

**ACADEMIA DE STUDII ECONOMICE DIN BUCUREȘTI FACULTATEA DE
CIBERNETICĂ, STATISTICĂ ȘI INFORMATICĂ ECONOMICĂ**

LUCRARE DE LICENȚĂ



**Aplicație informatică pentru gestiunea eficientă a
finanțelor personale**

Profesor coordonator:

Conferențiar universitar Andreescu Anca Ioana

Absolvent:

Popa Antonela Marina Narcisa

Cuprins

1. Introducere	3
2. Partea teoretică	5
2.1. Domeniul analizat	5
2.2 Aplicații existente în domeniul analizat	6
2.3 Tehnologii informatice utilizate pentru dezvoltarea aplicației	12
3. Proiectare	14
3.1 Scopul aplicației	14
3.2 Cerințele aplicației	14
3.3 Analiza sistemului informatic	16
3.3.1 Diagrama de activitate	16
3.3.2 Diagrama de clase	17
3.3.3 Diagrama de interacțiune	18
3.3.4 Diagrama de colaborare	19
3.3.5 Diagrama de desfășurare	20
3.4 Proiectarea bazei de date	21
3.4.1 FirebaseAuthentication	21
3.4.2 FirebaseDatabase	22
4. Implementare, utilizare și testare	24
5. Concluzii	33
6. Bibliografie	34
7. Anexe	35

1. Introducere

Gestiunea finanțelor personale reprezintă un aspect foarte important în viața oricărei persoane. Nu orice persoană are capacitatea de a-și administra cu chibzuință banii și de a lua decizii financiare raționale și informate, iar acest lucru are un impact semnificativ în ceea ce privește stabilitatea financiară și calitatea vieții.

Gestiunea eficientă a banilor are la bază diverse principii și tehnici financiare printre care se numără: bugetul personal, economisirea, reducerea și controlul cheltuielilor, prioritizarea și gestiunea datoriilor, investițiile și creșterea financiară, educația financiară. Însă, în ciuda acestor tehnici și principii financiare, cercetarea în domeniu nu a explorat partea de dezvoltare de algoritmi în ceea ce privește analiza datelor financiare sau integrarea inteligenței artificiale, utilă în a face recomandări în privința maximizării economiilor și reducerii cheltuielilor redundante.

Motivația alegerii acestei teme constă în efectul direct pe care îl are gestiunea finanțelor proprii asupra vieții individuale. Experiența ne-a demonstrat că nicio persoană, indiferent de cât de precaută ar fi, nu poate fi complet pregătită pentru o criză neașteptată. Acest lucru a devenit evident în timpul pandemiei, iar mulți oameni continuă să învețe această lecție în fiecare lună din pricina inflației actuale. Într-un astfel de context, având în vedere schimbările și incertitudinile economice cu care vin la pachet aceste evenimente, gestiunea eficientă a finanțelor proprii capătă o importanță deosebită pentru a face față și pentru a menține stabilitatea financiară. De asemenea, pandemia a evidențiat importanța economisirii și constituirii unui fond de urgență deoarece mulți oameni și-au pierdut locul de muncă ori s-au confruntat cu probleme de sănătate.

De asemenea, un studiu realizat de Institutul Român pentru Evaluare și Strategie (IRES) surprinde faptul că un român din patru, adică 26% nu ține deloc evidența cheltuielilor și veniturilor, 40% țin evidența în minte, 19% țin evidența în agendă sau pe hârtie, 12% folosesc o aplicație de Internet și doar 7% utilizează o aplicație pe telefon [SBC22]. Având în vedere că peste 65% din oameni nu țin evidența sau o țin în minte, demonstrează o planificare a bugetului ineficace. Acest lucru poate proveni din cauza lipsei cunoștințelor financiare, fapt ce duce la lipsa monitorizării active a veniturilor și cheltuielilor, iar neținându-se cont de un buget alocat pentru anumite cheltuieli se ajunge la instabilitate financiară, reușita economisirii fiind nerealizabilă. De asemenea, oamenii nu au capacitatea de a realiza cât de mult cheltuie pe anumite categorii de articole până nu vizualizează o analiză aprofundată.

Pentru a rezolva problemele enunțate este recomandată întocmirea unui buget financiar care să surprindă fluxul intrărilor și ieșirilor, lucru posibil prin intermediul aplicațiilor de gestionare a bugetului care dețin toate instrumentele necesare realizării unor astfel de rapoarte, benefice pentru persoanele care nu au cunoștințe financiare și nu numai. Utilizatorii trebuie doar să introducă datele necesare precum veniturile și cheltuielile înregistrate, iar aplicația se va ocupa de generarea rapoartelor și gestiunea banilor. Persoana în cauză devine astfel conștientă de cheltuielile pe care le face în fiecare lună și poate lua decizii în ceea ce privește starea financiară personală astfel încât să-și poată atinge obiectivele pe care și le propune.

Piața actuală pune în evidență o gamă largă de aplicații destinate acestei activități, însă problema principală a soluțiilor existente este complexitatea acestora. Majoritatea aplicațiilor de acest tip presupun o interfață greoaie, cu multe elemente și destul de greu de utilizat pentru un procent mare din populație. Din această cauză este nevoie de crearea unei aplicații ușor de înțeles, cu o interfață prietenoasă, dar în același timp eficientă.

2.Parte teoretică

2.1.Domeniul analizat

Unul dintre cele mai importante concepte ale finanțelor private este bugetul personal. Astfel că bugetul personal include toate sursele de venit precum și toate cheltuielile înregistrate.

Veniturile înregistrate sunt reprezentate de sumele de bani obținute într-un anumit interval de timp, ele fiind lunare în majoritatea cazurilor. Acestea pot fi venituri din salariu, din investiții, din încasări chirie, din bonusuri, din investiții etc.

Cheltuielile înregistrate sunt reprezentate de sumele de bani folosite pentru achiziția de bunuri și servicii sau pentru plata obligațiilor financiare. Ele sunt constituite din cheltuieli fixe precum chiria, abonamentele, facturile, ratele la bancă și din cheltuieli variabile precum cheltuieli cu mâncarea, achiziții de haine, produse de uz personal etc. În bugetul personal trebuie luate în calcul și cheltuielile care necesită să fie plătite trimestrial sau anual cum ar fi: impozitele pentru casă, asigurările pentru mașină etc.

Un buget este util pentru gestiunea eficientă a finanțelor personale deoarece prin bugetul personal se conturează o viziune mai clară a intrărilor și ieșirilor, monitorizarea, planificarea și controlul eficient al veniturilor și cheltuielilor devenind o sarcină facilă. O astfel de planificare poate conduce la atingerea obiectivelor pe termen scurt, cât și a celor pe termen lung. Obiectivele pe termen scurt definesc economiile realizate pe o perioadă mai mică de un an, eliminând cheltuielile mai puțin esențiale. Un exemplu practic în acest sens este reprezentat de comandarea mâncării de la restaurant în fiecare lună, fapt ce generează cheltuieli mari și neesențiale, iar acest obicei ar putea fi înlocuit cu prepararea meselor acasă și astfel se atinge un obiectiv pe termen scurt. Acei bani economisiți pot fi utilizați pentru crearea unui fond de urgență în ceea ce privește situațiile neprevăzute (o urgență medicală, pierderea locului de muncă) ori pentru un obiectiv setat dinainte cum ar fi achiziția unui laptop. Un obiectiv pe termen lung ar putea fi reprezentat de decizia de a cumpăra o mașină sau o casă, precum și de economisirea banilor pentru pensie, fapt ce presupune stabilirea unui plan consolidat de economisire și gestionare a finanțelor.

Economisirea reprezintă procesul de a păstra o parte din veniturile înregistrate cu scopul de a le utiliza în viitor. Bineînțeles, economiile pot fi realizate doar în momentul în care cheltuielile înregistrate sunt mai mici decât veniturile, iar pentru atingerea acestui țel este necesară impunerea unui buget pentru fiecare categorie de cheltuieli în parte care trebuie

respectată cu strictețe. Economii realizate pot fi folosite și în scopul investițiilor într-un anumit activ pentru a obține beneficii financiare viitoare. Investițiile pot fi o strategie eficientă de creștere a veniturilor, însă trebuie făcute în cunoștință de cauză presupunând o analiză riguroasă și o educație financiară bine consolidată.

Putem afirma că stabilitatea și satisfacția financiară durabilă este un rezultat al aptitudinii de a ține sub control cheltuielile și de a economisii. Reușita de a descoperii și înțelege modul în care să crești ritmul economiilor și să construiești un buget personal eficient îți va oferi capacitatea de a face alegeri informate.

2.2 Aplicații existente în domeniul analizat

Pentru gestiunea eficientă a tuturor acestor componente care alcătuiesc bugetul personal și conduc la luarea unor decizii financiare adecvate, precum și la stabilitate financiară au fost create diferite aplicații ce ușurează viața utilizatorilor prin intermediul automatizării unor procese.

Pentru o bună analiză a sistemelor existente trebuie să urmărim câteva lucruri definitorii a unei astfel de aplicații care face obiectul studiului precum:

- Existența unei interfețe intuitive și prietenoase cu ochiul utilizatorului astfel încât aceștia să se poată orienta repede și să utilizeze funcțiile specifice fără a întâmpina dificultăți.
- Configurarea rapidă și simplă a aplicației fără ca utilizatorii să fie nevoiți să parcurgă procese complexe sau să introducă prea multe informații. Ar trebui să se urmărească minimizarea setărilor esențiale în scopul economisirii timpului.
- Menținerea și actualizarea ușoară a informațiilor care să permită utilizatorilor să-și introducă și să-și actualizeze rapid și ușor datele din aplicație precum veniturile, cheltuielile, datoriile, economiile și bugetele.
- Opțiuni de personalizare și funcționalități esențiale pentru gestiunea eficientă a intrărilor și ieșirilor, utilizatorii având posibilitatea de a-și personaliza categoriile, bugetele și de a-și seta propriile obiective în funcție de nevoile personale.
- Protejarea datelor financiare este un punct important de luat în calcul pentru o astfel de aplicație.
- Disponibilitatea gratuită, precum și existența opțiunilor premium pentru persoanele care doresc eliminarea reclamelor sau funcționalități suplimentare. Desigur că și posibilitatea

încercării gratuite este un lucru benefic pentru utilizatorii pentru a vedea dacă rezonază cu aplicația sau nu.

În cele ce urmează vom analiza unele dintre cele mai populare aplicații de gestiune a finanțelor personale.

