

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Specializarea Informatică Economică

Aplicație Android de planificare a călătoriilor

Lucrare de licență

**Coordonator Absolvent**

**Conf. univ. dr. Cristian CIUREA Ramona-Ioana BOTEZATU**

București 2018

Declarație privind originalitatea continutului și asumarea răspunderii

Prin prezenta declar că rezultatele prezentate în această lucrare sunt în întregime rezultatul propriei mele creații cu excepția cazului în care se fac referiri la rezultatele altor autori. Confirm faptul că orice material folosit din alte surse (reviste, carti si site-uri de internet) este în mod clar referit in lucrare si este indicat în lista de referințe bibliografice.

Cuprins

[**Introducere** 3](#_Toc518001733)

[**1.** **Planificarea călătoriilor** 7](#_Toc518001734)

[**1.1.** **Descrierea în detaliu a problemei abordate** 8](#_Toc518001735)

[**1.2.** **Motivaţia pe baza căreia a fost aleasă tema** 14](#_Toc518001736)

[**1.3.** **Necesitatea unei soluţii informatice ca soluţie la problema prezentată** 15](#_Toc518001737)

[**1.4.** **Actualitatea si importanţa problemei** 16](#_Toc518001738)

[**2.** **Tehnologii/Metode utilizate** 17](#_Toc518001739)

[**2.1.** **Tehnologii utilizate** 17](#_Toc518001740)

[**2.2.** **Descrierea algoritmilor, metodologiilor si modelelor utilizate** 22](#_Toc518001741)

[**3.** **Arhitectura aplicaţiei** 24](#_Toc518001742)

[**3.1.** **Diagrama de componente** 24](#_Toc518001743)

[**3.2.** **Diagrama de flux** 25](#_Toc518001744)

[**3.3.** **Descrierea funcţiilor aplicaţiei si principalele module** 27](#_Toc518001745)

[**4.** **Implementarea soluţiei** 30](#_Toc518001746)

[**4.1.** **Cazuri de utilizare pentru principalele func**ţ**ii ale aplica**ţ**iei** 30](#_Toc518001747)

[**4.2.** **Modul de implementare a celor mai interesante func**ţ**ii** 33](#_Toc518001748)

[**4.3.** **Descrierea principalelor clase implementate** 40](#_Toc518001749)

[**4.4.** **Structura bazei de date si rolul tabelelor** 42](#_Toc518001750)

[**5.** **Concluzii** 46](#_Toc518001751)

[**Bibliografie** 48](#_Toc518001752)

[Anexa 1- Harta cu locatiile vizitate de utilizator 49](#_Toc518001753)

[Anexa 2- Activitatea cu vremea actualizata in timp real 50](#_Toc518001754)

[Anexa 3- Galeria de imagini (Cover Flow) 51](#_Toc518001755)

# **Introducere**

Obiectivul principal al lucrării este reducerea timpului consumat pentru căutarea unei locaţii potrivite pentru viitoarea călătorie. De asemenea, aplicaţia vine în sprijinul utilizatorilor săi prin afişarea unor recomandări de activităţi potrivite bugetului lor.

Necesitatea dezvoltării unei aplicaţii mobile pentru planificarea călătoriilor este dată de nevoia oamenilor de a-şi organiza într-un timp relativ redus următoarea vacanţă.

În zilele noastre este foarte greu să ne imaginăm o societate fără telefoane mobile. De aproape două decenii, telefoanele au schimbat în întregime societatea în care trăim: am trecut de la hărțile pe hârtie la aplicaţii ce ne localizează în timp real şi care ne arată cea mai avantajoasă rută ca timp şi distanță, de la pozele developate la telefoane cu camere foto extrem de performante, iar timpul irosit pentru căutarea anumitor informaţii s-a redus substanţial doar prin apăsarea unui click pe un site ce conţine toate informaţiile de care avem nevoie. În prezent telefoanele au devenit o parte integrantă a vieţii noastre, fără de care ne-ar fi prea greu să ne desfăşurăm activităţile de zi cu zi. Statisticile arată o creştere impresionantă a vânzărilor de telefoane mobile de la an la an, iar acest lucru nu se va schimba prea curând, deoarece este mult prea greu să se înlocuiască ceva ce este înrădăcinat în atâtea aspecte ale vieţii noastre.

În agitația vieții moderne este important să avem un program şi să ne planificăm lucrurile pentru a ne face viaţa puţin mai liniştită. Există multe moduri care joacă un rol-cheie în aceste situaţii şi din această cauză utilizarea unui telefon mobil a devenit o necesitate, mai ales că îl putem utiliza ca un laptop în miniatură, ceea ce ne permite să lucrăm într-o continuă deplasare oriunde ne-am afla.

Două dintre lucrurile care fac aceste dispozitive atât de vitale pentru viaţa noastră de zi cu zi sunt eficienţa şi funcţionalităţile lor. Viteza cu care putem efectua sarcini de pe un telefon mobil este aproape de neegalat şi de asemenea posibilitatea de a stoca date importante precum fişiere, informaţii sau documente ne face să devenim dependenţi de el. În prezent există aplicaţii pentru aproape orice, de la magazine online la aplicaţii ce ne monitorizează stilul de viaţă.

Datorită dezvoltării continue a tehnologiei, societatea tinde să îşi atingă scopurile printr-un singur click, dar cu toate acestea există un lucru pe care oamenii nu îl pot face folosind telefonul: să călătorească. Călătoria reprezintă un drum al cunoaşterii de sine, o iniţiere în arta de a trăi, de a fi fericit, de a simţi. Există numeroase dovezi care arată faptul că fiinţa umană a fost dispusă să călătorească dintotdeauna, cum sunt jurnalele sau descrieri existente încă din antichitate. Experienţele acumulate în urma călătoriilor sunt cele care ne arată asemănările şi diferenţele dintre cultura noastră şi a celorlalti. Destinaţiile sunt diverse şi de aceea excursiile ne lărgesc aria de cunoaştere prin discrepanţele existente şi uşor de observat între locaţii, astfel că într-un loc unde modul de gândire, societatea şi stilul de viaţa sunt diferite aceste aspecte ne vor influenţa într-o mare măsură, ne vor oferi alte perspective asupra lumii. Pe scurt, ne lărgim orizontul cunoaşterii cu fiecare călătorie, ne dezvoltăm multilateral şi ne descoperim pe noi înşine.

Timpul e preţios pentru noi şi ne dorim să îl fructificăm pe cât de mult posibil, de aceea această aplicaţie are ca scop principal simplificarea planificării vacanţelor, precum şi reducerea timpului rezervat alegerii destinaţiei potrivite.

Atunci când ne dorim să luăm o pauză de la agitaţia zilnică în care trăim, cel mai bun mod de relaxare este plecarea într-o călătorie. Indiferent de durata acesteia, zilele petrecute departe de probleme ne ajută să ne liniştim şi să ne echilibrăm emoţional, dar cu toate acestea, am dori să îmbinăm plăcutul cu utilul. Mai precis, ne dorim să alegem o destinaţie care să ne fie pe plac şi în acelaşi timp să fie cuprinsă de mister şi cu cât mai multe obiective turistice. După cum spune şi proverbul “Gusturile nu se discută”, alegerea unei locaţii devine dificilă, deoarece există multe destinaţii foarte diferite între ele. Sunt locaţii exotice, ori cu un trecut istoric impresionant, chiar şi locati cu influenţe moderniste, iar căutarea destinaţiei perfecte necesită mult timp şi răbdare. De aceea, aplicaţia simplifică efortul depus pentru această căutare, oferindu-ţi cele mai bune alternative în funcţie de preferinţe, hobby-uri şi bugetul disponibil.

Cum planificarea călătoriei presupune parcurgerea unor anumiţi paşi, pregătirea bagajelor reprezintă ultimul pas şi este întotdeauna o provocare a exploratorilor. Pentru că vacanţa să decurgă cu bine, avem nevoie de anumite lucruri sau produse care ne ajută în desfăşurarea activităţilor pe perioada vacanţei. Aceste lucruri sunt deseori uitate din cauza grabei dinaintea pregătirii bagajelor. Aplicaţia are ca obiectiv secundar uşurarea împachetării pe ultima sută de metri a bagajelor, întrucât aceasta pune la dispoziţia utilizatorilor săi liste standard sau personalizate cu lucruri ce nu trebuie uitate în călătorie. De asemenea, utilizatorii îşi pot adăuga lucrurile ce doresc să le ia în vacanţă şi astfel la pregătirea bagajelor, consultarea listei evită uitarea acestor obiecte.

În capitolul 1 se prezintă problema abordată de lucrare descrisă în detaliu, motivaţia pe baza căreia a fost aleasă tema, cât şi necesitatea implementării unei soluţii informatice ca rezolvare la problema abordată.

În capitolul 2 se prezintă tehnologiile folosite în implementarea soluţiei informatice.

În capitolul 3 se prezintă arhitecura soluţiei, mai exact structura aplicaţiei, constând în diagrame de componente şi de flux cât şi descrierea funcţiilor aplicaţiei.

În capitolul 4 se prezintă modul de implementare a soluţiei informatice, descrierea claselor folosite, explicarea bazei de date şi rolul tabelelor.

# **Planificarea călătoriilor**

## **Descrierea în detaliu a problemei abordate**

Călătoriile în străinătate au devenit o modalitate foarte des întâlnită pentru a evada din agitaţia lumii moderne, doarece preţurile biletelor de avion sunt mult mai accesibile ca în trecut, iar avionul a devenit un mijloc de transport preferat de marea majoritate a oamenilor când vine vorba de călătorit. De asemenea, locaţiile ce până ieri erau văzute doar în filme sau în reclame, astăzi foarte mulţi dintre oameni au posibilitatea de a achiziţiona un bilet de avion spre o ţară europeană la un preţ accesibil. În plus societatea de astăzi se schimbă constant, iar oamenii consideră o vacanţă ca fiind esenţială, nu luxul absolut care obişnuia să fie. Călătoriile, indiferent de natura lor, sunt benefice pentru sănătatea mentală şi fizică, dar şi pentru cunoaşterea de sine şi dezvoltarea personală.

