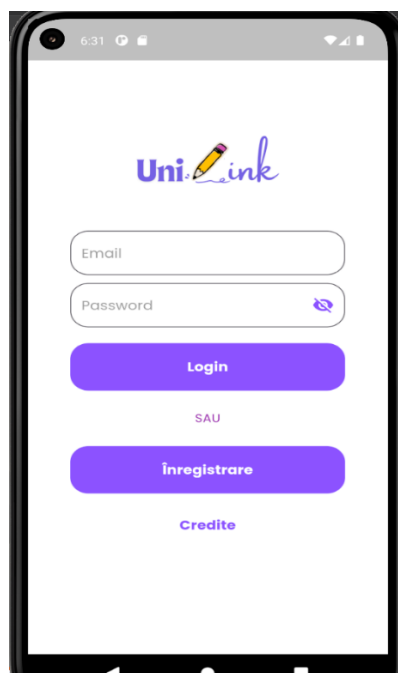


Figura 3.1.8 Interfața emulator Android

4 Proiectarea și Implementarea

Capitolul următor va oferi o analiză amănunțită a sistemului, abordând soluțiile implementate, structura arhitecturală și componentele utilizate. Se vor explora detaliat modulele care compun proiectul, descriindu-se fiecare modul în parte.

4.1 Arhitectura

Această aplicație are un sistem bazat pe un model client-server care beneficiază de funcționalitățile Firebase pentru a asigura utilizarea sistemului la nivel global, utilizatorii având acces la toate informațiile necesare. Această implementare este folosită pentru a facilita utilizarea aplicației.

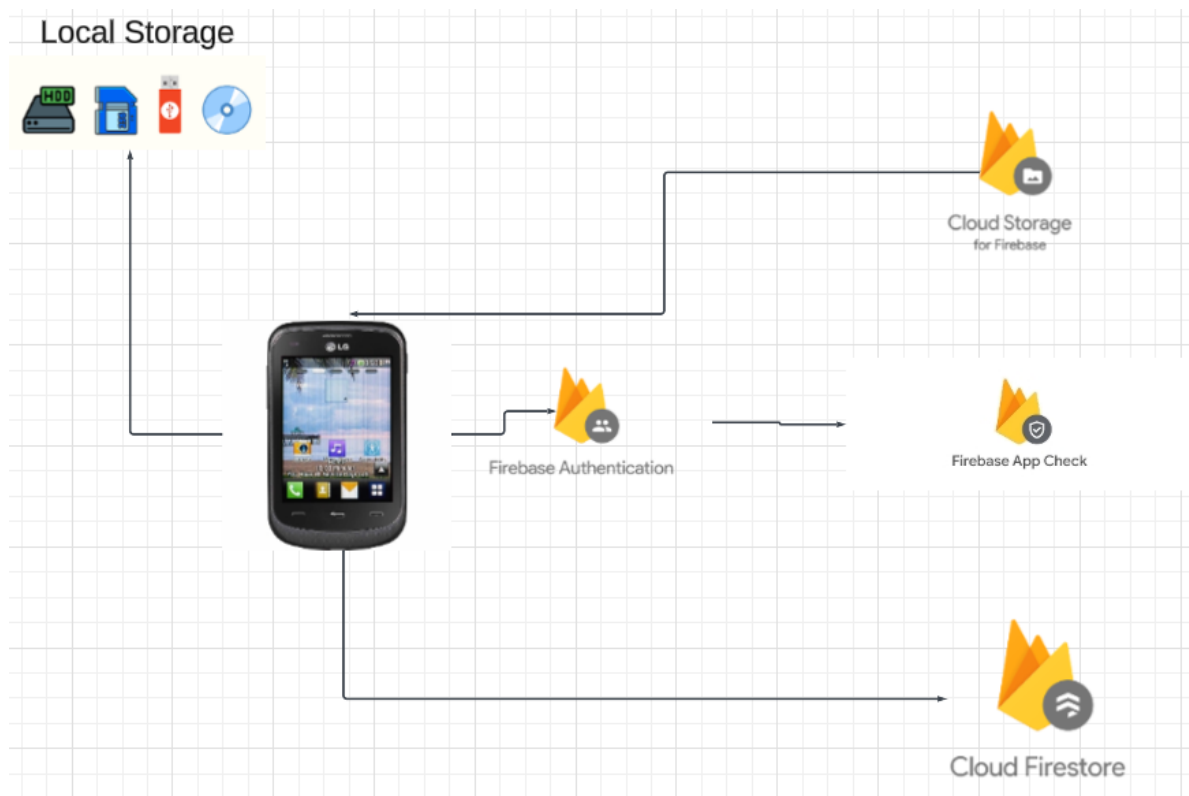


Figura 4.1.1 Structura conceptuală în detaliu

În Figura 4.1.1 este reprezentată arhitectura conceptuală a aplicației, alcătuită cu funcționalitățile Firebase.