Mint

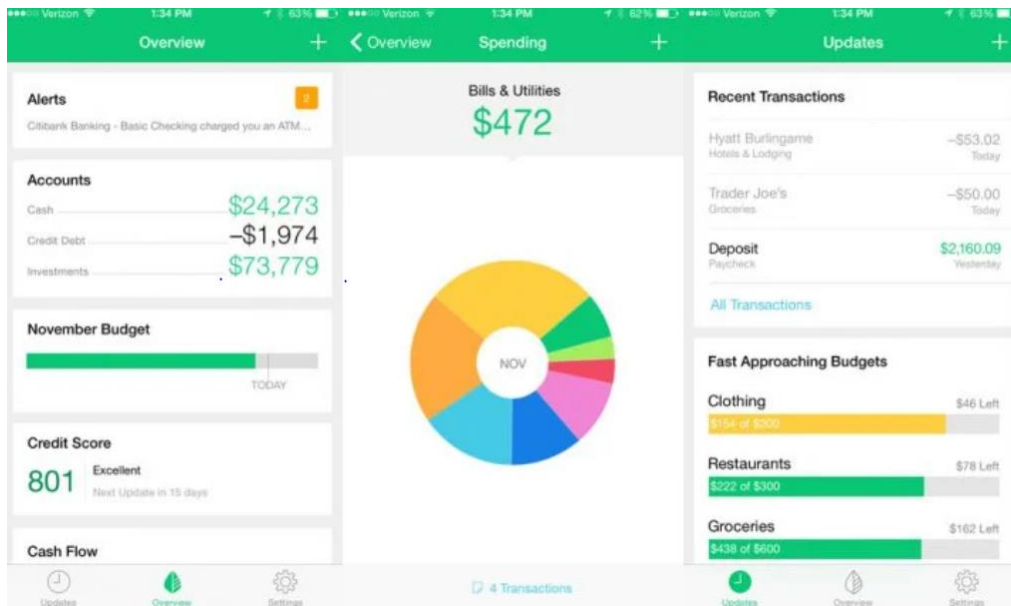


Fig. 2.1 Aplicația Mint

Mint este una dintre cele mai populare și utilizate aplicații de gestionare a bugetului. Cea mai utilizată funcție a aplicației este dată de posibilitatea conectării aplicației la contul bancar. Mint împarte tranzacțiile, le ține evidența și realizează un model personalizat de buget. De asemenea, aplicația poate trimite notificări prin care alertează utilizatorul de depășirea bugetului alocat. Mint integrează și o funcționalitate prin care se poate face plata facturilor din aplicație pentru a ajuta la o gestionare mai eficientă a acestora.

Aplicația este disponibilă pentru dispozitivele iPhone, iPad și Android, iar pe lângă acestea oferă și posibilitatea de a urmări bugetul pe o pagină web, online. [BBA20]

Wally

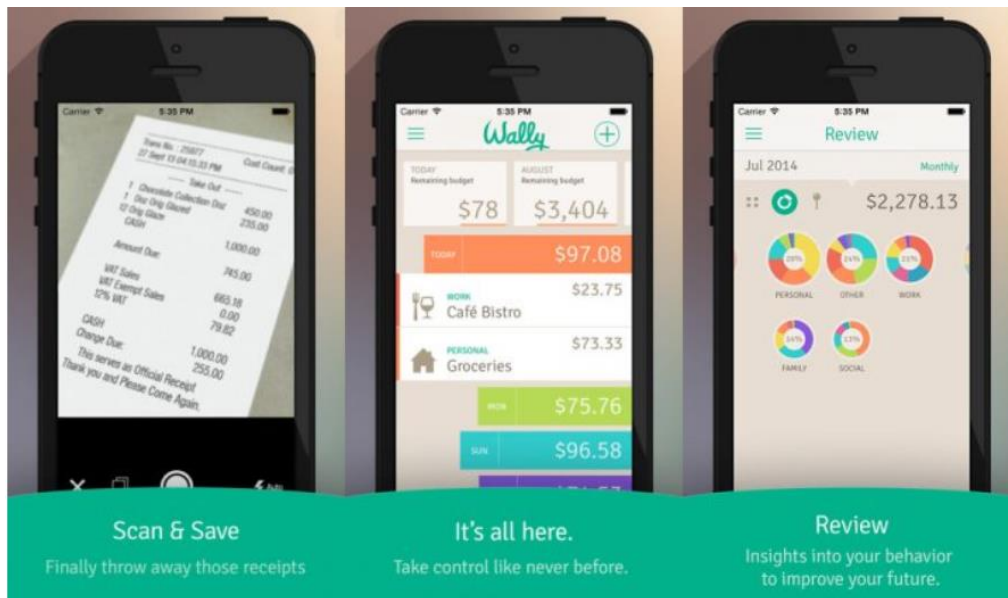


Fig. 2.2 Aplicația Wally

Wally este o aplicație foarte ingenioasă ce și-a câștigat clienții cu ajutorul interfeței prietenoase și prin funcționalitatea ce o diferențiază de toate celelalte aplicații din acest domeniu și anume implementarea unui algoritm care permite înregistrarea cheltuielilor direct de pe chitanțe, bonuri fiscale, facturi și așa mai departe. Pe lângă această funcționalitate, Wally poate trimite notificări în momentul în care o factura este scendentă sau când se ajunge la un obiectiv de economii dorit. Din păcate aplicația este disponibilă doar pentru iPhone.

You Need a Budget-YNAB

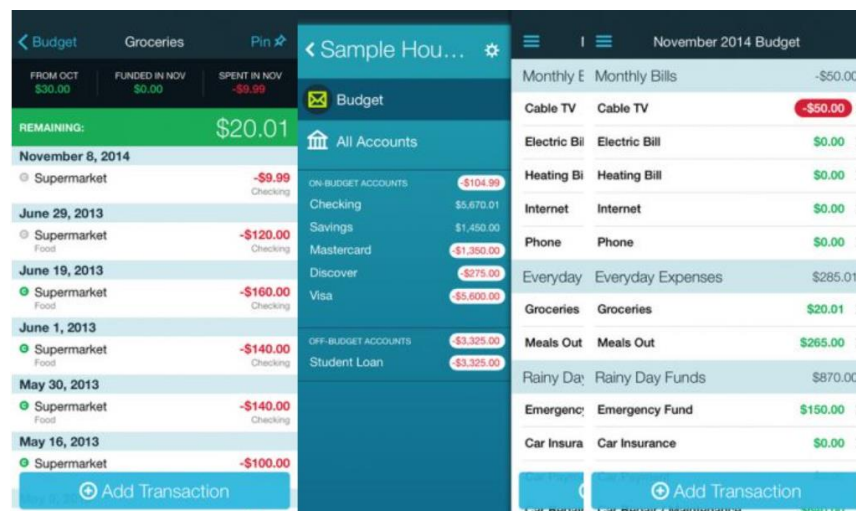


Fig. 2.3 Aplicația YNB

YNAB este o altă aplicație complexă și eficientă pentru conceperea unui buget personal pe care utilizatorii să-l poată urmări cu succes în scopul de a economisi. YNAB se diferențiază de restul aplicațiilor prin pricina faptului că are capacitatea de a te antrena direct în procesul de realizare a unui plan financiar care să funcționeze pentru tine. Aplicația îți oferă suport în ceea ce privește minimizarea cheltuielilor, punând la dispoziție inclusiv tutoriale, cursuri despre educație financiară, sfaturi și multe altele. Aplicația este disponibilă pentru calculator, iPhone și Android, nu este gratuită și se achită 84 de dolari pentru un plan anual, dar există posibilitatea unei luni de probă gratuite. [BBA20]

Mvelops



Fig 2.4 Aplicația Mvelops

Mvelops poate fi privită ca o combinație între Mint și Wally deoarece aceasta preia funcționalitatea pe care o implementează Mint, cea de a conecta conturile bancare și funcționalitatea de scanare implementată de Wally. Prin intermediul aplicației se pot planifica plățile facturilor și se poate vizualiza bugetul personal. Aplicația este disponibilă pentru iPhone și Android, nu este gratuită, abonamentul fiind de 6 dolari pe lună. [BBA20]

Spendee

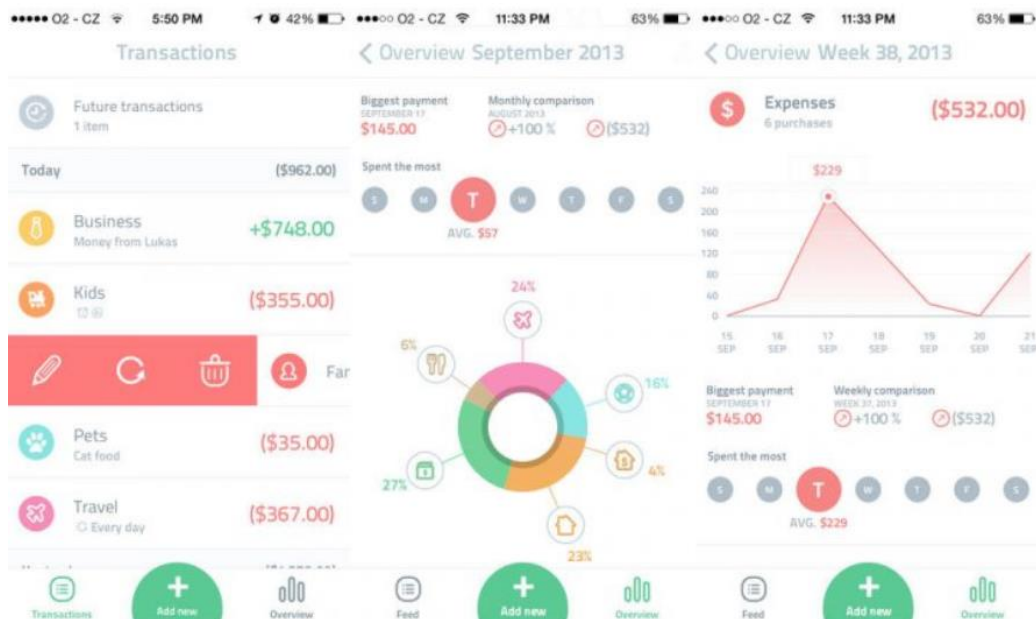


Fig. 2.5 Aplicația Spendee

Pentru cei ce vor să-și alcatuiască un buget eficient, dar nu știu de unde să înceapă, aplicația Spendee este una ideală. Aceasta este o aplicație ușor de înțeles și poate monitoriza cheltuielile înregistrate astfel încât să ofere o perspectivă de ansamblu asupra categoriilor de cheltuieli unde se irosesc cei mai mulți bani și unde se poate economisi. Aplicația gratuită și disponibilă pe iPhone și Android. [BBA20]

Acestea sunt doar câteva exemple din multitudinea de aplicații create în scopul gestiunii eficiente a finanțelor, iar în urma analizei putem sublinia câteva aspecte importante ale acestor aplicații care le fac utile și populare în ceea ce privește gestiunea finanțelor personale:

- Permit urmărirea în mod eficient a veniturilor și cheltuielilor oferind o viziune clară asupra modului de cheltuire a banilor. Țin evidența obiceiurilor de cheltuieli și a locurilor unde se pot realiza economii.
- Oferă funcții de creare a unui buget personalizat, oferind utilizatorului posibilitatea de a alocă anumite sume de bani pentru diferite categorii: mâncare, casă, transport, divertisment etc. Astfel se pot urmări cheltuielile pentru fiecare categorie creată și menținerea acestora în limitele bugetului alocat.