Conceptul „Există o relaţie solidă între amploarea experienţelor de călătorie în străinătate şi încrederea generală la nivelul societăţii” [1] este rezultat în urma studiului „Călătoria extinde mintea” (ro), ce presupune analiza efectului călătoriei asupra omului împărţit în cinci studii. Primul studiu examinează efectul în funcţie de numărul de ţări vizitate, dar nu şi de timpul petrecut în călătorie, al doilea şi al treilea studiu au stabilit un efect cauzal asupra încrederii de sine. Cel de-al patrulea studiu a utilizat un design longitudinal pentru a demonstra că există o strânsă corelaţie între călătoriile în ţări străine şi creşterea încrederii, iar ultimul studiu realizează o paralelă a asemănărilor şi deosebirilor dintre ţările vizitate de oameni, punând accent mai degrabă pe diferenţele dintre locaţii pentru a demonstra că încrederea generală la nivelul societăţii este influenţată de destinaţia aleasă de călător. Deşi fiecare studiu prezintă mici diferenţe, s-a dovedit că aceste călătorii în ţări cât mai diferite ca şi cultură, mentalitate, influenţe întăreşte încrederea de sine în viaţa de zi cu zi. Indiferent de tipul călătoriei, fie că este internă sau externă, acestea sunt cele mai bune investiţii pe plan personal pe care omul le poate face, căpătând astfel încredere în sine şi în ceilalţi într-un mod mai natural. În plus fiecare călătorie este o experienţă unică, iar în acelaşi timp o provocare pentru sine, întrucât atunci când oamenii călătoresc, experimentează lucruri noi, interacţionează cu alte persoane cu mentalităţi şi obiceiuri diferite de ale lor, iar toate aceste lucruri nu fac decât să le dezvolte capacităţile, să îi facă mai înţelegători, mai conştienţi de capacităţile lor şi să le schimbe perspectivele de viaţă.

În zilele noastre, experienţele călătoriei se pot trăi mult mai des, deoarece turismul a devenit foarte accesibil. Oamenii călătoresc tot timpul şi sunt permanent în căutarea locurilor ce se conturează pe preferinţele lor, dar acestea reprezintă intotdeuna un impediment în alegerea următoarei destinaţii. Întrucât există o multitudine de locaţii în lume, timpul alocat pentru găsirea locului potrivit este mult prea mare şi necesită multă navigare pe internet, citind multe blog-uri de călătorii şi păreri ale oamenilor. Astfel această încercare pentru descoperirea locului următoarei vacanţe a devenit o problemă pentru societatea de astăzi, deoarece oamenii au gusturi şi interese diferite. Locaţiile cu un trecut impresionant vor atrage oameni care sunt pasionaţi de istorie, dorind să cerceteze monumente vechi şi amprementele antichităţii vizibile şi astăzi. În schimb, locaţiile ce pun accent pe arta contemporană, îi vor atrage pe cei pasionaţi de domeniul artei. Artiștii contemporani continuă să utilizeze un vocabular variat al formelor abstracte și reprezentaționale pentru a-și exprima ideile. Este important să amintim că arta timpului nostru nu s-a dezvoltat într-un vid, mai degrabă reflectă preocupările sociale și politice ale contextului său cultural. Există şi clădiri ce au aspecte moderniste în structura lor, cum ar fi Burj Khalifa din Dubai, care este cea mai înaltă clădire din lume, cu o înălţime de 832 de metri. Anual 1.87 milioane de turişti vizitează acest turn [2], dar acest număr este mic în comparaţie cu Times Square din New York, care este cel mai vizitat loc din lume, atrăgând anual 35 de milioane de vizitatori, conform Statista.com [3].

Întrucât există o gamă atât de largă de locaţii şi foarte diferite între ele, oamenilor le este greu să se decidă. Astfel, această problemă este abordată de aplicaţie, obiectivul principal fiind generarea locaţiilor ce se conturează pe profilul fiecărui utilizator în parte, reducând considerabil timpul alocat de aceştia pentru planificarea călătoriilor lor.

Aplicaţia oferă utilizatorului posibilitatea de a alege locaţia potrivită în funcţie de preferinţele acestuia din cele şase recomandări ale aplicaţiei. Aceste recomandări ţin cont de interesele utilizatorului, întrucât acestea sunt generate pe baza unui formular ce este completat de utilizator ca prim pas în utilizarea aplicaţiei.

Formularul de interese conţine 8 întrebări cu scopul de a cunoaşte utilizatorul, preferinţele şi bugetul său disponibil alocat unei excursii. Pe baza acestor întrebări, generarea ofertelor se face în mod dinamic. Mai exact, la fiecare accesare a aplicaţiei se vor genera alte locaţii pe care utilizatorul le va putea examina pentru o viitoare călătorie. Aceste oraşe sunt salvate în baza de date, fiecare aparţinând unor anumite categorii. Atunci când utilizatorul se autentifică în aplicaţie, oraşele ce se încadrează în categoriile setate în formular, vor fi aduse din baza de date şi din acestea se vor afişa dinamic pentru a evita etalarea aceloraşi destinaţii.

Pentru ca aceste recomandări să nu fie redundante, în formularul de interese utilizatorul poate selecta opţiunea de a nu i se afişa oraşele pe care deja le-a vizitat, prin alegerea dintr-o listă a tuturor locaţiilor în care acesta a fost. Astfel se va evita repetarea locaţiilor care nu mai reprezintă un interes pentru utilizator. Ţinând cont de faptul că viaţa este volatilă, iar interesele oamenilor se pot schimba de la an la an sau chiar mai des, acest aspect a fost luat în considerare în proiectarea aplicaţiei, astfel că utilizatorul îşi poate modifica formularul oricând şi de oricâte ori doreşte, fără ca aceaste schimbări să afecteze funcţiile aplicaţiei. Astfel se vor genera locaţii din alte categorii diferite de cele setate anterior.

După cum se menţionează mai sus, oraşele vizitate de utilizator vor fi salvate în baza de date din două considerente. În primul rând pentru generarea ofertelor, deoarece locaţiile afişate utilizatorului să fie complet diferite de cele pe care acesta le-a vizitat şi în al doilea rând ca utilizatorul să ţină o evidenţă a tuturor călătoriilor sale. De asemenea, pe parcurs îşi poate adăuga alte locaţii pe care le-a vizitat între timp.

Când utilizatorul este interesat de o anumită locaţie, acesta o poate examina, mai precis poate afla informaţii utile pentru o viitoare călătorie. Aceste informaţii utile includ o descriere a oraşului, top-ul obiectivelor turistice de vizitat, sfaturi de vacanţă, recomandări de restaurante pentru diferite bugete, cât şi idei de activităţi sau excursii pe care acesta le poate face în acel oraş.

În cazul în care niciuna din locaţiile generate nu prezintă interes pentru utilizator, aplicaţia dispune de o galerie a tuturor locaţiilor existente în baza de date, fiecare dintre acestea este reprezentată de o poză specifică locaţiei. Astfel utilizatorul poate naviga prin această galerie de imagini şi poate selecta o destinaţie pentru a putea afla mai multe informaţii despre aceasta.

Când o locaţie prezintă interes pentru utilizator şi acesta doreşte să caute un bilet de avion sau un hotel, aplicaţia dispune de această opţiune. Astfel pentru consultarea preţurilor şi a disponibilităţii biletelor de avion, utilizatorul selectează data de plecare şi de întoarcere, cât şi aeroportul de unde doreşte să plece, aeroportul de aterizare va fi automat completat cu locaţia aleasă anterior din galerie sau din cele şase destinaţii generate. Aplicaţia îl va transfera pe site-ul momondo.com unde îi vor fi prezentate toate opţiunile de zbor ordonate în funcţie de preţul biletelor de avion. În cazul în care utilizatorul optează pentru consultarea disponibilităţii hotelului din destinaţia respectivă, paşii vor fi aceiaşi, excluzând completarea aeroportului de plecare. Rezultatele vor fi afişate în ordinea tarifelor pentru perioada selectată.

Tot pe baza formularului de preferinţe completat la prima autentificare în aplicaţie, ori modificat ulterior, aplicaţia oferă utilizatorului o listă cu activităţi, idei sau excursii de o zi din anumite locaţii în funcţie de bugetul său disponibil. Această listă va conţine de fiecare dată alte informaţii pentru a accentua dinamica şi personalizarea aplicaţiei, dar şi aspectul că aplicaţia se potriveşte pentru orice tip de utilizator.

Pentru a se contura ideea de personalizare a aplicaţiei, aceasta implementează un modul de împărtăşire a experienţelor a celor care utilizează aplicaţia. Odată cu evoluţia tehnologiei, comunicarea între oameni a devenit mult mai simplu de realizat, iar problema distanţei nu mai este o scuză pentru lipsa de comunicare. Reţelele de socializare au câştigat o creştere şi o popularitate uluitoare la nivel mondial, drept urmare au devenit cel mai simplu mod de a ne împărtăşi gândurile şi experienţele din viaţa de zi cu zi cu mulţi oameni în acelaşi timp, prin publicarea pozelor, filmuleţelor sau a unor articole pe pagina personală. Conform statisticilor, în primul trimestru al anului 2018, Facebook a înregistrat un total de 2.19 miliarde de utilizatori activi lunar, urmat de Youtube cu 1.5 miliarde de utilizatori, iar apoi Instagram cu 800 milioane [6]. De asemenea, în fiecare zi pe Instagram sunt publicate aproximativ 95 de milioane de poze şi aproximativ 4.2 miliarde de aprecieri la poze, iar pe Facebook se generează 4 milioane de „like-uri” pe minut şi 100 de milioane de ore de conţinut video sunt vizionate zilnic [7]. Pentru mulţi oameni aceste medii online sunt percepute ca o formă de divertisment, dar pentru alţii ca cea mai simplă modalitate de a păstra legătura cu apropiaţii. Atunci când îţi creezi un cont pe o reţea de socializare, de fapt, îţi creezi o identitate pe care o poţi personaliza şi gestiona după bunul plac. De exemplu un profil de facebook îţi permite să vezi postările celor pe care tu i-ai adăugat la lista de prieteni, doar că nu se face o filtrare a postărilor în funcţie de interesele tale. Nu poţi căuta doar anumite postări, doar din anumite categorii, deci nu există o triere a acestora. Aceeaşi problemă se aplică şi atunci când tu doreşti să împarţi din experienţele tale cu alţii. Nu toată lumea de pe reţelele de socializare are aceleaşi hobby-uri sau interese ca tine, drept urmare atunci când doreşti să împărtăşeşti din experienţele personale, acest lucru poate fi deranjant. Această problemă încearcă să fie rezolvată de aplicaţie prin implementarea unui mod mult mai organizat pentru a putea citi şi împărtăşi experienţe din călătorii.

