

UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL: Calculatoare și tehnologia informației
SPECIALIZAREA: Tehnologia informației

Aplicație Android pentru sesizări către autorități

Raport intermediar II

Student(ă)
Dumitru-Daniel Davidescu

Iași, 2019

Capitolul 1. Proiectarea hardware/software a aplicației

Aplicația Android pentru sesizări către autoritățile din orașul Iași este o aplicație destinată cetățenilor orașului Iași, care dețin un dispozitiv mobile pe care rulează sistemul de operare Android, și care vor să contribuie la modernizarea și buna organizare a orașului.

Aplicația reprezintă cea mai simplă conexiune între cetățean și autoritățile administrative, prin punerea la dispoziția cetățeanului a unei forme moderne de sesizare și raportare a problemelor de ordin civic cu care se poate confrunta în spațiul public.

Aplicația este formată din două componente: o aplicație client ce rulează pe dispozitive Android și o aplicație server. Aplicația client rulează pe dispozitive Android cu versiunea cel puțin 4.1, astfel, aplicația poate fi folosită pe majoritatea dispozitivelor care rulează sistemul de operare Android.

De asemenea aplicația va avea două interfețe: pentru utilizator (aplicație Android), și pentru administrator (aplicație WEB).

1.1. Funcțiile sistemului

Principalele funcționalități ale aplicației sunt:

- Înregistrarea/autentificarea utilizatorului:
 - necesară pentru a putea folosi capacitățile puse la dispoziție de către sistemelor
 - permite utilizatorilor să își creeze un cont sau să se autentifice în cadrul aplicației
- Vizualizarea profilului
 - permite utilizatorului să vadă câte sesizări a trimis, și câte sunt rezolvate
- Sesizare nouă
 - permite utilizatorului să trimită o sesizare nouă către autorități
 - utilizatorul selectează categoria din care face parte sesizarea
 - localizarea pe hartă pentru a afla adresa exactă de unde se trimite sesizarea
 - utilizatorul are opțiunea de a încărca o poză din galeria dispozitivului, fie de a face una direct din aplicație
 - utilizatorul poate scrie o mică descriere a problemei
 - vizualizarea sesizării înainte de a fi trimisă
- Vizualizarea tuturor sesizărilor trimise

1.2. Sistemul de operare Android

Pentru dezvoltarea aplicației am ales sistemul de operare Android, datorită faptului că este cea mai răspândită tehnologie pe dispozitivele mobile actuale, fiind totodată o platformă open-source, și în prezent cel mai popular sistem de operare dintre cele dedicate dispozitivelor portabile.

Android este o platformă software și un sistem de operare pentru dispozitive și telefoane mobile bazată pe nucleul Linux, fiind un model de arhitectura MVC¹ (Model-View-Controller) dezvoltată inițial de compania Google, iar mai târziu de consorțiul comercial Open Handset Alliance în care fac parte companii ca ARM Holding, HTC, Intel, LG, Motorola, Samsung Electronics.

Dezvoltarea aplicațiilor Android se realizează în limbajul Java, dezvoltatorul folosindu-se de API-uri/biblioteci dezvoltate de Google. De astfel, se pot scrie și aplicații în limbajul C, fiind

¹ MCV, sau Model-View-Controller este un șablon arhitectural folosit în industria de software pentru izolarea cu succes a părții logice de interfața proiectului, rezultând în aplicații extrem de ușor de modificat.

apoi compilate în cod mașină ARM și executate [1].

1.2.1. Arhitectura unei aplicații Android

Pentru a înțelege arhitectura unei aplicații mobile Android este nevoie de un minim de cunoștințe cu privire la conceptele cheie ale aplicațiilor Android. Înțelegerea acestora va permite dezvoltatorului să controleze:

- componentele aplicației;
- ciclul de viață al aplicației;
- resursele aplicației;

Principalele caracteristici ale unei aplicații Android sunt:

- este distribuită într-un pachet Android (Android Package), care este un fișier .apk;
- operațiile critice (acces la Internet, citire/scriere date de contact, monitorizare SMS, acces la modulul GPS, acces la camera foto), pot fi restricționate sau se poate solicita permisiunea utilizatorului, utilizând fișierul manifest de configurare a aplicației, *AndroidManifest.xml*.

În condițiile în care pe piața dispozitivelor mobile aplicațiile sunt cele care aduc avantajul competițional, beneficiul Android este reprezentat de abordarea unitară pentru dezvoltarea aplicațiilor. Cu alte cuvinte, o aplicație dezvoltată conform API-ului Android va putea rula pe mai multe dispozitive mobile pe care este instalat sistemul de operare respectiv [2].

1.3. Node.js

Node.js este o platformă pentru aplicații scalabile de tip server sau rețea. Aplicațiile Node.js sunt scrise în JavaScript, și pot fi rulate în runtime-ul Node.js pe Windows, Mac OS X sau Linux, fără a fi nevoie de nicio schimbare.

Aplicațiile Node.js sunt concepute pentru maximizarea randamentului și a eficienței folosind operații I/O neblocaante și evenimente asincrone. Aplicațiile Node.js rulează pe un singur fir de execuție deși platforma folosește mai multe fire pentru evenimentele legate de rețea și fișiere.

Node.js conține intern o bibliotecă asincronă pentru operații I/O, socket și comunicații HTTP. Suportul pentru socket și HTTP permite platformei să joace rolul unui server web, fără a mai fi nevoie de software adițional pentru servere web precum Apache².

1.4. MongoDB

MongoDB este o bază de date NoSQL open-source orientată pe documente. Această bază de date beneficiază de suport din partea companiei 10gen. MongoDB face parte din familia sistemelor de baze de date NoSQL. Diferența principală constă în faptul că stocarea datelor nu se face folosind tabele precum într-o bază de date relațională, MongoDB stochează datele sub formă de documente JSON³ cu scheme dinamice (MongoDB numește formatul BSON) făcând integrarea datelor în anumite tipuri de aplicații mai ușoară și mai rapidă.