- Sincronizează conturile bancare și este un aspect important al deoarece scutește utilizatorul de introducerea manuală a tranzacțiilor, permițând importul automat al veniturilor și cheltuielilor înregistrate și actualizarea în timp real a informațiilor financiare
- Oferă statistici pe baza intrărilor și ieșirilor efectuate, înfățișând o perspectivă mai aprofundată asupra situației financiare a utilizatorului. Acestea evidențiază prin grafice și diagrame fluxul de bani, evoluția economiilor și alte lucruri relevante.
- Sistemul de alertare, implementat doar de unele aplicații, anunță în timp real utilizatorul de anumite evenimente cum ar fi: depășirea bugetului alocat, plăți care trebuie efectuate în următoarea perioadă și multe altele.

Bineînțeles că fiecare aplicație se diferențiază de cealaltă prin specificul ei, prin interfață și prin funcționalitățile pe care le are.

În contrast cu aplicațiile studiate, putem observa că aplicația realizată de mine prezintă anumite asemănări și diferențe. Putem identifica asemănări precum funcționalități bazilare de creare a unui buget, monitorizarea intrărilor și ieșirilor și vizualizarea unor informații din care utilizatorul să-și dea seama dacă se află în limitele bugetului. Ceea ce diferențiază aplicația mea de cele existente este posibilitatea setării unor obiective și realizarea unui plan eficient de gestiune a finanțelor astfel încât să fie posibilă realizarea de economii care mai apoi să fie utilizate în scopul atingerii obiectivelor.

Pentru o mai bună înțelegere a asemănărilor și deosebirilor dintre aplicația mea și alte aplicații din același domeniu, am realizat tabelul următor:

Asemănări	Deosebiri
<ul style="list-style-type: none"> - Crearea unui buget - Monitorizarea intrărilor și ieșirilor - Vizualizare grafic sugestiv 	<ul style="list-style-type: none"> - Interfață prietenoasă - Setare de obiective - Stabilirea unor bugete pe categorii, în scopul economisirii și atingerii obiectivelor - Vizualizare de informații utile pentru a rămâne în parametrii bugetului

Tabel 2.1 Comparații

Spre deosebire de celelalte aplicații de gestiune eficientă a finanțelor proprii , aplicația mea nu implementează sincronizarea cu conturi bancare și nu integrează sisteme complexe de alertare a utilizatorului, dar acestea pot fi luate în calcul pentru viitoare extinderi.

2.3 Tehnologii informatice utilizate pentru dezvoltarea aplicației

Deoarece aplicațiile mobile oferă mai multă comoditate, accesibilitate și portabilitate, am ales ca aplicația implementată să fie una mobilă. Datorită faptului că este o aplicație mobilă, accesul la informațiile financiare se poate realiza oriunde și oricând, în timp real se poate urmări și actualiza bugetul, se pot introduce tranzacții sau se pot verifica balanțele actuale, chiar și atunci când te afli în deplasări.

Pe primele două poziții în topul sistemelor de operare pentru care se pot dezvolta aplicații mobile se află Android și iOS, unde Android deține monopolul de piață cu un procent de 71,47%, urmat de iOS cu 27,85 % , după cum putea vedea și în ilustrația de la Fig. 1.7 [PRW23]

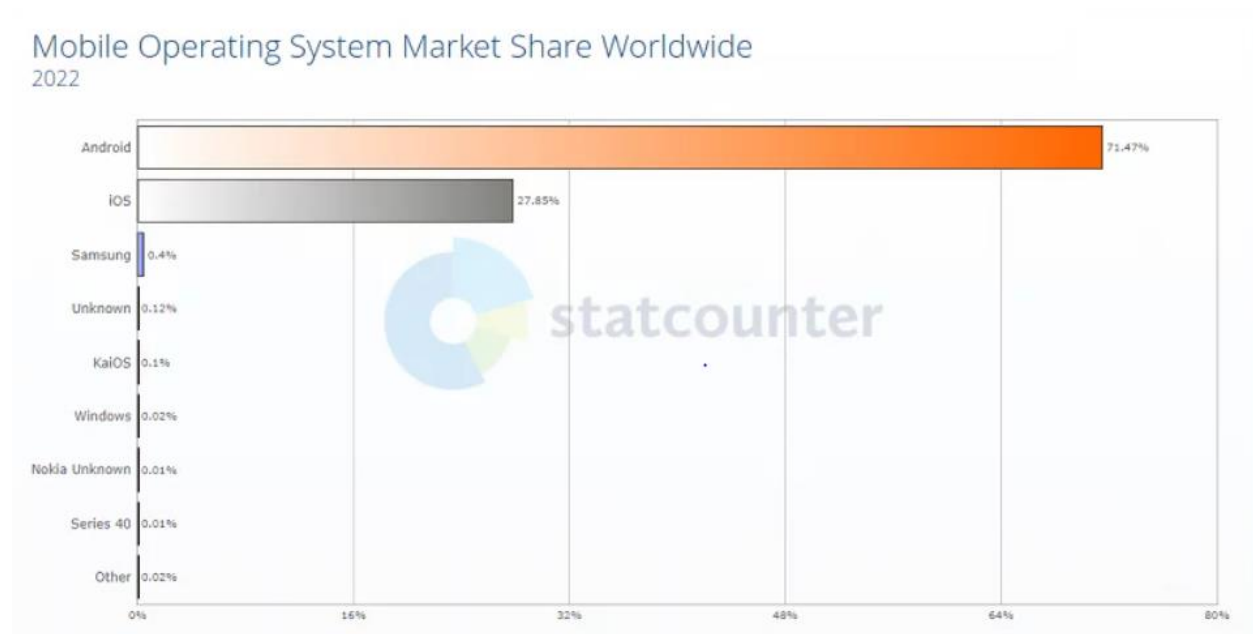


Fig. 2.6 Cele mai utilizate sisteme de operare pentru dispozitivele mobile în 2022

Având în vedere că Android deține cea mai mare popularitate în ceea ce privește numărul de utilizatori care folosesc dispozitive inteligente (telefoane, ceasuri, televizoare inteligente), am ales să-mi dezvolt aplicația pentru gestiunea eficientă a bugetului personal în această tehnologie pentru a fi disponibilă unui public larg.

Android este un sistem de operare și o platformă open-source, bazată pe Linux, conceput și dezvoltat de către compania Android Inc, care a fost achiziționată de către Google, sub administrația careia se află și în ziua de astăzi, împreună cu Open Handset Alliance [FGA23] [DGA19]. Caracteristica principală a acestui sistem de operare este versatilitatea. Acesta poate rula pe diverse dispozitive și este adaptabil la diferite dimensiuni de ecrane, rezoluții și specificații hardware, fiind beneficiu atât pentru dezvoltator cât și pentru utilizatori. De exemplu o aplicație dezvoltată pentru Android poate rula atât pe telefoanele mobile noi cu specificații hardware de top cât și pe cele mai vechi cu specificații hardware limitate. Acest lucru este posibil deoarece aplicația poate adapta în mod automat nivelul de detalii grafice, rezoluția sau alte aspecte pentru a se potrivi cu capacitățile dispozitivului pe care rulează.

Mediul de dezvoltare pentru implementarea aplicației mele este AndroidStudio care este bazat pe un editor de cod foarte purernic și are în componență: Android SDK, Android Virtual Device Manager(AVD) și Android Emulator. Android SDK reprezintă o colecție de biblioteci și instrumente utile pentru a crea aplicații Android. AVD Manager este un instrument utilizat pentru a crea și gestiona dispozitivele virtuale pentru a testa aplicațiile. Android Emulator este un dispozitiv virtual, util pentru dezvoltatori deoarece permite testarea aplicației fără a fi necesar un dispozitiv fizic. [MAS] [OAS]

În scopul dezvoltării aplicației, pentru stocarea datelor și pentru autentificarea utilizatorilor, am folosit Firebase care este o platformă de furnizare a serviciilor în cloud (BaaS). Aceasta oferă o gamă largă de servicii pentru dezvoltarea aplicațiilor web, iOS și Android, punând la dispoziție funcționalități de back-end, fără a fi nevoie să se creeze propriul server și infrastructură. [FBI] Firebase pune la dispoziție mai multe funcționalități utile în procesul de dezvoltare a aplicațiilor Android printre care putem enumera Firebase Authentication și Firebase Database, de care m-am folosit în implementarea aplicației mele. Firebase Authentication este folosit în special pentru autentificarea utilizatorilor din aplicația noastră [FBA]. Firebase Database este o bază de date NoSQL cu ajutorul căreia putem stoca și sincroniza datele în timp real. Datele din această bază de date sunt stocate sub formă de JSON, iar prin utilizarea API-urilor le putem accesa și manipula. [FRD]

3.Proiectare

3.1 Scopul aplicației

Pentru a începe partea de proiectare avem nevoie să știm exact care este scopul aplicației. Aplicația dezvoltată are ca scop optimizarea gestiunii bugetului pentru utilizatorii săi în vederea atingerii obiectivelor dorite. Odată ce aplicația este instalată pe dispozitivul personal, aceasta trebuie configurată prin introducerea manuală, de către utilizator, a câtorva elemente cheie necesare în realizarea bugetului. Mai întâi de toate utilizatorul introduce în aplicație veniturile de care dispune și își setează un buget pentru fiecare categorie de cheltuieli în parte în limitele căruia trebuie să se încadreze. Mai apoi urmează ca pe parcursul lunii să-și adauge toate cheltuielile efectuate, pe baza cărora se menține o evidență a bugetului rămas reprezentând economiile. De asemenea, utilizatorul are posibilitatea de a-și seta și anumite obiective pe care ar dori să le atingă cum ar fi achiziția unui laptop sau achiziția unui curs, iar în funcție de economiile realizate se poate efectua achiziția. Dacă nu, economiile din luna curentă se adaugă la veniturile lunii viitoare.