Astfel utilizatorul aplicaţiei poate să îşi descrie experienţa pe care a avut-o într-o anumită locaţie, descriind locul, recomandând obiective, restaurante sau orice sfat de călătorie pe care îl consideră util pentru alţii. De asemenea, are opţiunea de a îşi menţiona pagina sau blog-ul personal, dacă doreşte să intre în contact cu alţi exploratori. Privind din cealaltă perspectivă, cea a celui care doreşte să alfe informaţii, utilizatorul poate citi părerilor oamenilor care au fost într-un anumit oraş, putând cu uşurinţă să intre în contact cu aceştia pentru a cere mai multe informaţii.

De asemenea, în orice călătorie pregătirea bagajelor este cu siguranţă una dintre cele mai grele părţi ale plecării în vacanţă. Majoritatea oamenilor consideră acest ultim pas o corvoadă şi îl amână cât mai mult timp cu putinţă, drept urmare se ajunge în final la o dezorganizare completă, cu o pregătire a bagajelor făcută în grabă. Înainte de a te apuca de împachetarea bagajelor trebuie să te gândeşti la ce ai nevoie pe parcursul călătoriei. Este important să ţii cont de perioada pe care o petreci în vacanţă, dar şi de anotimpul în care te afli. De asemenea, trebuie să iei în considerare şi nevoile tale personale, cum ar fi medicamente sau alte produse de care nu poţi să te lipseşti. Aşadar, aplicaţia pune la dispoziţia utilizatorilor săi liste standard cu lucruri obligatoriu de luat în călătorie, cât şi posibilitatea de personalizare a listei în funcţie de necesităţile fiecăruia. Lista generată de aplicaţie conţine lucruri obişnuite, dar obligatorii iar atunci când se pregătesc bagajele, utilizatorii pot bifa acele lucruri odată adăugate în bagaj. De asemenea au

posibilitatea de a şterge sau de a adăuga alte lucuri pe care le vor bifa sau debifa după caz. Se evită omiterea anumitor lucruri indispensabile în vacanţă.

## **Motivaţia pe baza căreia a fost aleasă tema**

Dorinţa de a călători am simţit-o încă de mică, dar s-a accentuat atunci când am aplicat la programul Work&Travel USA. Pentru o perioadă de patru luni am muncit în statul California, explorând cu fiecare ocazie locurile şi parcurile naţionale. Am intrat în contact cu oameni de diferite naţionalităţi cu care am împărtăşit bucuria acestei experienţe. Mi-am descoperit pasiunea pentru fotografie şi pe parcursul celor patru luni am imortalizat cele mai frumoase momente şi peisaje. După această experienţă, dorinţa de a călători, de a descoperi locuri noi şi de a cunoaşte cât mai mulţi oameni din ţări diferite a devenit tot mai intensă şi încerc pe cât de mult posibil să călătoresc din ce în ce mai des.

Studiind toate aplicaţiile de călătorie, am realizat că niciuna dintre ele nu este personalizată pentru fiecare om în parte în funcţie de interesele sale. Există o multitudine de astfel de aplicaţii care oferă informaţii turistice şi prin intermediul cărora îţi poţi planifica o călătorie, dar doar introducând toate datele necesare. De aceea, am ales pentru lucrarea de licenţă să creez o aplicaţie, care nu doar că îţi oferă toate informaţiile esenţiale ci şi îţi propune idei de destinaţii potrivite pentru tine.

## **Necesitatea unei soluţii informatice ca soluţie la problema prezentată**

Aplicaţia abordează trei probleme:

* Reducerea timpului pentru alegerea destinaţiei potrivite
* Împachetarea bagajelor
* Împărtăşirea experienţelor din călătorii doar cu oamenii cu aceleaşi interese

Aplicaţia a fost gândită să vină în sprijinul celor care doresc să călătorească, dar pentru care alegerea unei destinaţii este întotdeauna cea mai mare dificultate. De aceea, aplicaţia pune în prim plan interesele utilizatorilor săi şi se creează o relaţie între aceştia. Se prezintă destinaţiile ce sunt generate într-un mod dinamic şi care se conturează pe preferinţele acestuia. Pentru că locaţiile sugerate să îndeplinească cerinţele utilizatorilor, s-a implementat un formular de interese care trebuie obligatoriu completat la prima autentificare în aplicaţie, cu posibilitatea modificării ulterioare.

Astfel se ţine cont de destinaţiile pe care utilizatorul le-a vizitat în trecut, cât şi de buget şi categoriile de interes ale acestuia. Vor fi generate în mod automat şi dinamic şase destinaţii potrivite profilului fiecărui utilizator, cât şi idei de activităţi în funcţie de buget.

A doua problemă abordată de aplicaţie este cea a lucrurilor ce trebuie adăugate în bagaj înainte de o excursie. De cele mai multe ori, bagajele sunt făcute în grabă cu o zi sau chiar în ziua plecării şi, din această cauză, sunt omise multe lucruri esenţiale vacanţei. Aplicaţia rezolvă această problemă prin implementarea unor liste standard sau personalizate ce conţin toate lucrurile esenţiale ce nu trebuie uitate. De asemenea, lista poate fi modificată de utilizator, adăugând sau eliminând, după caz, obiectele ce le consideră de luat în vacanţă şi care nu există în listă sau de care se pot lipsi. Odată adăugate în bagaj, utilizatorul bifează acele lucruri şi ulterior verifică lista pentru a se asigura că tot bagajul este pregătit cu bine.

Ultima problemă abordează un aspect actual al societăţii de astăzi. După cum am prezentat în capitolul anterior, mediile online sunt foarte utilizate, iar unul din motive este cel de a împărtăşi experienţe şi evenimentele importante. Problema existentă în acest aspect este despre filtrarea persoanelor care au interese comune cu utilizatorul. Întrucât nu există încă o astfel de posibilitate, aplicaţia pune la dispoziţie un modul de poveşti. Mai precis, utilizatorul îşi poate publica experienţele din călătorii, de a sugera idei sau sfaturi de vacanţă pentru o anumită locaţie, iar acestea vor fi vizibile doar utilizatorilor interesaţi de acea destinaţie.

## **Actualitatea si importanţa problemei**

Acest planificator pentru călătorii îndeplineşte cerinţele societăţii de astăzi, nevoia de socializare, prin posibilitatea de împărtăşire a experienţelor din vacanţe şi de a intra în contact cu persoane cu interese comune. În plus, după cum bine ştim, călătoria nu mai reprezintă un lux ca altă dată, iar acum cu toţii optăm pentru o vacanţă în străinătate. Cum timpul sau resursele nu ne permit să plecăm în fiecare lună, pentru a alege următoarea destinaţie trebuie să alocăm foarte mult timp. Aplicaţia este cel mai uşor şi util mod de a reduce considerabil acest timp alocat căutării, deoarece totul se bazează pe un formular de preferinţe completat la prima autentificare în aplicaţie.

# **Tehnologii/Metode utilizate**

## **Tehnologii utilizate**

Aplicaţia mobilă rulează doar pe telefoanele ce au ca sistem de operare Android.

Android-ul este un sistem de operare, cât şi o platformă software, dezvoltate de Google pentru smartphone-uri sau tablete, adică dispozitive cu touch screen.

Android permite dezvoltatorilor, care doresc să creeze aplicaţii mobile, să utilizeze în mod gratuit platforma, să scrie cod Java, controlând dispozitivul prin intermediul librăriilor Java dezvoltate de Google. De asemenea, există şi o altă modalitate de a dezvolta aplicaţii mobile, utilizând alte limbaje de programare, dar compilarea acestora se face doar prin cod maşină ARM, care nu este sprijinit de către cei de la Google.

Limbajul de programare ce stă la baza dezvoltării aplicaţiilor mobile Android este Java, lansat în anul 1995 de SUN Microsystems. Acest limbaj este orientat-obiect şi cu toate că împrumută multe aspecte din sintaxa de la C/C++, acesta prezintă un model al obiectelor mult mai simplu.

Kit-ul de dezvoltare software (Android SDK) conţine un set complet de instrumente, care include programe de depanare, emulator de dispozitiv, biblioteci, tutoriale, mostre de cod şi documentaţie, cât şi JDK, kit-ul de dezvoltare Java. [8]

Cei mai mulţi dintre programatori folosesc Android Studio pentru dezvoltarea aplicaţiilor mobile, întrucât acesta este un mediu oficial de dezvoltare integrat (IDE) pentru sistemele de operare Google Android. A fost construit folosind IntelliJ IDEA de la JetBrains şi în 2014 a fost lansată prima versiune stabilă. A fost proiectat special pentru dezvoltarea Android-ului. În altă ordine de idei, Android Studio reprezintă un înlocuitor pentru Eclipse Android Development Tools ca IDE primar pentru dezvoltarea aplicaţiilor Android native [9].

Pentru dezvoltarea unei aplicaţii Android folosind platforma menţionată mai sus, avem 5 componente esenţiale, precum:

1. Activitate(Activity)

Reprezintă interfaţa cu utilizatorul sub forma unei ferestre sau a unui formular. O aplicaţie poate avea mai mult de o activitate, dar fiecare activitate un ciclu de viaţă propriu, care este independent de ciclul de viaţă al întregului proces al aplicaţiei. O altă caracteristică importantă a activităţii este reprezentată de faptul că aceasta are propria sa stare, adică datele şi informaţiile pot fi restaurate şi salvate independent.

1. Intenţie(Intent)

Reprezintă un obiect ce permite aplicaţiei să coordoneze funcţiile unor activităţi diferite pentru a atinge o anumită sarcină, mai precis reprezintă o operaţiune ce urmează să fie executată. Intent-ul poate fi un mesaj sau un obiect transmis către o altă activitate. Este asemănător cu event-handler din .Net sau Java.

1. Serviciu(Service)

Reprezintă un task care rulează în fundal, fără a interacţiona direct cu utilizatorul.