Firebase Authentication, acest API este utilizat pentru a incorpora în aplicație funcționalități de autentificare. Permite utilizatorilor să se înregistreze, să se autentifice, să își recupereze parolele și să rămână conectați, astfel încât să nu fie necesară reautentificarea la fiecare deschidere a aplicației.

Firebase App Check oferă un strat suplimentar de securitate prin validarea cererilor către serviciile Firebase. Aceasta asigură că numai cererile provenite de la aplicația oficială de campus virtual sunt acceptate, protejând astfel datele și resursele educaționale de accesul neautorizat sau abuziv.

Firebase Cloud Firestore, este folosit în aplicațiile pentru a stoca și sincroniza datele utilizatorilor în timp real, permițând accesul eficient și actualizat la informații educaționale, chiar și în modul offline. Aceasta asigură o experiență de învățare conectată și accesibilă pentru studenți și cadre didactice, indiferent de locație.

Firebase Cloud Storage este implementat pentru a gestiona și a distribui eficient fișiere mari, inclusiv videoclipuri educative, documente PDF și alte materiale didactice. Această soluție asigură acces sigur și rapid la conținutul educațional pentru utilizatori, oriunde s-ar afla.

Local Storage este esențială pentru îmbunătățirea experienței utilizatorului prin stocarea datelor pe dispozitivul local. Acest mecanism permite aplicației să salveze și să recupereze date direct din browserul utilizatorului sau din memoria internă a dispozitivului, fără a necesita conexiune constantă la internet.

Componenta hardware implicată în rularea aplicației de campus virtual este telefonul mobil al utilizatorului. Telefonul servește ca dispozitiv client pe care se execută aplicația, gestionând interfața utilizatorului și procesând datele locale. Performanța și claritatea aplicației pot fi influențate de puterea de procesare și memoria telefonului. De asemenea, este esențial ca aplicația să fie optimizată pentru a funcționa pe diferite versiuni de sisteme de operare, pentru a oferi o experiență utilizator uniformă pe diferite dispozitive.

În concluzie, această aplicație de campus virtual utilizează arhitectura client-server, care separă funcționalitățile între servere, ce procesează cereri și gestionează date, și clienți, cum ar fi aplicații mobile sau browsere web, care afișează informații utilizatorilor. Aceasta structură dispune de actualizări centralizate, securitate îmbunătățită și o scalabilitate eficientă, deși este necesară o conexiune permanentă la internet pentru acces total la funcționalități.

4.2 Arhitectura bazei de date

Baza de date a aplicației de campus virtual este esențială pentru organizarea și gestionarea eficientă a informațiilor. Structură tipică a bazei de date pentru această aplicație, presupune prin folosirea a mai multor tabele pentru a separa și a organiza datele relevante, astfel structura acestora se poate observa în Figura 4.2.1