Pentru o mai viziune mai clară a cheltuielilor înregistrate pentru luna în curs, utilizatorul are posibilitatea de a analiza un grafic sugestiv cu cheltuielile pe categorii din luna în curs astfel încât să realizeze mai ușor cum ar trebui să-și ajusteze cheltuielile pentru a-și maximiza economiile.

3.2 Cerințele aplicației

Pentru a identifica și a descrie cerințele aplicației vom folosi reprezentarea grafică oferită de diagrama cazurilor de utilizare care surprinde comportamentul vizibil al sistemului din perspectiva utilizatorilor.

Pentru o mai bună înțelegere a ceea ce trebuie să facă sistemul putem privi diagrama cazurilor de utilizare din Fig. 3.1.

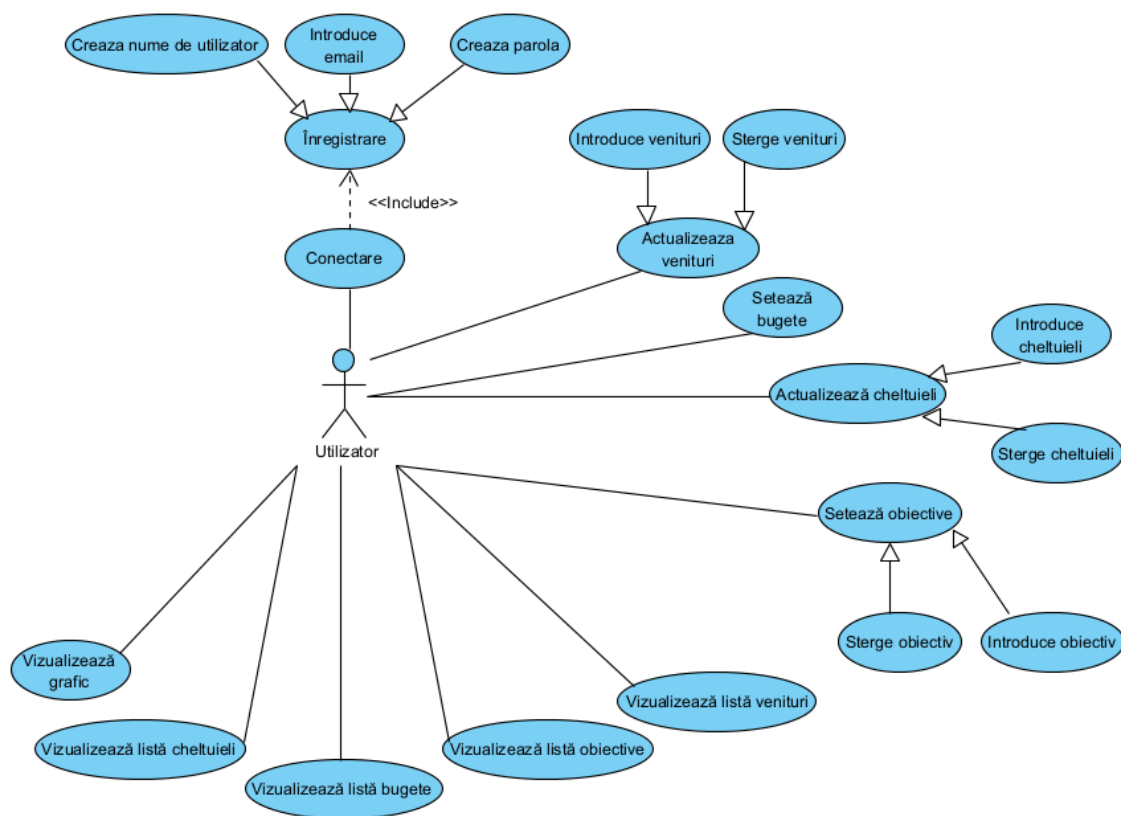


Fig. 3.1 Diagrama cazurilor de utilizare

Pentru a folosi aplicația, utilizatorul trebuie să se conecteze în aplicație pe baza numelui și parolei asociate. În situația în care utilizatorul nu are un cont atunci acesta trebuie să-și creeze unul, după care va putea să se conecteze. Pentru a actualiza sistemul cu informațiile utilizatorului, acesta trebuie să introducă veniturile înregistrate, având și posibilitate de a le șterge. Mai departe este nevoie de setarea bugetului pentru fiecare categorie de cheltuieli în parte. Actualizarea cheltuielilor pe parcursul lunii se realizează prin operațiuni de adăugare și ștergere, în caz că utilizatorul a introdus o dată eronată. Setarea obiectivelor se realizează prin adăugarea unui obiectiv sau ștergerea acestuia în caz că nu mai prezintă interes pentru utilizator. Pe lângă aceste operațiuni de actualizare a veniturilor, cheltuielilor, bugetului și obiectivelor, utilizatorul are posibilitatea de a vedea toate aceste actualizări sub forma de listă pentru a contura o imagine mai clară. Vizualizarea graficului pe categorii este o altă cerință a sistemului, de care utilizatorul poate beneficia. Dacă se dorește utilizatorul are posibilitatea de a se deconecta.

3.3 Analiza sistemului informatic

Analiza sistemului informatic este un pas foarte important în privința proiectării și mai apoi a implementării aplicației. Aceasta implică investigarea detaliată a componentelor sistemului informatic precum și a interacțiunilor dintre acestea. Astfel că, în cele ce urmează vom trece printr-o serie de diagrame ce vor modela și vor evidenția interacțiunile sistemului informatic.

3.3.1 Diagrama de activitate

Diagrama de activitate este utilizată în modelarea proceselor, aceasta surprinzând pașii sau acțiunile care sunt executate în cadrul unui proces sau a unei activități, oferind o reprezentare vizuală asupra modului în care sunt legate acțiunile și cum se desfășoară fluxul de lucru în cadrul sistemului.

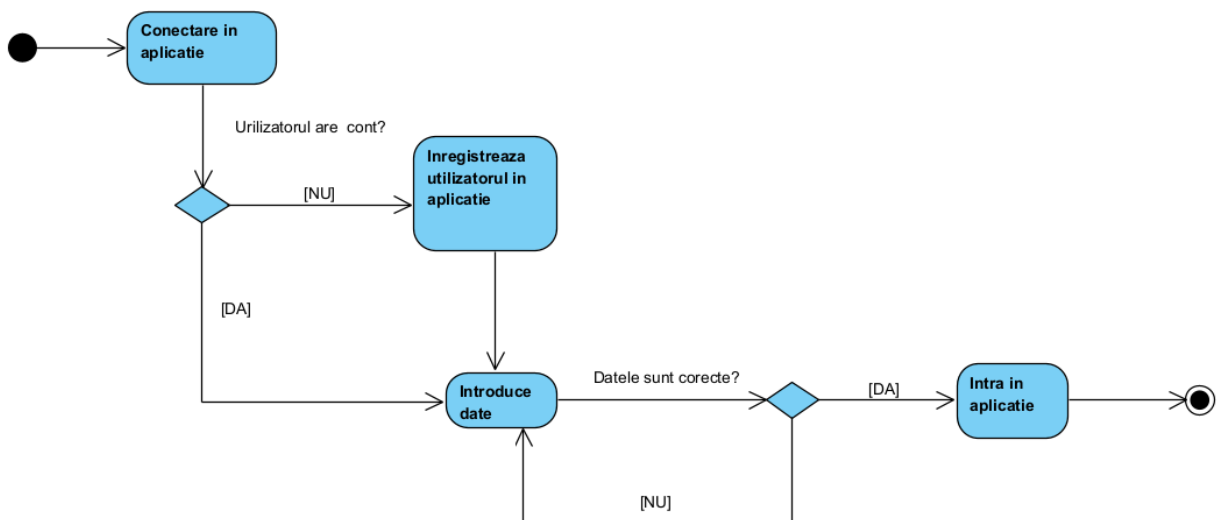


Fig. 3.2 Diagrama de activitate - Conectare și înregistrare

Diagrama de activitate din Fig. 3.2 prezintă modul în care funcționează procesul de autentificare în aplicație de la deschidere până la utilizarea propriu zisă a acesteia. Dacă utilizatorul a instalat aplicația și dorește să se conecteze, dar nu are cont, acesta trebuie să se înregistreze prin adăugarea unui nume de utilizator, a unui email și parola. După aceea sistemul verifică dacă datele sunt corecte și le adaugă în baza de date, iar utilizatorul poate intra în aplicație. În cazul în care datele nu sunt corecte sistemul va anunța utilizatorul și datele vor trebui introduse din nou. În al doilea caz, dacă utilizatorul are un cont, acesta își introduce datele de conectare, iar sistemul verifică dacă sunt corecte prin confruntarea cu baza de date. Dacă

datele sunt corecte utilizatorul poate intra în aplicație,iar dacă nu este nevoit sa-si introducă din nou datele de conectare.

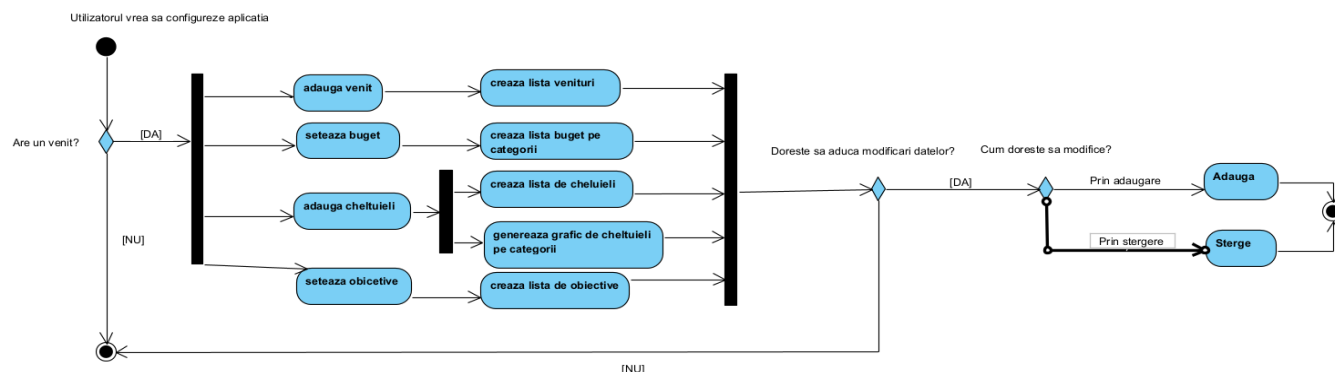


Fig. 3.3 Diagrama de activitate - Configurare date în aplicație

Diagrama de activitate din Fig.3.3 prezintă fluxul pe care îl urmează sistemul în momentul în care utilizatorul dorește să adauge date în aplicație. Astfel că dacă utilizatorul nu are venituri fluxul se închide fiind imposibilă introducerea datelor în sistem fără acestea. În cazul în care acesta are venituri, se introduc în sistem date precum: venitul, bugete, cheltuieli și obiective. Sistemul crează pe baza datelor introduse listele aferente, iar pentru cheltuieli crează și un grafic. Dacă utilizatorul nu dorește să aducă modificări fluxul se închide. Dacă utilizatorul dorește să aducă modificari, are două opțiuni, stergere și adaugare, iar după ce modificările sunt facute , fluxul se închide.