1. Furnizor de conţinut (Content provider)

Reprezentat ca o subclasă a clasei ContentProvider, acesta furnizează date la cerere unei alte aplicaţii, iar aceste cereri sunt tratate de metodele din clasa ContentResolver. Datele pot fi salvate într-o bază de date, fişiere sau reţea, întrucât acest furnizor de conţinut se comportă ca o bază de date în care se poate interoga, edita, adauga sau şterge prin utilizarea metodelor insert(), delete(), update(), respectiv query(). De asemenea, furnizorul de conţinut încapsulează informaţiile şi oferă mecanisme pentru securizarea datelor, dar şi o modalitate de a partaja datele stocate cu alte aplicaţii

1. Broadcast receiver

Reprezintă un mecanism asemănător cu conceptul de eveniment de sistem, doar că aceste broadcast-uri permit altor aplicaţii să cunoască datele care au fost descărcate pe dispozitiv şi sunt disponibile pentru a fi utilizatae. Deci acestea interceptează aceste comunicări şi iniţiază acţiuni corespunzătoare. [10]

De când tehnologia a evoluat există un permanent război între dispozitivele IOS şi Android, fiecare dintre aceste corporaţii se lupă pentru a fi lider pe piaţă, deţinând un procent cât mai impresionant. Împreună reprezintă mai mult de 99% în vânzările de smartphone-uri, dar separat există o concurenţă foarte uşor de observat. Întrucât pe piaţă sunt mult mai multe dispozitive ce utilizează Android ca sistem de operare, cum ar fi Samsung, HTC sau Google Nexus, aceştia ocupă un procent mai mare decât Apple Inc, după cum se poate observa şi în Figure 1.

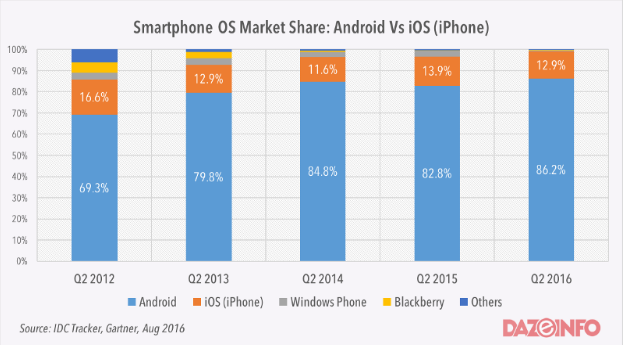


Figure 1. Utilizatori Android vs IOS (procente)

Sursa: <https://android.jlelse.eu/apple-vs-android-a-comparative-study-2017-c5799a0a1683>

O altă tehnologie folosită pentru dezvoltarea aplicaţiei, mai exact pentru crearea şi gestionarea bazei de date este MySQL. Aceasta a fost produsă de compania MYSQL AB şi distribuit sub Licenţa Publică Generală GNU. Este cel mai popular sistem pentru manipularea bazelor de date relaţionale, cu ajutorul căruia se pot construi aplicaţii în orice limbaj major (JAVA, C#, PHP).

Pentru a putea vizualiza tabelele existente în baza de date s-a folosit MYSQL Workbench, care e un instrument vizual unificat ce oferă posibilitatea modelării datelor, dezvoltarea SQL, crearea clonelor, cât şi instrumente de administrare pentru configurarea serverului.

Ca şi celelalte platforme existente, Android permite implementarea şi folosirea API-urilor pentru dezvoltarea aplicaţiilor mobile.

Application Programming Interface(API) reprezintă o colecţie de subprograme şi unelte pentru dezvoltarea software. Odată integrate în aplicaţie, acestea îţi oferă acces la clase şi metode care pot apela serviciile sistemului de operare.

În aplicaţie au fost folosite următoarele API-uri:

* **Google Maps API** care adaugă hărţile Google şi încarcă locaţiile după anumite coordonate. De asemenea, cu ajutorul acestui API, aplicaţia îi oferă posibilitatea utilizatorului de a-şi vizualiza toate locaţiile vizitate prin markere puse automat pe hartă.
* **The Places API** care oferă posibilitatea căutării de informaţii dintr-o anumită locaţie în funcţie de categorii, cum ar fi: restaurante, hoteluri, farmacii, bănci etc. De asemenea, conform localizării dispozitivului în timp real, acesta va afişa toate locaţiile precizate aflate în apropiere.
* **Whether Api** care afişează condiţiile meteo, cât şi ora locaţiilor selectate, actualizând informaţiile o dată la 30 de minute.
* **Mail API** cu ajutorul căruia aplicaţia poate trimite email-uri utilizatorului.
* **FaceBook LogIn API** care oferă posibilitatea de autentificare rapidă a utilizatorului, folosind datele de log-are pentru Facebook.

## **Descrierea algoritmilor, metodologiilor si modelelor utilizate**

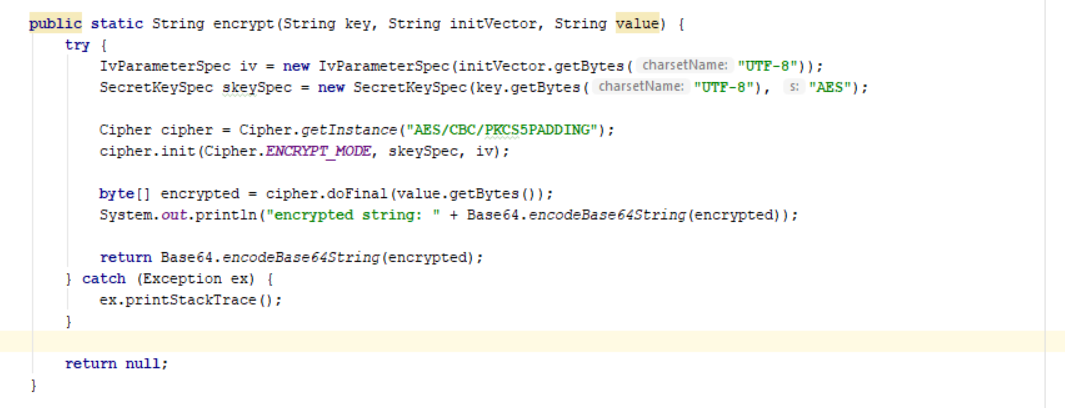
 Către toate aplicaţiile existente pe piaţă se transmit zilnic informaţii confidenţiale ale utilizatorilor, cum ar fi datele de identificare ale cardurilor de credit/debit. Pentru a putea securiza informaţiile stocate ori transmise se va folosi criptografia. Aceasta este ştiinţa care utilizează matematica pentru a cripta şi a decripta datele. Criptarea datelor se face cu ajutorul unui algoritm şi a unei chei de criptare [12].

Figure 2. Metoda de criptare a datelor

În figura 2 este prezentată metoda de criptare a datelor folosită de aplicaţie pentru criptarea parolelor. Aceasta primeşte 3 parametri: cheia de criptare, vectorul iniţial şi valoarea. Într-o variabilă de tip IvParameterSpec, care semnifică un vector de iniţializare în care se salvează noua valoare a parametrului a cărui caractere sunt codificate într-o secvenţă de octeţi.

În acelaşi mod se creează şi un obiect de tip SecretKeySpec. Această clasă specifică o cheie secretă într-un mod independent de furnizori şi poate fi folosită pentru a construi un SecretKey dintr-un tablou de octeţi.

Clasa Cipher oferă funcţionalitatea unui cifru criptografic pentru criptare şi decriptare. Pentru a crea un obiect Cipher, aplicaţia apelează metoda GetInstance a acestei clase prin care transmite numele pentru transformarea specificată. Transformarea este descrisă printr-un şir de caractere ce descrie operaţia sau setul de operaţii ce urmează să fie realizată pe datele de intrare pentru a produce un anumit rezultat şi care include întotdeauna numele unui algoritm criptografic, cum ar fi AES, DES, CBC etc. Se iniţializează obiectul de tip Cipher, specificându-se modul operaţiei: DECRYPT sau ENCRYPT.

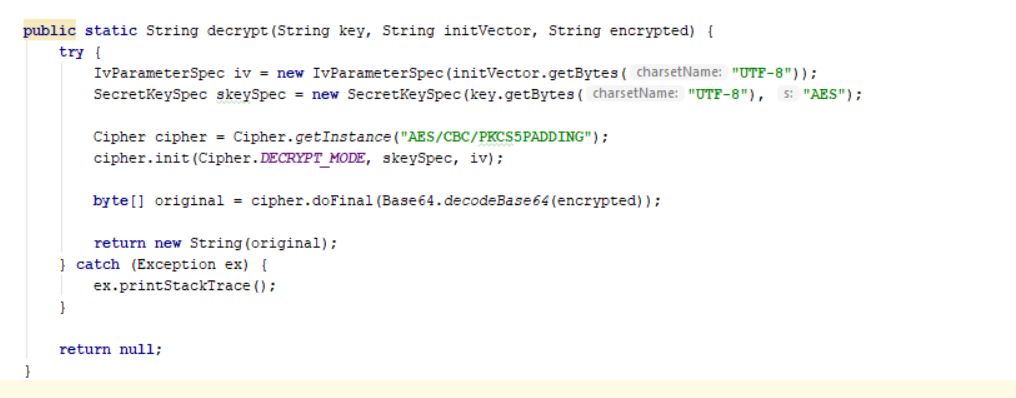
Într-o variabilă de tip byte se păstrează rezultatul apelării metodei doFinal, ce are rolul de a cripta datele şi astfel această variabilă va fi returnată şi salvată ulterior în baza de date în tabela Utilizator în câmpul parolă, astfel datele importante ale utilizatorilor sunt securizate.

Figure 3. Metoda de decriptare a datelor

În figura 3 este prezentat algoritmul pentru decriptarea parolei. Cele două metode sunt similiare, diferenţa se face la metoda de decriptare, întrucât obiectul de tip Cipher va fi iniţializat pentru modul de decriptare, iar valoarea salvată în parametrul original va returna valoarea originală înainte de criptare.

# **Arhitectura aplicaţiei**

## **Diagrama de componente**

Diagrama de componente este o parte integrantă a construirii sistemului software, deoarece ajută echipa să înţeleagă structura sistemelor existente şi apoi să construiască altele noi. Structura fizică a codului este prezentată prin această diagramă, realizându-se o mapare la view-ul componentelor şi arată relaţia dintre diferitele componente ale sistemului. Diagrama compune sistemul actual aflat în curs de dezvoltare pe diferite niveluri de funcţionalitate şi descrie componentele utilizate în implementarea funcţionalităţilor sistemului, dar nu le şi descrie.

În UML, termenul de “componentă” se referă la modul de clase, care reprezintă sisteme sau subsisteme, mai exact o parte modulară a unui sistem care încapsulează conţinutul.

Componenta poate fi realizată în 3 moduri:

1. Un dreptunghi cu numele componentei
2. Un dreptunghi cu pictograma componentei
3. Un dreptunghi cu stereotipul şi/sau pictograma

Componentele includ documente, executabile, tabele de baze de date, biblioteci, fişiere ce sunt sunt necesare sistemului şi mai ales pentru executarea funcţiilor.