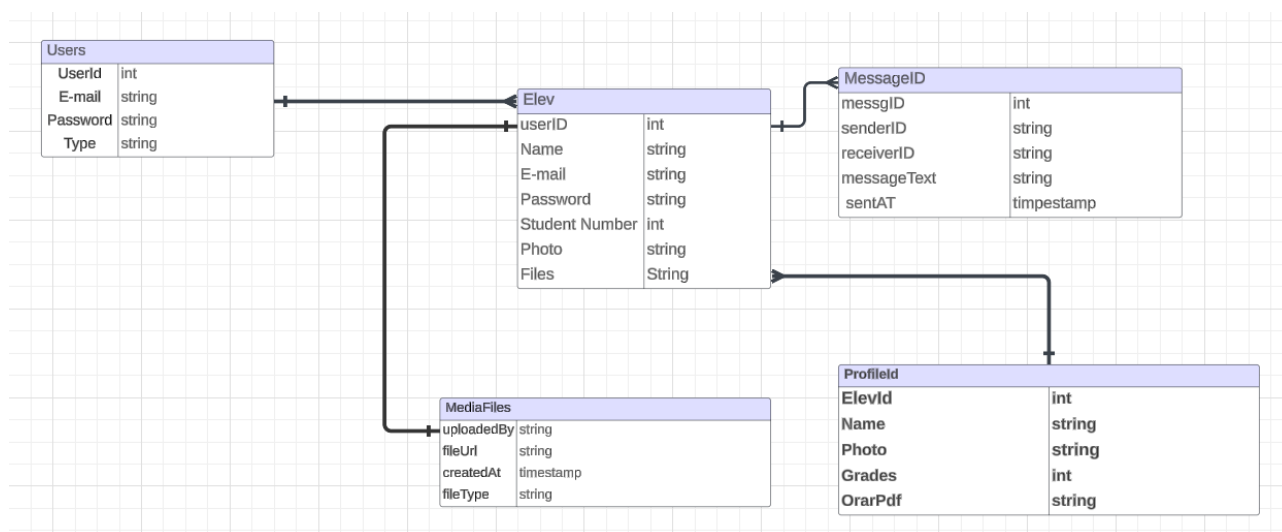


Figura 4.2.1 Structura bazei de date

Tabela *Users* este structurată cu câmpuri esențiale pentru gestionarea accesului în aplicație, incluzând identificatori unici, nume, adrese de email, parole criptate, roluri diferite (cum ar fi student, profesor, administrator), și data înregistrării. Această structură permite autentificarea securizată, personalizarea interacțiunilor și monitorizarea activității utilizatorilor.

Tabela *Elevi* în cadrul bazei de date a acestei aplicații este esențial pentru a oferi o structură detaliată și specifică pentru gestionarea informațiilor elevilor. Extinderea informațiilor din tabelul *Users* cu detalii suplimentare în acest tabel permite aplicației să ofere funcționalități personalizate și eficiente pentru elevi.

Tabela *MediaFiles* joacă un rol crucial în gestionarea și organizarea fișierelor media încărcate în sistem, care sunt folosite ca materiale didactice sau alte resurse educaționale. Aceasta asigură stocarea materialelor didactice, și conține toate fișierele media la care are acces utilizatorul.

Tabela *ProfileID* are o structură simplă, care stochează datele de interes ale utilizatorului conectat. Interacțiunea cu tabela *Elev* se realizează prin câmpul *elevID*. Prin aceasta se poate observa o imagine detaliată a profilului academic al fiecărui elev.

Tabela *MessageID* este dedicată gestionării mesajelor schimbate între utilizatori, facilitând astfel comunicarea eficientă în cadrul platformei. Aceasta are un rol esențial în menținerea conexiunii și colaborării între elevi, profesori și personalul administrativ.

4.3 Interfața aplicației

În secțiunea următoare, vom prezenta interfața aplicației, detaliind componentele sale vizuale și funcționale pentru a oferi o înțelegere clară a modului în care utilizatorii interacționează cu diferitele caracteristici și servicii disponibile. Vom explora designul paginilor principale, funcționalitățile cheie și modul în care acestea sunt organizate pentru a asigura o experiență de utilizare optimă și intuitivă. În Figura 4.3.1 este reprezentată o diagrama UML a interfeței și conexiunile dintre opțiunile acestei aplicații.