3.3.2 Diagrama de clase

Diagrama de clase vine să structureze sistemul prin evidențierea claselor și a relațiilor dintre ele.

În diagrama de clase din Fig. 3.4 sunt evidențiate câteva dintre clasele care fac parte din sistem. Aceasta surprinde atât legături de compunere, cele reprezentate printr-un romb umplut, cât și legături de agregare, reprezentate printr-un romb gol. Astfel putem afirma că activitățile reprezentate de clasele AdaugăvenituriActivity, AdaugaCheltuialaActivity, AdaugăBugetActivity, AdaugăObiectiveActivity depind în mod direct de clasa AcasaActivity, fără de existența careia, celelalte nu ar exista. Observăm că între AcasaActivity și Conectare exista o legătură de 0 la mai multi ceea ce înseamnă că la clasa AcasaActivity se pot conecta mai multi utilizatori.

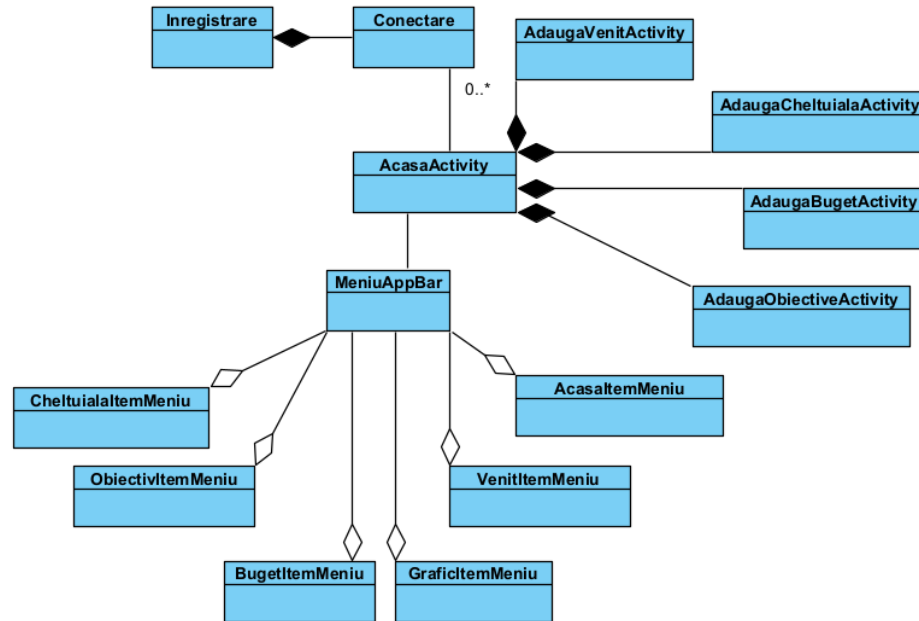


Fig. 3.4 Diagrama de clase

3.3.3 Diagrama de interacțiune

Diagrama de interacțiune pune în evidență dinamica sistemului, cu ajutorul obiectelor care comunică între ele prin intermediul mesajelor .

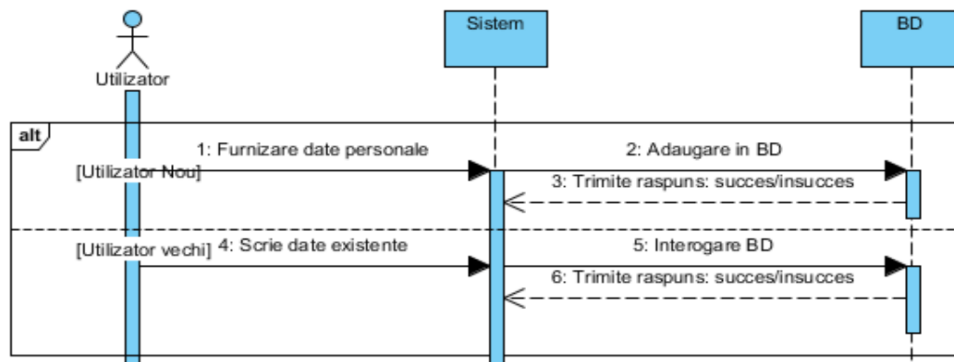


Fig. 3.5 Diagrama de interacțiune - Conectare și înregistrare

Diagrama de interacțiune prezentată în Fig. 3.5 surprinde comunicarea dintre utilizator, sistem și baza de date. Dacă utilizatorul este nou, înseamnă că acesta nu are un cont și trebuie să se înregistreze în sistem. Astfel că, utilizatorul trimite sistemului datele personale, iar sistemul

trimite aceste date bazei de date pentru adaugare. Baza de date, la rândul ei, trimite sistemului un raspuns afirmativ dacă datele au fost adaugate cu succes ,respectiv un raspuns negativ dacă datele nu sunt adăugate. Dacă utilizatorul este vechi, înseamnă că acesta are un cont și tot ceea ce trebuie să facă este să se conecteze. Acesta trimite datele existente de conectare către sistem. Sistemul face o interogare pe baza de date,iar dacă datele corespund cu cele introduse de utilizator, baza de date trimite un raspuns pozitiv către sistem, iar utilizatorul poate intra în aplicație.

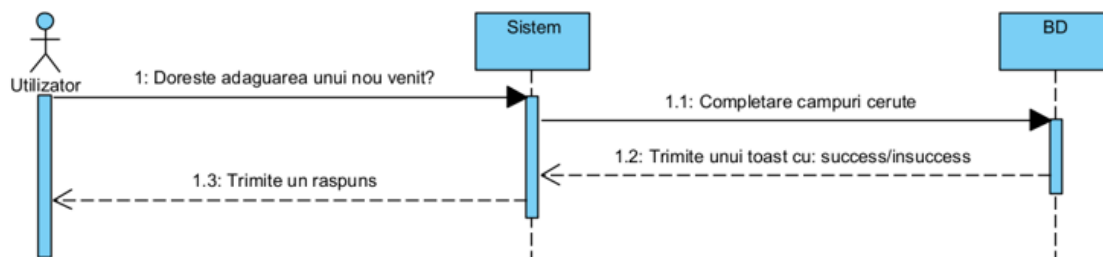


Fig. 3.6 Diagrama de interacțiune – Adaugare date în aplicație

Diagrama din Fig. 3.6 reprezintă comunicarea dintre utilizator și cele doua componente cu care interacționează. Astfel că, dacă utilizatorul dorește să adauge un venit, trebuie introduse datele despre acesta care urmează să fie salvate în baza de date. Baza de date trimite raspuns sistemului dacă datele au fost înregistrate cu succes, iar utilizatorul este înștiințat de către sistem în ceea ce privește adăugarea datelor.

Această diagramă poate fi adaptată și pentru cazul în care avem de adăugat cheltuieli, bugete sau obiective. Tot același principiu se respectă și în cazul ștergerilor.

3.3.4 Diagrama de colaborare

Diagrama de colaborare este creată în limbajul BPMN (Business Process Model and Notation) și permite reprezentarea grafică a interacțiunilor dintre participanții, în cazul nostru sistemul și utilizatorul, prin intermediul fluxurilor de mesaje

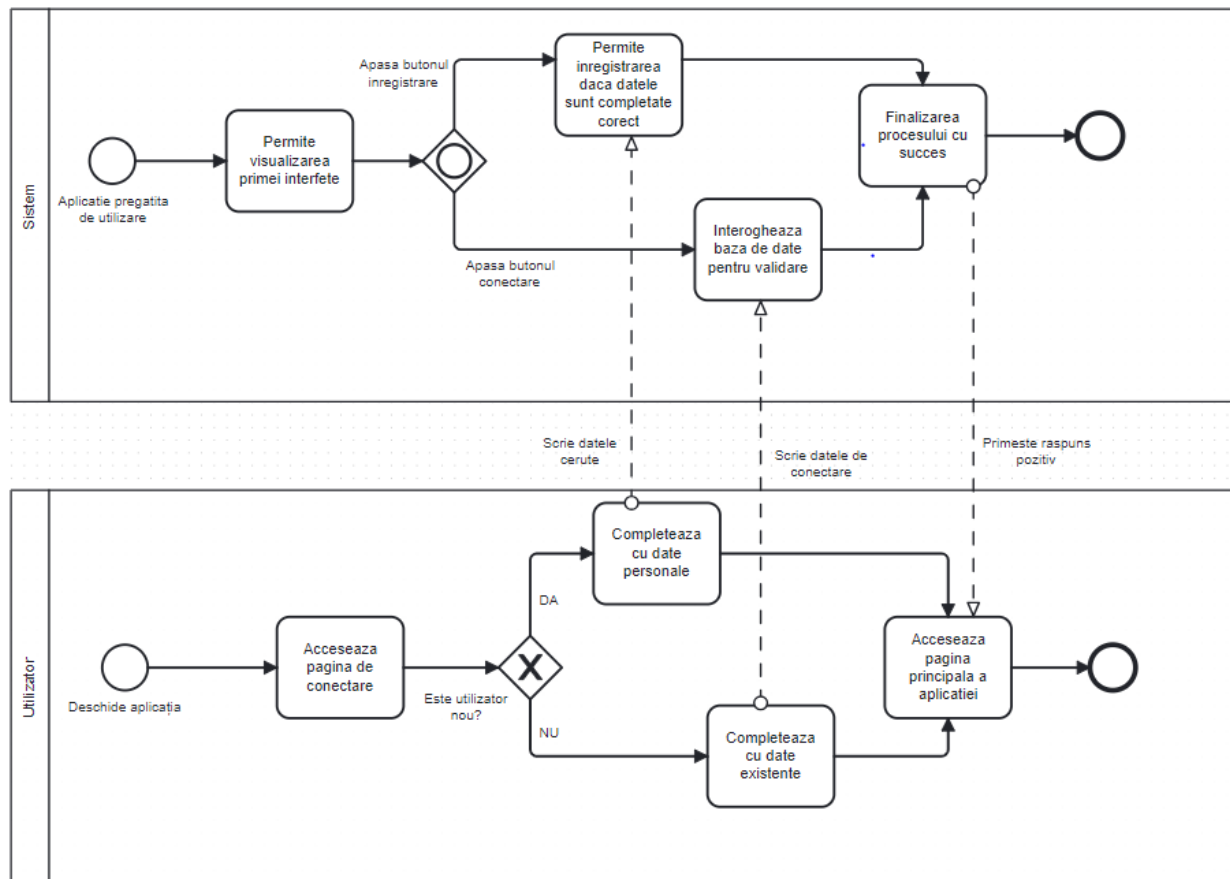


Fig. 3.7 Diagrama de colaborare

Diagrama reprezentată în Fig. 3.7 crează o imagine mai detaliată a modului în care utilizatorul interacționează cu sistemul în privința conectării în aplicație.