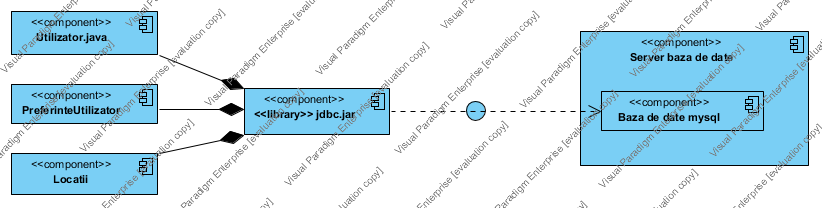


Figure 4. Diagrama de componente

## **Diagrama de flux**

Pentru reprezentarea vizuală a secvenţelor dintr-un proces, mai exact momentul de început şi de sfârşit al acestuia, cât şi paşii importanţi aplicaţi pentru obţinerea unui rezultat, se foloseşte diagrama de flux.

Numindu-se şi diagrama de activitate, aceasta descrie aspectele dinamice ale unui sistem informatic, la diferite nivele: de la “business process” până la nivelul de operaţie a unei clase. Din această cauză se folosesc multe simboluri, fiecare reprezentând o caracteristică importantă.

Acţiunile într-o diagramă de flux reprezintă un singur pas în cadrul unei activităţi, care e reprezentată sub forma unui flux coordonat pe aceste acţiuni.

Elementele unei diagrame de activitate:

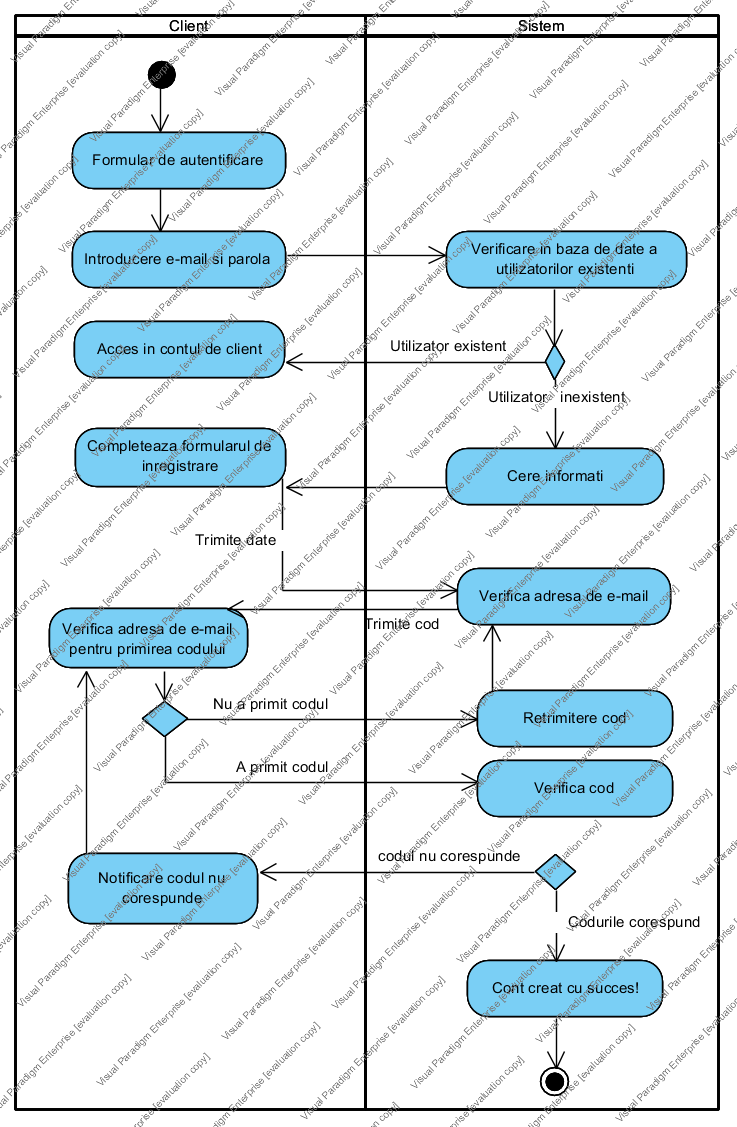
1. Noduri:
   1. Nod iniţial: punctul de inceput al diagramei
   2. Nod final:
      1. Nod final al activităţii
      2. Nod final al fluxului
2. Fluxuri si noduri

Figure 5. Diagrama de activitate pentru autentificarea in aplicaţie

* 1. Flux de control
  2. Flux de obiecte
  3. Condiţie tranzitorie

În figura 5 este prezentat fluxul activităţii de autentificare în aplicaţie a utilizatorului. Acesta completează datele de log-in regăsite în formularul de autentificare, iar odată ce acestea au fost completate se va verifica existenţa utilizatorului în baza de date. Dacă utilizatorul există în baza de date, atunci se va autentifica cu succes, iar în caz contrar acestuia i se va cere să completeze formularul de înregistrare, completându-l cu datele personale. După completarea formularului, utilizatorul va primi pe adresa de e-mail un cod de confirmare pe care, ulterior, trebuie să îl verifice în aplicaţie. Dacă cele două coduri coincid atunci adresa de e-mail este validă, iar contul este creat cu succes şi utilizatorul se poate autentifica în aplicaţie. În cazul în care codurile nu corespund sau utilizatorul nu a primit pe adresa de e-mail niciun cod de confirmare, acesta poate solicita retrimiterea codului, urmând să execute aceiaşi paşi prezentaţi anterior.

## **Descrierea funcţiilor aplicaţiei si principalele module**

La baza dezvoltării aplicaţiei stă programarea modulară, reprezentând o paradigmă de programare, grupând aplicaţia pe module. Cu ajutorul acestei programări, unitățile de cod oferă funcţionalităţi specifice (funcţii, proceduri, metode) grupate în module distincte de cele mai multe ori numite biblioteci, cataloage sau clase. Se foloseşte programarea modulară deoarece astfel putem să reutilizăm modulele pentru o organizare mult mai practică şi o structurare mai eficientă a codului sursă.

Aplicaţia este împarţită in 3 module:

1. Generare oferte

2. Publicare poveşti din călătorii

3. Împachetarea bagajelor

Toate cele 3 module sunt conturate pe nevoile oamenilor din secolul XXI. Utilitatea unei astfel de aplicaţii a apărut în momentul în care oamenii rămâneau cu foarte puţin timp dedicat nevoilor personale şi de aceea cele 3 module abordează într-un mod deosebit problema timpului.

Primul modul abordează problema timpului de căutare pe care oamenii îl acordă pentru planificarea următoarei călătorii. Practic, aplicaţia verifică la fiecare autentificare, dacă utilizatorul şi-a completat formularul de interese. În cazul în care acesta nu a fost completat, utilizatorul este obligat să-l finalizeze pentru a putea stabili ce tip de destinaţie îl interesează. Acest formular este salvat în baza de date şi poate fi actualizat oricând, utilizatorului pundandu-i-se la dispoziţie o opţiune care se regăseşte în ecranul “Profil”.

Pe baza formularului de interes completat, aplicaţia ştie să caute ofertele perfecte pentru utilizatorul curent. La fiecare accesare a activităţii “acasă”, aplicaţia pune la dispoziţia utilizatorului patru oferte conform intereselor sale, cât şi două recomandări din partea aplicaţiei. Acest lucru este posibil printr-un request în baza de date, bazat pe alegerile anterioare ale utilizatorului activ. Atât diversitatea ofertelor cât şi multitudinea acestora reprezintă puncte forte ale aplicaţiei.

Accesarea fiecărei oferte generează informaţii diverse despre acel oraş. Începând de la temperatură şi ora, informaţii aduse cu ajutorul unui API, care face câte un request la fiecare jumătate de oră, până la cele mai bune locuri unde poţi mânca sau unde poţi călători, informaţii ce se găsesc în baza de date.

Totodată, este posibilă şi căutarea unor oferte de zboruri sau hoteluri într-o perioadă aleasă de utilizator, acest lucru având ca rezultat afişarea zborurilor şi hotelurilor de la diferiţi “distribuitori”. În această parte a aplicaţiei, utilizatorul are posibilitatea de a-şi selecta oraşul de plecare, oraşul destinaţie cât şi perioada călătoriei, fiind ulterior redirecţionat pe site-ul momondo.ro, care îi va oferi acestuia cele mai bune oferte ordonate ascendent în funcţie de preţ.

De asemenea, dacă nici măcar o ofertă nu prezintă interes pentru utilizator, acesta poate naviga prin lista completă de locaţii, afişată sub forma unei galerii de imagini specifice fiecărui oraş, disponibilă din activitatea “Explore”. Utilizatorul îşi poate alege o locaţie care nu se află în cele şase destinaţii generate de aplicaţie în activitatea “acasă”. Toate aceste locaţii sunt aduse din baza de date, iar avantajul folosirii unei baze de date, constă în faptul că pot fi adăugate locaţii noi, fără ca aplicaţia să fie modificată sau actualizată.

Cel de-al doilea modul vine în încurajarea exprimării libere a oamenilor de a îşi împărtăşi experienţele trăite în vacanţe. Utilizatorul îşi poate publica opinia/experienţa despre unul din oraşele vizitate, poveste ce poate fi vizibilă în ecranul “Vezi poveşti”. Aici o listă cu toate poveştile este afişată pentru fiecare oraş în parte. De asemenea, utilizatorul poate să îşi publice datele personale, cum ar fi numărul de telefon, adresa de email sau profilul de pe o reţea de socializare pentru a putea comunica cu alţi utilizatori. Odată puse la dispoziţie aceste informaţii, în partea drepta a elementelor din listă va apărea un buton pentru a face vizibile datele personale a utilizatorului care a publicat povestea.

Cel de-al treilea modul este gândit tot pentru utilizator şi pentru a rezolva cea mai mare problemă când vine vorba de călătorii: bagajele. Pe baza unei liste standard, salvată în baza de date, ce conţine lucruri obligatorii de luat, utilizatorul poate alege ce produse au fost adăugate în bagaj şi de asemenea, poate să îşi adauge în listă obiecte preferenţiale, pe care ulterior le poate şterge. Astfel, pregătirea bagajelor nu o să mai fie o problemă, întrucât acesta poate consulta lista în timpul împachetării. Aplicaţia gestionează aceste liste la nivelul bazei de date pentru fiecare utilizator în parte.