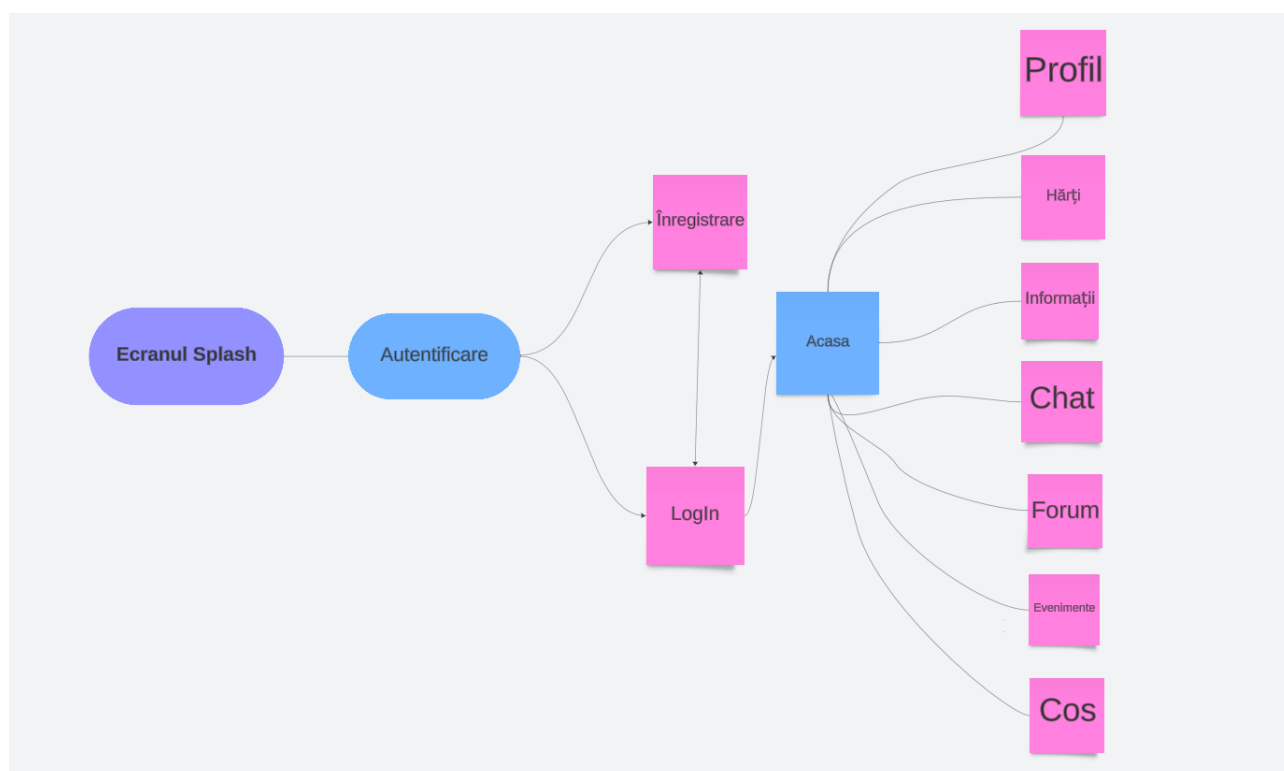


Figura 4.3.1 Diagrama UML a aplicației

Ecranul de start afișează logo-ul aplicației. Acesta servește ca introducere în aplicație și oferă prima interacțiune cu utilizatorul. Ecranul este afișat pentru câteva secunde, perioadă în care aplicația încarcă resursele esențiale necesare pentru funcționarea sa eficientă.

Ecranul de Autentificare și Înregistrare oferă formulare pentru autentificarea și înregistrarea utilizatorilor noi. Designul este simplu și curat, cu câmpuri clare pentru introducerea informațiilor și butoane de acțiune evidențiate. Utilizatorii pot alege să se conecteze cu credențiale existente sau să creeze un nou cont.

Ecranul Principal (Acasă) servește drept nucleul central al aplicației și oferă acces rapid la toate funcțiile importante. Acesta poate include un meniu de navigare, widget-uri interactive și scurtături către cele mai folosite funcții. De aici, utilizatorii pot naviga către

profilul personal, hărți, informații, chat, forum, evenimente și coșul de cumpărături. Fiecare secțiune este accesibilă printr-o singură atingere sau clic.

Ecranul de Profil, pagina de profil oferă detalii personale ale utilizatorului și opțiuni de personalizare a informațiilor de cont. Utilizatorii pot modifica detalii precum numele, adresa de email, fotografia de profil și pot gestiona setările de securitate ale contului.

Ecranul de Chat permite utilizatorilor să comunice în timp real, făcând posibilă schimbul rapid de mesaje, imagini sau fișiere. Aceasta este o funcționalitate vitală pentru conversații directe și instantanee între doi sau mai mulți utilizatori, facilitând discuții fluide și dinamice.