3.3.5 Diagrama de desfășurare

Diagrama de desfășurare reprezintă o vizualizare simplificată a componentelor hardware care alcătuiesc sistemul.

După cum putem observa în diagrama de mai jos, Fig 3.8 , sistemul nu are multe componente hardware, aplicația putând fi accesată de pe mai multe dispozitive. De asemenea, aplicația se folosește de conexiunea la internet pentru a putea accesa toate datele stocate în Firebase.

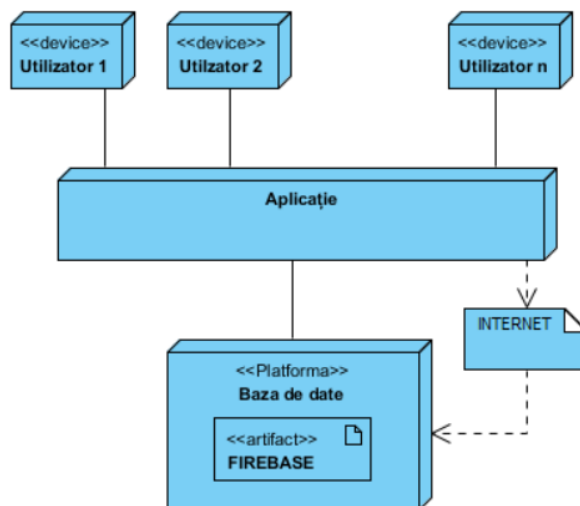


Fig. 3.8 Diagrama de desfășurare

3.4 Proiectarea bazei de date

Pentru a începe implementarea aplicației este nevoie să ne structurăm o bază de date corectă, acest pas fiind unul complex și critic pentru aplicația noastră deoarece pe baza acesteia se vor gestiona și interoga datele. După cum am menționat și mai sus, am ales să utilizez platforma Firebase deoarece serviciile oferite de acestea sunt ușor de utilizat și eficiente. Dintre serviciile oferite, eu am folosit FirebaseAuthification și FirebaseDatabase.

3.4.1 FirebaseAuthification

Acest serviciu l-am folosit pentru înregistrarea și conectarea utilizatorilor în aplicație. După ce utilizatorul introduce datele în aplicație, acestea sunt preluate în cloud și li se va atribui un cod unic numit UID (UniqueIdentifier).

<div> Search by email address or user UID <div>ADD USER</div> </div>				
Email	Providers	Created	Signed In	User UID ↑
popanto33@gmail.com		Feb 3, 2023	Feb 3, 2023	56xat02YyFUMHIO22vB3Fon9z593
antopopa22@hotmail.com		Feb 3, 2023	Feb 3, 2023	Zq133eq1G40Bh2PKm3d4WbRs7x...
<div> Rows per page: 50 1-2 of 2 < > </div>				

Fig. 3.9 FirebaseAuthification

Conform Fig. 3.9 , putem observa că utilizatorii au fost adăugați în baza de date. De asemenea se poate observa codul unic al fiecărui utilizator, data la care s-a înregistrat în sistem precum și ultima lui conectare.

3.4.2 FirebaseDatabase

Pentru a stoca și sincroniza datele am creat baza de date, numită licență, care conține venituri, categorii bugetele, categorii cheltuielile și obiective, reprezentate în Fig. 3.10



Fig. 3.10 FirebaseDatabase - Bază de date



Fig. 3.11 Firebase - Cheltuieli

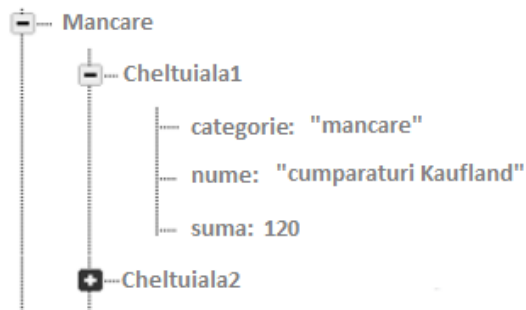


Fig. 3.12 Firebase - Conținut cheltuieli

Din capturile de ecran reprezentate în figurile de mai sus se poate observa structura arborescentă a datelor. Astfel avem un nod principal numit licență unde se află toate datele utilizatorului. În nodul Cheltuieli sunt inserate categoriile: Mâncare, Transport, UzPersonal, Haine și Altele. Expandând nodul Mâncare regăsim cheltuielile efectuate corespunzătoare acestei categorii. Fiecare cheltuială în parte include date precum categoria din care face parte, denumirea acesteia și suma aferentă.

4.Implementare, utilizare și testare

Pentru implementarea aplicației am avut nevoie de următoarele instrumente: AndroidStudio ca mediu de dezvoltare și platforma Firebase ca mediu de stocare.

Primul pas în dezvoltarea aplicației a fost crearea unui proiect nou în Android Studio cu versiunea Android 7.0 (Nougat) și un API minim de 24, astfel încât aplicația are posibilitatea de a rula pe 95.4% din dispozitive. După acest prim pas, am decis că este necesar să conectez proiectul din AndroidStudio la platforma Firebase. Astfel că, am intrat pe platforma Firebase unde mi-am făcut un cont și mi-am creat un proiect nou. Pentru a adauga acest proiect în AndroidStudio am avut nevoie de numele fișierul generat de platformă care este de tip json. Mai apoi, a fost necesară setarea câtorva dependențe pentru a realiza conectarea și accesul la serviciile oferite de platformă.

În primă fază, m-am ocupat de crearea layout-urilor pentru conectarea și înregistrarea utilizatorilor. Activitățile corespunzătoare acestor două layout-uri sunt ConectareActivity și ÎnregistrareActivity. Aplicația se deschide cu pagina de conectare unde se află un cardView ce conține două câmpuri implementate sub formă de textInputEditText și butonul “Conectare”, iar în afara cardView-ului se găsește butonul “Înregistrare”. Pe baza componentelor din layout am trecut la crearea claselor aferente.

În momentul în care este apăsat butonul Conectare, se face mai întâi o verificare dacă câmpurile au fost completate. Dacă nu au fost completate, utilizatorul este înștiințat printr-un toast că trebuie să completeze datele. În schimb, dacă datele au fost completate se face o validare cu ajutorul FirebaseAuthification, iar dacă datele corespund cu cele introduse, utilizatorul poate accesa meniul principal al aplicației. Dacă se înregistrează un insucces în ceea ce privește regasirea datelor, utilizatorul este avertizat că datele introduse nu sunt corecte și este rugat să-și introducă din nou datele.

La apăsarea butonului Înregistrare, se deschide fereastra aferentă activității ÎnregistrareActivity. Utilizatorul introduce date precum: numele, adresa de email și parola, iar la apăsarea butonului “Înregistreză-mă” acestea sunt salvate în baza de date. În cazul în care utilizatorul nu completează toate câmpurile, acesta va fi înștiințat că trebuie să completeze toate datele. Odată cu apăsarea butonului, utilizatorul este redirecționat în pagina de conectare unde trebuie să-și introducă datele create anterior.

După ce m-am asigurat că înregistrarea și conectarea funcționează cum trebuie , am trecut la construirea meniului lateral. Acesta este creat pe baza `NavigationViewDrawerLayout` și permite navigarea printre paginile aplicației, unde utilizatorul poate vizualiza pagina Acasa, lista de venituri, lista de bugete alocate, lista de cheltuieli, obiectivele setate, precum și graficul asociat cheltuielilor introduse, pe categorii.

După efectuarea conectării, utilizatorul intră direct în pagina Acasa, al cărei layout-ul este compus dintr-un `textView` prin care se întâmpină utilizatorul cu un salut și un `cardView` unde sunt dispuse date importante pentru utilizator precum totalitatea veniturilor, cheltuielilor și bugetul rămas, care se actualizează pe măsură ce sunt introduse date. De asemenea, există și patru butoane `AdaugaVenituri`, `SeteazăBugetele`, `AdaugăChetuieli` și `AdaugăObiective`, fiecărei corespunzând unei activități diferite. Aici am implementat o condiție foarte importantă în ceea ce privește funcționalitatea aplicației conform căreia dacă utilizatorul nu înregistrează măcar un venit, acesta nu poate seta ceilalți parametri, butoanele fiind dezactivate. Vom lua acum fiecare activitate în parte și vom discuta modul de realizare și funcționalitatea acestora.

`AdaugăVenituriActivity` se deschide în momentul în care utilizatorul apasă butonul de adăugare venituri și conține date precum proveniența și valoarea venitului înregistrat. Prin apăsarea butonului `adaugăVenit` corespunzător acestei activități, se realizează salvarea în baza de date, reactualizarea listei de venituri din meniul lateral și închiderea activității cu revenire în pagina acasa. Pentru salvarea în baza de date am utilizat clasa utilă `Venituri` pe baza căreia am creat obiectele. După aceea am obținut o referință a locului unde am vrut să salvez datele cu ajutorul metodei `getReference()`, iar într-un final am folosit metoda `setValue()` pentru a salva obiectul în `Firebase` la locația specifică. Actualizarea listei a presupus să accesez referința corespunzătoare listei de venituri și crearea unui ascultator `childEventListener` pentru a putea monitoriza adăugările. Astfel prin metoda `onChildAdded()` am obținut datele preluate din `DataSnapshot` și le-am utilizat pentru a actualiza lista din interfața grafică corespunzătoare veniturilor. Pentru conceperea listei a fost nevoie să utilizez un adapter personalizat pentru a regla aspectul acesteia astfel încât să avem o vizualizare cât mai clară.