În plus, aplicaţia implementează o funcţionalitate ce îl înştiinţează pe utilizator în legătură cu bagajul. Utilizatorul setează data plecării în excursie, iar conform acestei date se calculează timpul rămas şi utilizatorul o să primească o notificare cu obiectele din listă pe care acesta nu le-a bifat înainte de această dată.

# **Implementarea soluţiei**

## **Cazuri de utilizare pentru principalele func**ţ**ii ale aplica**ţ**iei**

Pentru a descrie destinaţia funcţională a sistemului informatic, mai exact ceea ce va executa în procesul de funcţionare se foloseşte diagrama cazurilor de utilizare. Aceasta reprezintă un model pentru procesul de proiectare şi exploatare, reprezentând relaţiile dintre cazurile de utilizare care descriu sistemul.

Modelul cazurilor de utilizare foloseşte un mijloc de comunicare între client şi analist pentru determinarea cerinţelor proiectului şi a contextului sistemului în fazele iniţiale ale dezvoltării, mai exact definesc comportamentul unui sistem fără a preciza modul de realizare al comportamentului. În timpul dezvoltării, modelul validează arhitectura sistemului şi de asemenea, se foloseşte pentru testarea sistemului executabil.

Elementele principale ale diagramei sunt:

* Actori
* Cazurile de utilizare

Actorul comunică cu sistemul în contextul unui caz de utilizare, fiind reprezentat de orice entitate externă cu care acesta comunică pentru atingerea unor scopuri şi rezolvarea anumitor probleme.

Cazurile de utilizare sunt reprezentate printr-o elipsă care conţine numele sau denumirea prescurtată a acţiunii la nivelul acelui caz. Scopul acestora este să determine fragmentul de comportament al actorului fără a explica structura internă.

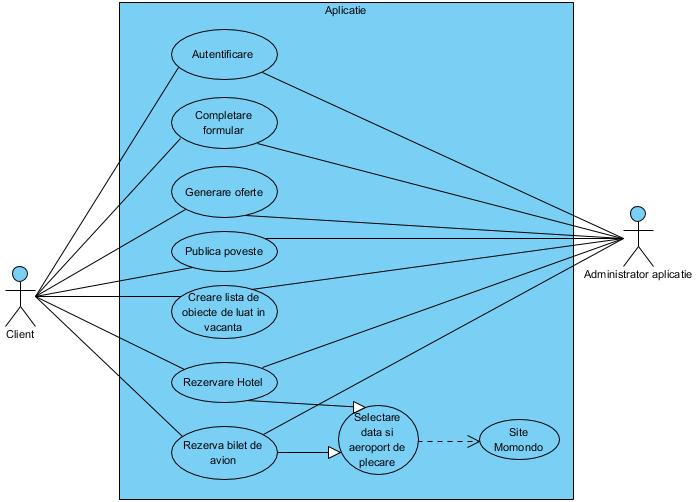


Figure 6. Diagrama cazurilor de utilizare

După cum se poate observa în figura 6, diagrama cazurilor de utilizare este reprezentată prin doi actori: clientul şi administratorul aplicaţiei şi avem următorul scenariu:

Actorul:

* se autentifică în aplicaţie
* completează formularul la prima log-are în aplicaţie
* i se generează oferte de către aplicaţie
* publică poveste
* creează lista cu obiectele esenţiale pentru vacanţă
* Poate căuta bilete de avion şi hotel pentru o destinaţie aleasă, fiind automat transferat pe site-ul momondo.ro

În figura 7, reprezentată de diagrama detaliată a cazurilor de utilizare în modul Client putem observa că există relaţii de incluziune şi extindere între cazurile de utilizare.

O relaţie de tip include este echivalentă cu un apel al unei subrutine în programare şi denotă un comportament obligatoriu, nu unul opţional, prin care se doreşte evitarea redundanţei părţilor cu comportament comun.

În schimb, relaţia de extindere este folosită atunci când unul din cazurile de utilizare au loc doar în anumite condiţii, prin urmare sunt opţionale.

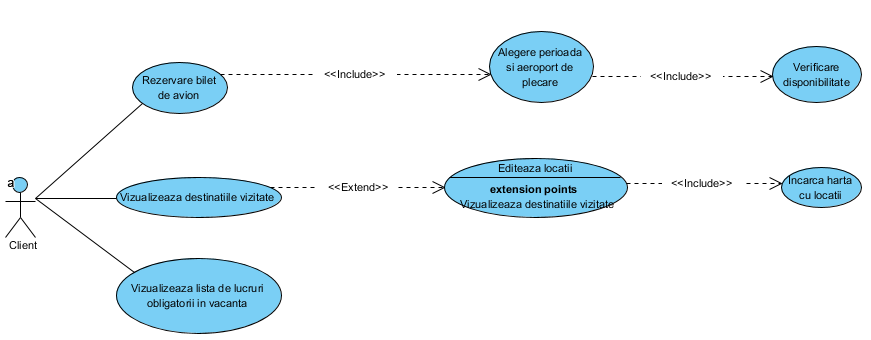
După cum putem observa în figura de mai jos, atunci când clientul doreşte să rezerve un bilet de avion este obligat să aleagă şi o perioadă pentru zbor, cât şi aeroportul de plecare, pentru a verifica disponibilitatea biletelor. Pe de altă parte, atunci când îşi vizualizează destinaţiile vizitate prin încărcarea hărţii cu locaţii, acesta nu este obligat să îşi modifice locaţiile, deci între aceste 2 cazuri de utilizare va exista o relaţie de tip extinde.

Figure 7. Diagrama detaliata a cazurilor de utilizare in modul Client (Utilizator)

## **Modul de implementare a celor mai interesante func**ţ**ii**

Galeria Cover Flow

Pentru a putea crea galeria de imagini pentru fiecare locaţie, se utilizează interfaţa grafică Cover Flow. Galeria este reprezentată sub forma unui carusel cu fotografii, utilizatorul glisând cu degetul pe ecranul telefonului mobil pentru derularea fotografiilor.

Primul pas în implementarea galeriei este adăugarea librăriei specifică acestei interfeţe, astfel se configurează proiectul pentru recunoaşterea proprietăţilor acestei librării. În Gradle Buildig vom adăuga librăria conform figurii 8.



Figure 8. Implementare librarie CoverFlow

Odată ce proiectul a fost configurat, ne vom deschide o activitate nouă pentru a crea galeria de imagini. Activitatea va avea 2 componente: clasa principală şi layout-ul activităţii, adică partea vizuală a acesteia. Prin adăugarea<it.moondroid.coverflow.components.ui.containers.FeatureCoverFlow/> ne vom crea galeria ce conţine un ImageView pentru a încărca imaginile corespunzătoare fiecărei locaţii şi un TextSwitcher pentru afişarea denumirii locaţiei sub imagine, textul schimbându-se odată cu glisarea imaginilor. De asemenea, e nevoie de un nou layout pentru componentele cover flow-ului, mai precis pentru a crea o reflexie a fiecărei imagini. Aceasta va conţine un layout de tip Frame, care va include un view de tip ImageView pentru că şi reflexia imaginii este reprezentată de acelaşi tip de componentă, doar că se schimbă intensitatea contrastului.

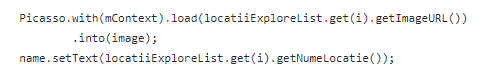
Pentru adăugarea pozei si a denumirii locaţiei, se creează 3 clase: Clasa concretă LocatiiExplore ce conţine 2 atribute de tip String pentru nume si pentru URL-ul imaginii de unde va fi incărcată.

Figure 9. Creare coverflow utilizand Picasso

Cea de-a doua clasă “LocaţiiAdapter” va conţine o listă de tipul primei clase (LocaţiiExplore) şi un obiect de tip Context pentru afişarea denumirii locaţiei în cele 2 limbi ale aplicaţiei: Engleză şi Română. Pentru această clasă există constructor şi metode de get pentru obiectele din listă, dar şi o metodă de tip View ce returnează imaginea conform URL-ului obiectului din listă. Pentru setarea imaginii vom folosi librăria Picasso, metoda fiind prezentată în figura 9. Aceasta manipulează descărcarea imaginilor printr-un adapter.

În clasa activităţii menţionate anterior ne vom declara 4 atribute: primul de tip FeatureCoverFlow pentru galeria în sine, al doilea de tip LocatiiAdapter a putea seta lista locaţiilor, al treilea element e de tip List, iar ultimul va fi de tip TextSwitcher pentru a putea schimba textul în mod dinamic.

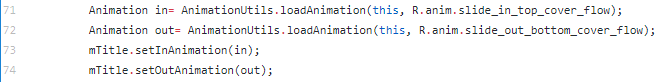
Pentru schimbarea textului s-au creat 2 animaţii în resursele proiectului: una pentru ca textul să apară de sus, iar a doua pentru ca textul să dispară în jos. Acestea vor fi preluate din resurse şi adăugate într-o variabilă de tip Animation pentru setarea mutării textului. Putem observa acest lucru în figura 10.  În metoda principală onCreate () se va apela metoda iniţializare () care adaugă în listă locaţiile aduse din baza de date, fiecare fiind definită de URL-ul imaginii şi denumirea locaţiei. Pe baza listei se creează adapterul pentru a fi folosit la setarea cover flow-ului. Odată ce acesta este creat, galeria va conţine toate pozele şi denumirile locaţiilor, după cum se poate observa în figura 11, dar şi în anexă 3.

Figure 10. Preluare animatii din resurse si setarea tranzitiei textului

Figure 11. Galeria de imagini ale locatiilor

Bara de progres (Wave View)

Wave View-ul este un tip de bibliotecă reprezentat sub forma unui val, care imită în Android progresul unei anumite metode/acţiuni. Este un mod dinamic de indicare a acestui progres şi de asemenea se personalizează acest val prin schimbarea culorii, vitezei de undă, formei de undă a containerului.

În altă ordine de idei, wave view-ul înlocuieşte clasica bară de progres din Android, printr-o animaţie ce imită mişcarea unui val. Pentru implementarea în aplicaţie trebuie adăugată dependinţa necesară în Gradle Building.



Figure 12. Confirgurarea dependintelor proiectului pentru implementarea WaveView-ului

Wave view-ul a fost implementat în formularul de preferinţe. Utilizatorul răspunde la întrebările din formular, iar la fiecare pas parcurs, wave view-ul se extinde, cuprinzând din ce în ce mai mult din ecran. În layout-ul activităţii, s-a adăugat componenta de tip wave view, iar înălţimea acesteia trebuie să fie cât înălţimea layout-ului părinte. În clasa activităţii se declară 2 atribute: prima de tip Wave View, iar a doua de tip FlipProgressDialog.