Ecranul de Forum oferă o platformă mai structurată, unde discuțiile sunt organizate în subiecte și categorii, permițând utilizatorilor să posteze întrebări, răspunsuri și să inițieze discuții pe teme variate. Utilizatorii pot naviga prin diverse categorii și subiecte, găsind ușor discuții care le stârnesc interesul sau unde pot contribui cu informații valoroase.

Ecranul de evenimente din aplicație este destinată să ofere utilizatorilor o platformă centralizată unde pot descoperi, se pot înscrie și participa la diverse activități și evenimente viitoare. Aceasta funcționalitate este vitală pentru comunități și organizații care doresc să încurajeze implicarea activă și să ofere membrilor săi oportunități de networking, învățare și distracție.

Ecranul "Coș de Cumpărături" din aplicație este un element esențial pentru facilitarea tranzacțiilor comerciale. Aceasta permite utilizatorilor să gestioneze produsele selectate pentru achiziție, să verifice detalii, să facă ajustări și să finalizeze cumpărăturile într-un mod eficient și securizat.

5 Bibliografie

- [1]. „Care a fost prima aplicație mobilă pe piața?”,
<https://consumatoreuropean.ro/care-a-fost-prima-aplicatie-mobila-de-pe-piata/>
(accesat la data de 15.04.2024)

- [2]. D. Chiriac, „Istoria telefonului mobil- Ani 90 sau era Nokia”,
<https://www.gsmfit.ro/blog/post/istoria-telefonului-mobil-anii-90-era-nokia>
(acesat la data de 15.04.2024)

- [3]. Zearul Clujean, „Primele jocuri originale pe telefoanele mobile”,
<https://ziarulclujean.ro/primele-jocuri-originale-pe-telefoane-mobile-tetris-snake-brick-breaker-pairs-space-impact/>
(accesat la data de 15.04.2024)

- [4]. Universitate Politehnica Timisoara, „Campus Virtual”,
<https://apps.apple.com/ro/app/campus-virtual-upt/id1397170074?l=ro>
(accesat la data de 15.04.2024)

- [5]. Universitatea de Vest din Timisoara, platforma „e-Learning”,
<https://apps.apple.com/ro/app/elearning-uvt/id1581783733?l=ro>
(accesat la data de 15.04.2024)

- [6]. Unicampus,
<https://apps.apple.com/ro/app/unicampus/id1397170089?l=ro>
(accesat la data de 15.04.2024)

- [7]. Visual Studio Code, „Git îmbunătățit pentru IDE-ul dvs”,
<https://www.gitkraken.com/gitlens>
(accesat la data de 15.04.2024)

- [8]. Visual Studio Code,
<https://code.visualstudio.com/docs>
(accesat la data de 23.04.2024)

- [9]. Flutter –Wikipedia,
[https://en.wikipedia.org/wiki/Flutter_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software))
(accesat la data de 23.04.2024)
- [10]. Documentație Flutter,
<https://docs.flutter.dev/>
(accesat la data de 23.04.2024)
- [11]. Dart limbaj de programare – Wikipedia,
[https://en.wikipedia.org/wiki/Dart_\(programming_language\)](https://en.wikipedia.org/wiki/Dart_(programming_language))
(accesat la data de 23.04.2024)
- [12]. Dart- GitHub,
<https://github.com/dart-lang>
(accesat la data de 23.04.2024)
- [13]. Material Design –Introduction,
<https://m2.material.io/design/guidelines-overview>
(accesat la data de 23.04.2024)
- [14]. Widget-uri Flutter,
<https://docs.flutter.dev/ui/widgets/material>
(accesat la data de 25.04.2024)
- [15]. Firebase –Wikipedia,
<https://en.wikipedia.org/wiki/Firebase>
(accesat la data de 25.04.2024)
- [16]. Firebase pentru platforma Flutter,
<https://firebase.google.com/docs/flutter/setup?platform=web>
(accesat la data de 25.04.2024)
- [17]. Folosirea SDK-urilor terță în alte aplicații,
<https://support.google.com/googleplay/android-developer/answer/10358880?hl=ro>
(accesat la data de 25.04.2024)
- [18]. Android Studio – Wikipedia,
https://en.wikipedia.org/wiki/Android_Studio
(accesat la data de 25.04.2024)

- [19]. Rulează aplicații pe Emulatorul Android -Developers
<https://developer.android.com/studio/run/emulator>
(accesat la data de 25.04.2024)