`AdaugăCheltuieliActivity` are în componență un spinner pentru alegerea categoriei din care face parte cheltuiala, pe baza căruia am realizat inserarea în baza de date. Astfel ca referința locului în care a trebuit să fie adăugată cheltuiala a fost condiționată de această categorie. Pe lângă acest spinner există și două `textInput`-uri pentru a introducerea denumirii și sumei. Pentru

salvarea în lista cheltuielilor am făcut aceeași pași ca în cazul listei de venituri. În plus, pe măsură ce se introduc cheltuieli, am actualizat graficul aferent cheltuielilor pe categorii. Mai înăi am creat graficul cu ajutorul funcțiilor din biblioteca MPAndroidChart. Pentru a realiza actualizarea graficului am avut nevoie să parsez datele din DataSnapshot și să le colectez într-un HashMap deoarece mi-am dorit generarea graficului pe categorii de cheltuieli unde cheia va fi reprezentată de categorie, iar valoarea de suma cheltuită pentru acea categorie. Aastă valoare a fost calculată în prealabil de către mine. Lista finală am trimis-o ca parametru pentru a genera graficul pe baza acestor date.

În cazul activităților AdaugăObiective și SeteazăBugete am urmat același procedeu ca la AdaugăVenituriActivity.

Pentru a șterge elemente din listă a fost nevoie să realizez reactualizarea fragmentul de care ține elementul precum și baza de date. Pentru a șterge elementul din baza de date am accesat elementul care trebuia șters prin metoda child() urmând să apelez metoda removeValue(). Reactualizarea fragmentului a presupus totodată reactualizarea listei, iar pentru acest lucru m-am ajutat de metoda onDataChange() pentru a obține datele actualizate din DataSnapshot, iar pentru actualizarea adapterului am folosit metoda notifyDataSetChanged(). Prin implementarea acestei metode am reușit să actualizez lista de fiecare dată când șterg din baza de date.

Pentru a putea observa efectele produse de implementare am rulat aplicația, extrăgând câteva părți din aceasta pentru a exemplifica procesul de testare și în același timp de utilizare a aplicației. Acestea vor fi prezentate în rândurile ce urmează.

După cum putem observa în Fig. 4.1, la deschiderea aplicației apare fereastra de conectare. În Fig. 4.2 se introduce un utilizator care are deja un cont, însă parola acestuia este greșită și se afișează mesajul de eroare corespunzător.

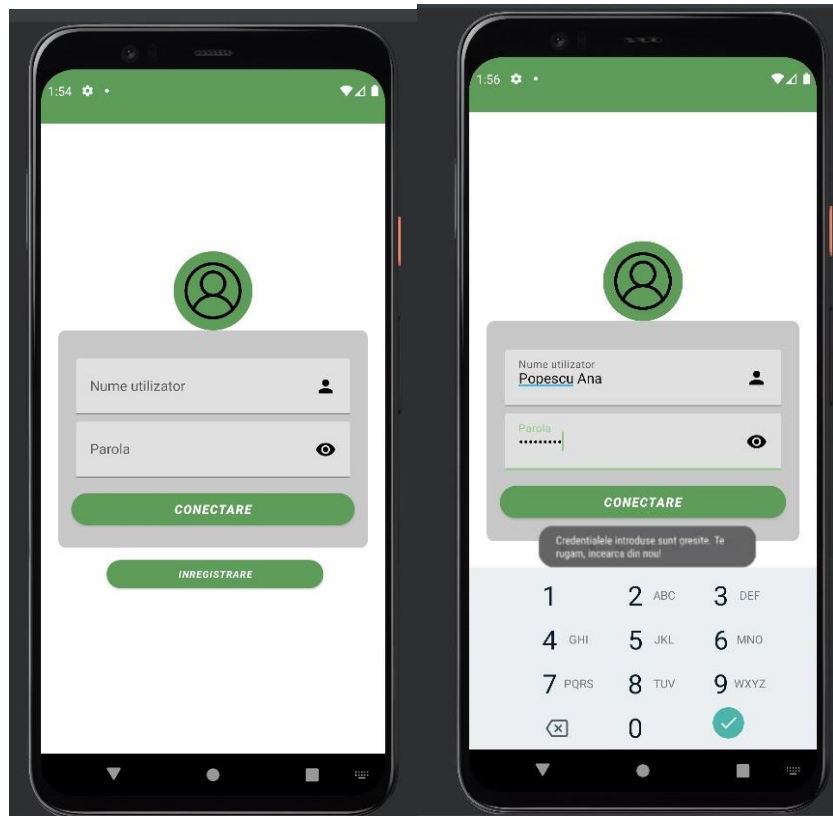


Fig. 4.1 Conectare utilizator

Fig. 4.2 Conectare cu eroare



Fig. 4.3 Înregistrare utilizator

În Fig. 4.3 este reprezentată fereastra de înregistrare în aplicație, iar dacă datele sunt introduse în fiecare câmp, utilizatorul va reveni înapoi în fereastra de conectare de unde va avea acces la aplicația propriu zisă prin completarea câmpurilor cu datele introduse în fereastra anterioara.



Fig. 4.4 Pagina Acasa

Fig. 4.4 ilustrează faptul că utilizatorul Popescu Ana a reușit să se conecteze în aplicație parametrii reprezentați de totalul veniturilor, cheltuielilor și bugetul rams fiind inițial setați pe zero. Acești parametri se modifică în funcție de datele adaugate. Dupa cum am spus și mai sus, butoanele aferente cheltuielilor, bugetelor și obiectivelor rămân dezactivate până când utilizatorul nu introduce un venit.

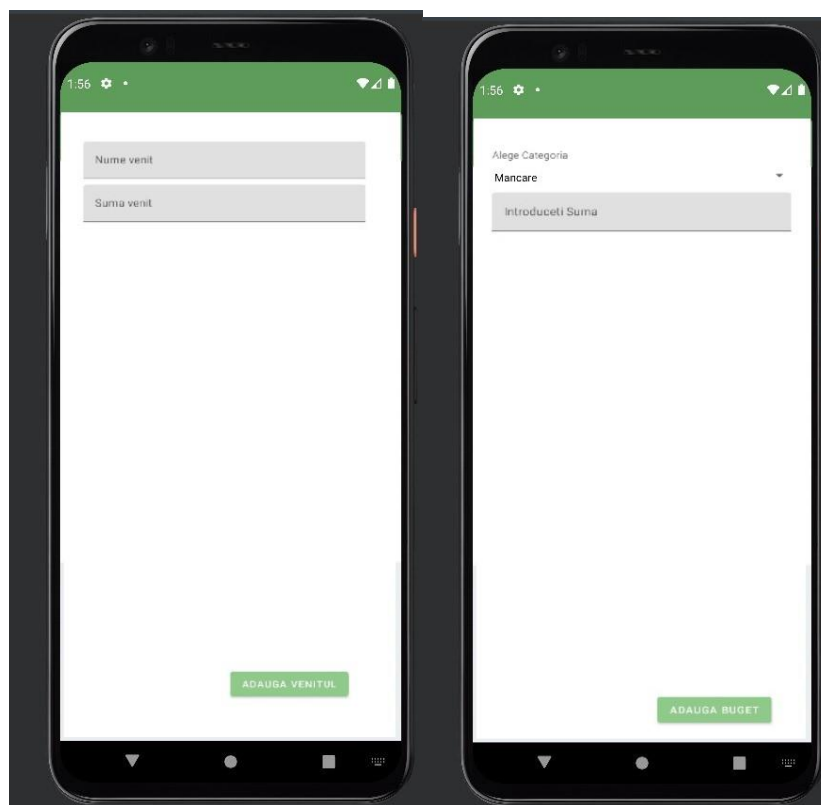


Fig. 4.5 Adaugă venituri

Fig. 4.6 Adaugă bugete

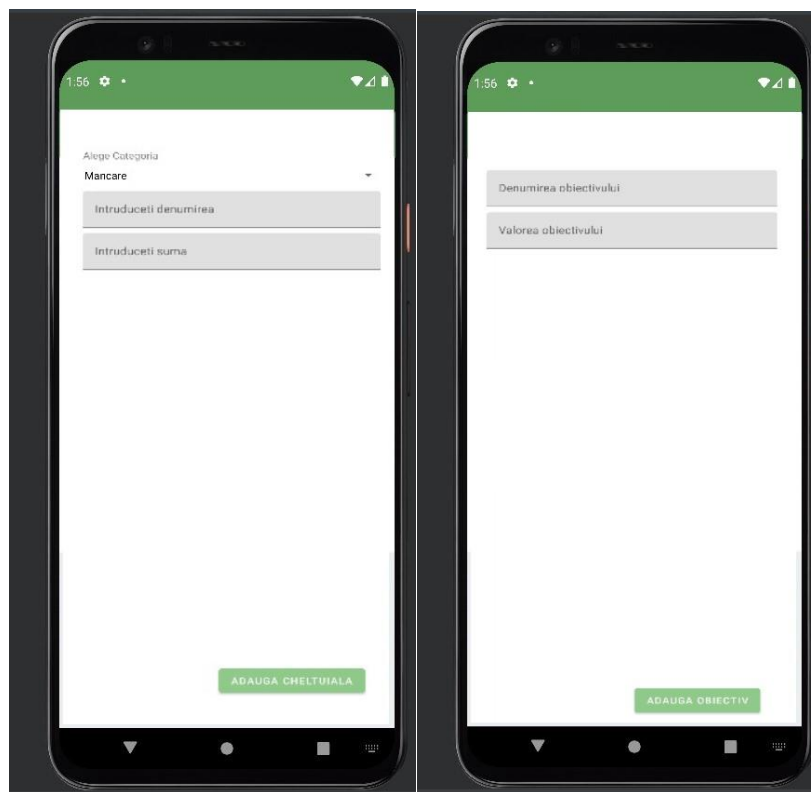


Fig. 4.7 Adaugă cheltuieli

Fig. 4.8 Adaugă obiective

Interfețele reprezentate în Fig.4.5, Fig. 4.6, Fig.4.7 și Fig. 4.8 reprezintă metodele de adăugare a veniturilor, cheltuielilor, bugetelor și obiectivelor care vor fi salvate în listele corespunzătoare.