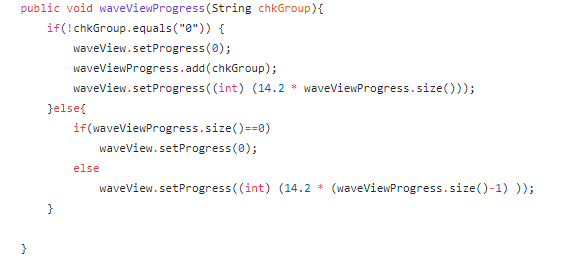
În constructorul clasei, progresul este setat cu 0, actualizându-şi starea prin completarea fiecărei întrebări. Întrucât fiecare întrebare are mai multe variante de răspuns, s-a folosit pentru identificarea acestora, componente de tip RadioButton, respectiv RadioGroup. Astfel nu se pot selecta mai multe răspunsuri şi de asemenea, pentru fiecare radioGroup s-a creat un eveniment pe acesta. Pe fiecare grup se verifică dacă a fost ales unul din răspunsurile disponibile, iar dacă acesta a fost selectat, se apelează metoda waveViewProgress, care actualizează progresul şi astfel se modifică wave screen-ul. Funcţia prezentată mai jos, primeşte ca parametru un String, iar acesta poate conţine “0” când nici un răspuns nu a fost ales, şi un text în caz contrar. Se verifică acest parametru, iar pe baza acestuia, wave view-ul este actualizat.

Figure 13. Funcţie actualizare bara de progres (Wave View)

Generarea locurile din apropiere

Google Maps pune la dispoziţie un API key cu care se poate crea o activitate ce aduce informaţiile necesare despre o anumită locaţie aleasă de utilizator. Se foloseşte acest API key la fiecare request, pe baza căruia se întoarce un răspuns care conţine informaţii despre locaţii împărţite pe categorii, cum ar fi: restaurante, bănci, farmacii, hoteluri şi alte locaţii de interes pentru turişti.

Practic, se creează o succesiune de metode, care caută locaţiile din apropierea utilizatorului, în funcţie de locaţia curentă al acestuia. Răspunsul returnat în urma interogării conţine şi informaţii sumare despre fiecare loc. Toate datele obţinute din răspuns sunt vizibile pe hartă şi pot fi accesate atât pentru examinare, cât şi pentru obţinerea celui mai scurt traseu către acel obiectiv.

De asemenea, cu ajutorul unor radio button-uri, se poate realiza filtrarea locaţiilor din proximitate, afişându-se astfel pe hartă doar anumite locaţii de interes pentru turist din diferite categorii. Toate acestea sunt afişate în partea stângă a ecranului, utilizatorul putând alege una sau mai multe tipuri de locuri pe care doreşte să le găsească într-un anumit moment.

Locatiile vizitate afisate pe harta Google

Cu ajutorul setului pus la dispoziţie de Google (SDK Maps pentru Android) sunt adăugate pe baza coordonatelor geografice locaţii pe harta Google din aplicaţie. Tot cu ajutorul unui API, care gestionează automat accesul la Google Maps, prin intermediul unui “key”, sunt afişate pe hartă toate locaţiile pe care utilizatorul şi le-a selectat iniţial, în formularul de preferinţe, ca locaţii deja vizitate. Harta este personalizată în funcţie de locurile vizitate de fiecare utilizator în parte.

La încărcarea activităţii, se aduc din baza de date locaţiile şi informaţiile necesare, apoi se adugă pe harta Google un marcator de poziție pentru fiecare locaţie în funcţie de coordonatele geografice, iar harta va fi afişată pe continente. Utilizatorul îşi poate vedea toate locaţiile, iar la selectarea uneia dintre acestea, se vor afişa şi informaţii despre locaţie, precum nume, latitudine şi longitudine, după cum se poate observa în anexa 1.

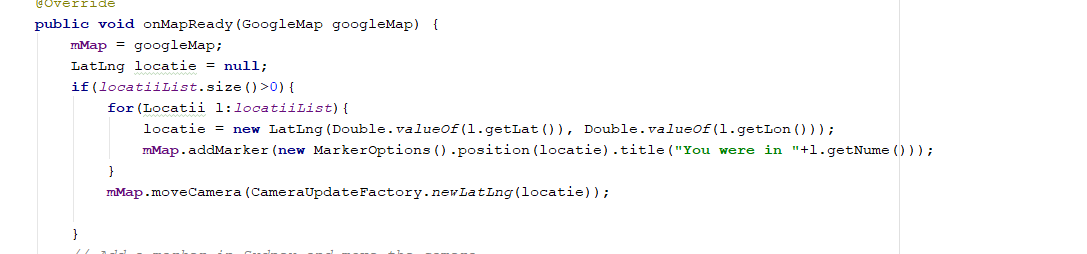


Figure 14. Functie de desenare a markerekor in locatiile vizitate

Vremea in timp real

Deşi nu este o aplicaţie meteo, aceasta furnizează utilizatorului şi informaţii despre situaţia meteorologică în timp real pentru un anumit oraş, de care utilizatorul este interesat. Astfel se foloseşte un serviciu pus la dispoziţie de furnizorul Whether pentru a se obţine informaţii cu privire la vreme. Pe baza setului de servicii web de restful puse de la dispoziţie de furnizor, aplicaţia face un request la fiecare jumătate de oră, pe baza unor coordonate geografice, în urma căruia se primeşte un răspuns care conţine următoarele informaţii: umiditate, temperatură, oră.

La fiecare accesare a activităţii se parsează răspunsul primit, care vine în format Json şi se afişează datele în componente de tip TextView, după cum se poate observa în anexa 2.

Control vocal (Google Cloud Speech-to-Text)

Utilizând tehnologia de învăţare a maşinilor Google, aplicaţia conţine o activitate care procesează în timp real mesajele audio transmise de utilizator, astfel activând controlul vocal al aplicaţiei, pe baza unui dicţionar pus la dispoziţie. Având la bază un Api uşor de utilizat, care recunoaşte peste 120 de limbi, se activează comanda vocală în aplicaţie pentru a oferi aplicaţiei o interacţiune dinamică cu utilizatorii.

Activitatea transpune într-un text view ceea ce utilizatorul a spus, iar dacă aplicaţia regăseşte în dicţionar acea comandă, aceasta va pune la dispoziţia utilizatorului activitatea cerută. Un exemplu concret este atunci când utilizatorul dă o comandă vocală pentru schimbarea limbii aplicaţiei în română sau engleză şi este automat transferat în activitatea de schimbare a limbii.

## **Descrierea principalelor clase implementate**

Pentru dezvoltarea aplicaţiei mobile sunt implementate clase întrucât limbajul de programare este orientat obiect. Astfel, rolul claselor este de a grupa datele și unitățile de prelucrare a acestora într-un modul, unindu-le astfel într-o entitate mult mai naturală. Deși tehnica se numește "Programare Orientată Obiectual", conceptul de bază al ei este Clasa. Clasa, pe lângă faptul că abstractizează foarte mult analiza/sinteza problemei, are proprietatea de generalitate, ea desemnând o mulțime de obiecte care împart o serie de proprietăți. [11]

Pentru crearea obiectelor am grupat fiecare tip în clase diferite, precum:

Clasa Utilizator care conţine 5 atribute (id, nume, email, parolă, codConfirmare) cu scopul de a crea un obiect de tip Utilizator şi pentru salvarea datelor în baza de date, ulterior verificându-se datele salvate cu cele introduse în formularul de autentificare. Dacă datele coincid, înseamnă că utilizatorul a fost creat şi există în baza de date, deci se poate autentifica în aplicaţie. În caz contrar, este necesar ca utilizatorul să îşi creeze cont, ceea ce înseamnă că se va crea un obiect de tip Utilizator folosind constructorul clasei şi setând toate atributele existente în clasă.

Atributul codConfirmare, va fi generat automat de o metodă şi trimis pe adresa de e-mail introdusă de utilizator. Codul odată generat va fi salvat şi în baza de date şi valoarea acestuia se va compara cu codul primit de utilizator şi introdus în activitatea specifică acestei verificări. Dacă cele două coincid, codul de confirmare din baza de date se actualizează cu valoarea 1 şi astfel utilizatorul este creat cu succes.

Tot pentru definirea utilizatorilor s-a implementat clasa PreferinteleUtilizatorului ce va conţine toate informaţiile despre utilizator, precum: statusul relaţiei, dacă are sau nu copii, cât de des pleacă în vacanţă, ce categorii de locaţii îl interesează, bugetul disponibil pentru o vacanţa cât şi o listă cu oraşele vizitate. Toate aceste atribute vor fi setate atunci când utilizatorul completează formularul de preferinţe. Aceste informaţii vor fi salvate în baza de date, întrucât pe baza acestora se vor genera toate ideile de destinaţii, dar se va realiza şi harta locaţiilor vizitate.

O altă clasă importantă în implementarea soluţiei informatice este clasa Locaţii. Aceasta ne ajută să ne creăm obiectul de tip Locaţie şi conţine 6 atribute: id, nume, categorie1, categorie2, latitudine şi longitudine. Clasa conţine atât constructor cât şi metode pentru accesarea şi modificarea valorilor. Pe baza acestei clase va fi creat obiectul de tip Locaţie, dar şi lista de locaţii ce va conţine obiecte de acest tip, iar toate informaţiile vor fi aduse din baza de date şi astfel se va crea harta locaţiilor vizitate, galeria de imagini, dar se va actualiza şi vremea pentru locaţiile ce vor apărea în cele 6 recomandări din activitatea principală.

Unele dintre cele mai importante clase sunt cele folosite pentru realizarea legăturii cu baza de date, stabilind conexiunea, dar şi cele ce salvează, actualizează şi restaurează datele în aceasta. Pentru aceste proceduri s-au implementat 2 clase: ConnectionHelper care stabileşte conexiunea la baza de date, aceasta aflandu-se pe un server.

Clasa DatabaseOperation conţine toate metodele pentru inserare sau restaurarea datelor din baza de date, iar pentru fiecare dintre acestea se verifică conexiunea la baza de date. În caz ca aceasta s-a realizat, se vor efectua operaţiile corespunzătoare metodei, iar în caz contrar se va afişa un mesaj.

## **Structura bazei de date si rolul tabelelor**

Baza de date conţine datele de care are nevoie această aplicaţie pentru a funcţiona. Baza de date conţine 5 tabele ce stochează atât informaţii legate de utilizatori cât şi date despre locaţii.

În tabela “Utilizator” se stochează datele personale ale utilizatorului, precum email, parola (câmp criptat) şi are ca şi cheie primară câmpul “id”. Această tabelă se populează cu date atunci când utilizatorul îşi creează cont în aplicaţie.

Tabela “User\_preferences” conţine toate datele cu ajutorul cărora se creează profilul utilizatorului, urmând să se afişeze ofertele personalizate. Cheia primară a tabelei este câmpul “id”.

O altă tabelă importantă este “Locaţii”, care stochează coordonatele geografice ale locaţiilor pentru determinarea acestora şi sunt salvate în câmpurile “lat”, “lon” pe baza cărora se creează atât “harta personală” cât şi informaţiile meteorologice afişate în activatea de descriere a oraşelor. Totodată fiecare locaţie este încadrată în două categorii prin intermediul câmpurilor “id\_categorie\_1”,”id\_categorie\_2”, iar prin aceste id-uri se face lagătura cu tabela “Categori”.

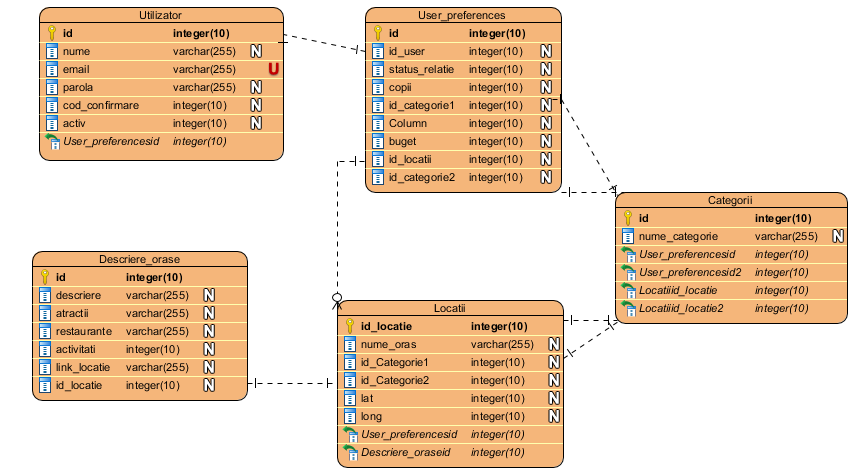
Aceasta din urmă conţine tipurile de categorii pentru locaţii. Cheia primară a acestei tabele este câmpul “id” şi aceasta mai conţine şi denumirea categoriei (exotic, istoric, familie etc).

Figure 15. Schema bazei de date

A cincea tabelă prezentă în baza de date este “Descriere\_oraşe”, care are ca rol principal arhivarea informaţiilor despre fiecare locaţie. Câmpul “ID” este cheia primară a acestei tabele şi de asemenea, conţine şi informaţiile cu privire la restaurante, obiective turistice, activităţile disponibile, cât şi o descriere pentru fiecare locaţie.

Figura 15 prezintă schema bazei de date necesară aplicaţiei. Avem doua tipuri de relaţie prezente între tabele şi anume: “Unu-la-unu” şi “Unu-la-multi”.

Relația “unu-la-unu” dintre două tabele presupune că unei înregistraţi din prima tabelă îi corespunde o singură înregistrare în cea de-a doua.

Relația unu-la-mai-mulți explică faptul că o inregistrare din primul tabel poate să aibă cel puţin două inregistrări corespondente în tabelul 2, dar unei inregistrări din tabelul 2 îi poate corespunde o singură înregistrare în tabelul 1.

Această legătură se realizează prin adăugarea câmpurilor care constituie cheie primară din tabelul 1 şi care serveşte drept cheie externă în tabelul 2.

**Legaturile dintre tabele:**

* Tabela Utilizator <-> User\_preferences – tabela utilizator are următoarele câmpuri: “id”, “nume”, “email”, “parolă”, “cod\_confirmare” şi “activ”. Câmpul “ID” este cheie primară în tabela Utilizator, şi cheie externă în tabela User\_preferences, prin urmare este câmpul de legătură dintre aceste două tablele. Legătura este de tip “unu la unu”.
* Tabela Categorii <-> User\_preferences – legătura dintre cele două tabele se face prin intermediul câmpurilor: “id\_categorie\_1”,”id\_categorie\_2”, practic aceste 2 câmpuri reflectă categoriile de interes ale utilizatorului, iar “ID” este cheie primară în tabela Categorie şi cheie externă în tabela User\_Preferences. Tipul legăturii dintre aceste tabele este “unu la unu”.
* Tabela Locatii <-> Categorii – între cele două tabele există o legătură de tip “unu la unu” care se realizează asemănător punctului anterior, adică fiecare locaţie este încadrată în 2 categorii, prin intermediul câmpurilor “id” cu “id\_categorie\_1”, ”id\_categorie\_2”.
* Tabela Descriere\_orase <-> Locatii: între cele două tabele este conturată o legătură de tip “unu-la-unu” prin intermediul câmpului “id”, folosit ca şi cheie primară în tabelă Locaţii, iar câmpul “id\_locaţie” din tabela Descriere\_oraşe îi permite administratorului de baze de date să asocieze fiecărei descrieri un identificator.
* Tabela Locatii <-> User\_preferences – Între aceste două tabele există o legătură unu-la-mulţi (one to many). Practic, fiecare utilizator poate selecta mai multe oraşe pe care le-a vizitat, dar fiecărui id de locaţie îi corespunde o singură locaţie din tabela “Locaţii”

# **Concluzii**

Această aplicaţie a avut ca scop simplificarea organizării călătoriilor viitoare ale utilizatorilor săi, păstrarea unui istoric al călătoriilor, recomandarea unor excursii şi activităţi în diferite oraşe în funcţie de bugetul fiecărui utilizator, cât şi de a veni în sprijinul acestora pentru o împachetare a bagajelor mai eficientă, întrucât acestea întotdeauna reprezintă ultimul şi cel mai dificil pas înainte de călătorie. De asemenea, aplicaţia a încercat să creeze o comunitate prin care utilizatorii să îşi împărtăşească experienţele din vacanţă. Pentru ca aplicaţia să aibă o interfaţă prietenoasă s-au implementat multe librării ce au un aspect diferit şi mai dinamic, precum: Wave view, galeria Cover flow şi Card View. Aceasta îşi îndeplineşte obiectivele propuse, iar grupul ţinta este reprezentat de orice persoană pasionată de arta călătoriei. Un avantaj al aplicaţiei este dat de faptul că aceasta este configurată în două limbi: engleză şi romană. Astfel se extinde grupul ţintă şi pentru persoanele din străinătate.

Un dezavantaj al aplicaţiei este dat de numărul mic al locaţiilor existente în baza de date, mai exact 42 de oraşe din cele 7 continente şi de faptul că fiecare oraş face parte doar din 2 categorii. Acest lucru poate duce la recomandarea unor locaţii ce nu sunt conform preferinţelor utilizatorilor, dacă acesta deja a vizitat o mare parte din destinaţii.

Întrucât tehnologia evoluează rapid, iar călătoriile devin tot mai accesibile, aplicaţia poate fi dezvoltată în viitor. Se pot adăuga mult mai multe locaţii în baza de date şi de asemenea, se poate implementa şi posibilitatea ca utilizatorii să îşi încarce poze şi filmuleţe din vacanţă.

În plus, se poate implementa o funcţionalitate care să îi organizeze vacanţa utilizatorului în funcţie de locaţia aleasă şi de numărul de zile pe care acesta doreşte să le petreacă în vacanţă, ţinându-se cont şi dacă turistul călătoreşte singur sau împreună cu familia ori copii.

O altă idee de dezvoltare este implementarea citirii codurilor QR pentru boarding pass-uri şi biletele de avion achiziţionate de utilizator şi salvarea acestora în aplicaţie pentru ca acesta să le poată accesa mult mai uşor şi de oricâte ori doreşte, dar şi extinderea aplicaţiei mobile în varianta web şi IOS.

În concluzie, planificatorul de călătorii realizat reprezintă un instrument folositor pentru oamenii din ziua de astăzi şi pentru toţi cei care plănuiesc să călătorească, dar şi o aplicaţie extrem de utilă pentru cei care doresc câteva recomandări şi îndrumări în aceste aspecte ale călătoriei.

# **Bibliografie**

[1] J. CAO, A.D. GALINSKY, W.W. MADDUX (December 2013). Does Travel Broaden the Mind?

[2] Burj Khalifa, cel mai vizitat obiectiv turistic  [https://thearabianpost.com/tap/2014/02/burj-khalifa-is-dubais-most-visited-tourist-attraction.html](%20https://thearabianpost.com/tap/2014/02/burj-khalifa-is-dubais-most-visited-tourist-attraction.html)

[3]Cele mai populare obiective turistice din lume <https://www.statista.com/statistics/264510/ranking-of-the-20-most-popular-tourist-attractions-in-the-world/#0>

[4]Statisa, Obiective turistice populare in lume <https://www.statista.com/statistics/264510/ranking-of-the-20-most-popular-tourist-attractions-in-the-world/>

[5]Statisa, Statistici privind utilizatorii activi de pe Facebook<https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/>

[6]Statisa, Mediile online in functie de numarul de utilizatori <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>

[7]SocialPilot, Statistici privind retelel de socializare <https://www.socialpilot.co/blog/social-media-statistics>

[8] Android, <https://ro.wikipedia.org/wiki/Android_(sistem_de_operare)>

[9]Android Studio, <https://en.wikipedia.org/wiki/Android_Studio>

[10]Aplicatia android, <http://www.itcsolutions.eu/2011/09/08/android-tutorial-concepte-activitati-si-resurse-ale-unei-aplicatii-android/>

[11]POO, <https://ro.wikipedia.org/wiki/Programare_orientat%C4%83_pe_obiecte>

[12] Criptarea datelor, <http://www.scientia.ro/tehnologie/34-cum-functioneaza-calculatorul/409-cum-functioneaza-criptarea-datelor.html>

# Anexa 1- Harta cu locatiile vizitate de utilizator



# Anexa 2- Activitatea cu vremea actualizata in timp real

# Anexa 3- Galeria de imagini (Cover Flow)