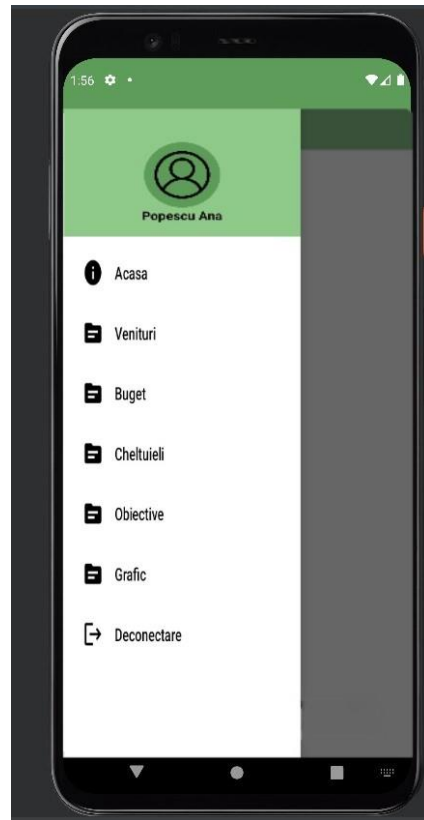


Fig. 4.9 Meniu lateral

O ilustrație a meniului lateral se află în Fig 4.9 . Acest meniul lateral permite utilizatorilor să vizualizeze listele și să șteargă elemente din acestea. Este posibilă și vizualizarea unui grafic, iar de aici utilizatorul poate să se și deconecteze. Conform Fig. 4.10, putem observa că în urma adăugării unor cheltuieli, lista se actualizează.

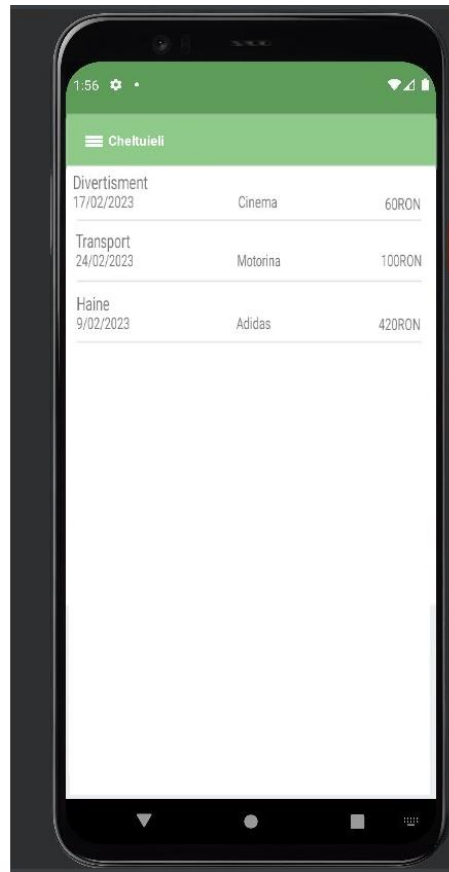


Fig 4.10 Listă Cheltuieli

Dacă dorim să ștergem un element din listă, vom merge pe elementul pe care dorim să-l ștergem și tragem spre dreapta. Astfel se generează o fereastră de dialog, reprezentată în Fig 4.11, unde utilizatorul este întrebat dacă este sigur că dorește ștergerea. Efectul acestor acțiuni se pot observa în Fig. 4.12

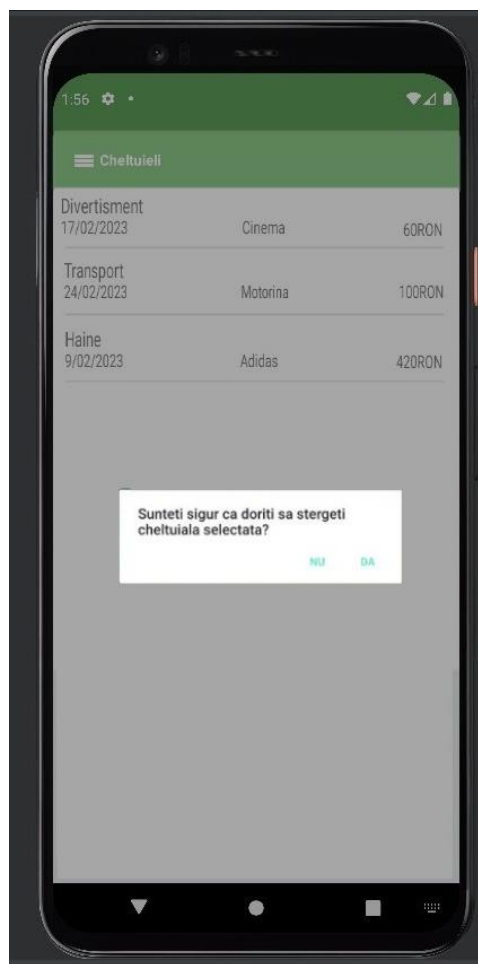


Fig. 4.11 Confirmare ștergere



Fig 4.12 Listă după ștergere

5. Concluzii

Lucrarea de față a avut ca obiectiv principal dezvoltarea unei aplicații mobile pentru gestiunea eficientă a bugetului personal, destinată tuturor persoanelor care doresc să țină evidența veniturilor și cheltuielilor în scopul atingerii obiectivelor. Gestionarea finanțelor proprii este extrem de relevantă în societatea contemporană, jucând un rol crucial în atingerea stabilității și succesului financiar individual. Prin urmare, am ales să realizez o aplicație intuitivă și ușor de utilizat, care oferă utilizatorilor posibilitatea de a-și monitoriza și controla cheltuielile într-un mod convenabil și eficient.

Pentru elaborarea acestei lucrări a fost nevoie de o documentare serioasă în ceea ce privește soluțiile deja existente. Am observat diferite funcționalități pe care le oferă aplicațiile din acest domeniu și astfel am reușit să extrag câteva lucruri de baza pe care trebuie să le facă și aplicația mea. De asemenea, am observat complexitatea aplicațiilor existente ceea ce m-a influențat în crearea unei interfețe grafice simple care să pună accentual pe funcționalitățile implementate.

Prin dezvoltarea acestei aplicații am avut în vedere să facilitez procesul de monitorizare al cheltuielilor, de planificare a bugetelor și de economisire, oferind diferite instrumente și informații necesare pentru gestiunea finanțelor personale. Printre instrumentele oferite, ce reprezintă și funcționalități ale aplicației, putem enumera: sistemul de monitorizare al cheltuielilor pe categorii care menține utilizatorul în limitele bugetului instituit de el, realizarea graficului și sistemul prin care utilizatorul este anunțat dacă își atinge obiectivul. Toate acestea influențează comportamentul financiar al utilizatorului, stimulându-l și motivându-l să nu mai facă cheltuieli neesențiale pentru a-și vedea obiectivul împlinit.

Desigur că există și multe posibilități de extindere ale acestei aplicații. Pe viitor pot fi adăugate funcționalități suplimentare precum: sincronizarea cu conturile bancare, sisteme complexe de alertare a utilizatorului și generarea unor grafice mai detaliate pe zile, luni și ani pentru a putea înregistra un progres sau un regres.

În concluzie, aplicația realizată este una ușor de utilizat și eficientă care vine cu soluții în ceea ce privește gestiunea eficientă a bugetului.

6.Bibliografie

[SBC22]-Studiu bancar-comportamentul consumatorilor.Interesul pentru economisire, mai mare in randul tinerilor, disponibil la adresa:

<https://cursdeguvernare.ro/studiu-bancar-comportamentul-consumatorilor-interesul-pentru-economisire-mai-mare-in-randul-tinerilor.html>

[BBA20]-Josh Smith,The Best Budget Apps for 2020,disponibil la adresa:

<https://www.gottabemobile.com/best-budget-apps/?gbmsl=15>

[PRW23]-Piedad Ramirez-What are the most used operating system in 2022?,disponibil la adresa:

<https://itsoftware.com.co/content/most-used-operating-systems/>

[MAS]- Meet Android Studio, diponibil la adresa:

<https://developer.android.com/studio/intro>

[OAS]-Overview of Android Studio,disponibil la adresa:

<https://www.geeksforgeeks.org/overview-of-android-studio/>

[FGA23]-Florin Groza-Ce este Android?Iată un scurt istoric și principalele caracteristici,disponibil la adresa:

<https://aplicatiiandroid.com/ce-este-android/>

[DGA19]-Daniel Grigori-Ce este Android,diponibil la adresa:

<https://www.grigdroid.ro/2016/05/ce-este-android-o-scurta-poveste.html?m=1>

[FBA]-Firebase Authentification,disponibil la adresa:

<https://www.javatpoint.com/firebase-authentication>

[FRD]-Firebase Realtime Database, disponibil la adresa:

<https://www.javatpoint.com/firebase-realtime-database>

[FBI]-Firebase Introduction, disponibil la adresa:

<https://www.javatpoint.com/firebase-introduction>

7.Anexe

Anexă 1 – Lista figurilor

Fig. 2.1 Aplicația Mint	7
Fig. 2.2 Aplicația Wally	8
Fig. 2.3 Aplicația YNB.....	8
Fig 2.4 Aplicația Mvelops	9
Fig. 2.5 Aplicația Spendee.....	10
Fig. 2.6 Cele mai utilizate sisteme de operare pentru dispozitivele mobile în 2022	12
Fig. 3.1 Diagrama cazurilor de utilizare.....	15
Fig. 3.2 Diagrama de activitate - Conectare și înregistrare	16
Fig. 3.3 Diagrama de activitate - Configurare date în aplicație.....	17
Fig. 3.4 Diagrama de clase	18
Fig. 3.5 Diagrama de interacțiune - Conectare și înregistrare	18
Fig. 3.6 Diagrama de interacțiune – Adaugare date în aplicație.....	19
Fig. 3.7 Diagrama de colaborare.....	20
Fig. 3.8 Diagrama de desfășurare	21
Fig. 3.9 FirebaseAuthification	21
Fig. 3.10 FirebaseDatabase - Bază de date.....	22
Fig. 3.11 Firebase - Cheltuieli	22
Fig. 3.12 Firebase - Conținut cheltuieli.....	23
Fig. 4.1 Conectare utilizator.....	27
Fig. 4.2 Conectare cu eroare.....	27

Fig. 4.3 Înregistrare utilizator	27
Fig. 4.4 Pagina Acasa	28
Fig. 4.5 Adaugă venituri.....	29
Fig. 4.6 Adaugă bugete	29
Fig. 4.7 Adaugă cheltuieli.....	29
Fig. 4.8 Adaugă obiective.....	29
Fig. 4.9 Meniu lateral	30
Fig 4.10 Listă Cheltuieli	31
Fig. 4.11 Confirmare ștergere.....	32
Fig 4.12 Listă după ștergere	32

Anexa 2 - Lista tabelelor

Tabel 2.1 Comparații.....	11
---------------------------	----