MongoDB suportă căutarea după câmpuri, interogări sau după expresii regulate. Interogările pot returna câmpuri specifice ale documentelor și pot de asemenea include funcții JavaScript definite de către utilizator. Orice câmp din interiorul unui document stocat în interiorul bazei de date poate fi indexat. De asemenea, indecșii secundari sunt și ei disponibili.

² Apache este o aplicație de tip server web ce a jucat un rol important în creșterea și dezvoltarea World Wide Web.

³ JSON este un acronim în limba engleză pentru JavaScript Object Notation, și este un format de reprezentare și interschimb de date între aplicații informatice. JSON este alternativa mai simplă, mai facilă decât XML.

Datorită faptului că MongoDB este ușor de folosit, driverul pentru Node.js poate fi o soluție optimă pentru o aplicație simplistă. Totuși, pentru aplicații mai complexe, se recomandă folosirea unui Object Document Mapper pentru modelarea funcțiilor. Mongoose este ODM-ul oficial suportat pentru platforma Node.js.

Mongoose este o bibliotecă de modelare a datelor obiect pentru MongoDB și Node.js. Mongoose transformă datele din baza de date în obiect JavaScript pentru a putea fi folosite în cadrul aplicației.

1.5. Diagrame UML

1.5.1. Diagrame use-case

Use case diagram (Figura 1.1) este un tip de diagramă din care reiese modul de utilizare a sistemului informatic, modul în care utilizatorii interacționează cu acesta (în corespondență directă cu task-urile acestor utilizatori).

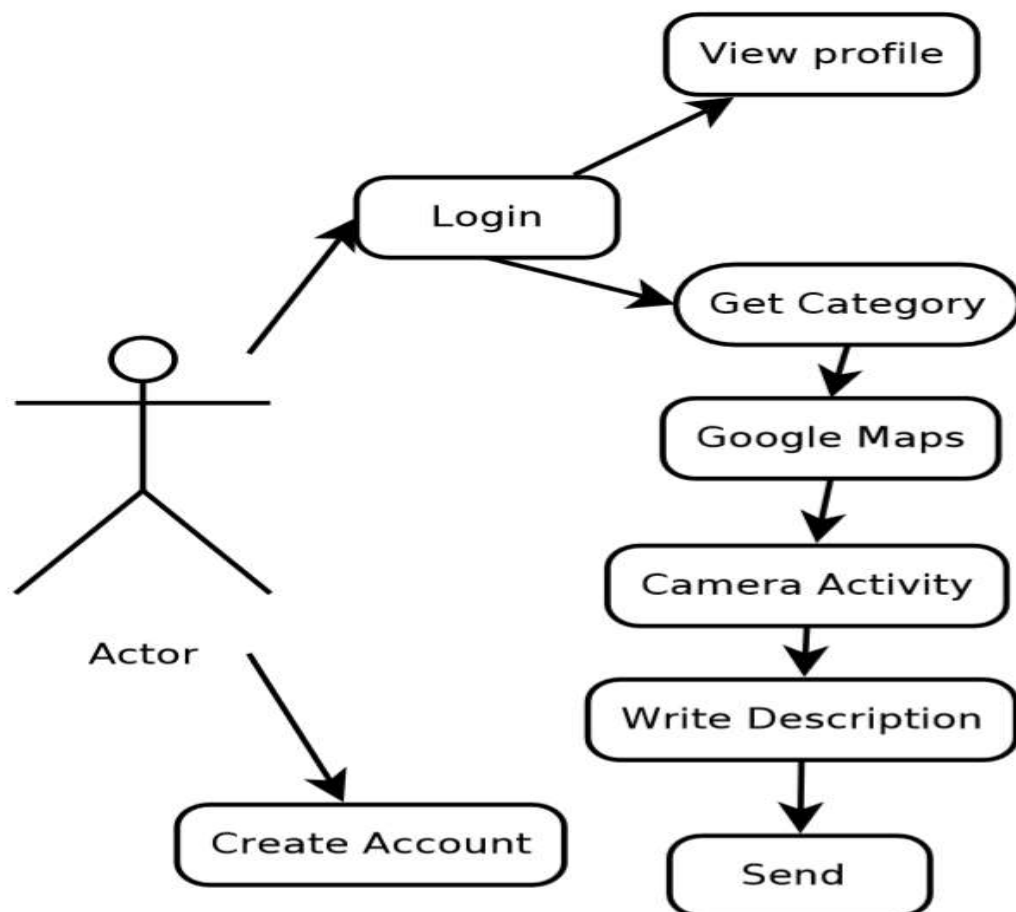


Figura 1.1: Diagrama use-case aferentă utilizării aplicației Android

Odată ce și-a creat un cont, actorul are posibilitatea de a-și vedea profilul, care conține numărul de sesizări trimise, numărul de sesizări rezolvate, sau în curs de rezolvare. De asemenea, utilizatorul poate vedea ce sesizări a trimis în trecut precum și statusul acestora.

Totodată, el poate să trimită o nouă sesizare, selectând butonul „Sesizare nouă”, apoi

trebuie să aleagă categoria în care se încadrează sesizarea, după care va fi direcționat către Google Maps pentru a afla adresa de unde se trimite aceasta. Pasul următor este să încarce o fotografie, și să scrie o mică descriere a problemei.

După ce sesizarea a fost trimisă, datele vor fi stocate într-o bază de date MongoDB. Administratorul va avea acces la aceste date, prin intermediul unui website (Figura 1.2).

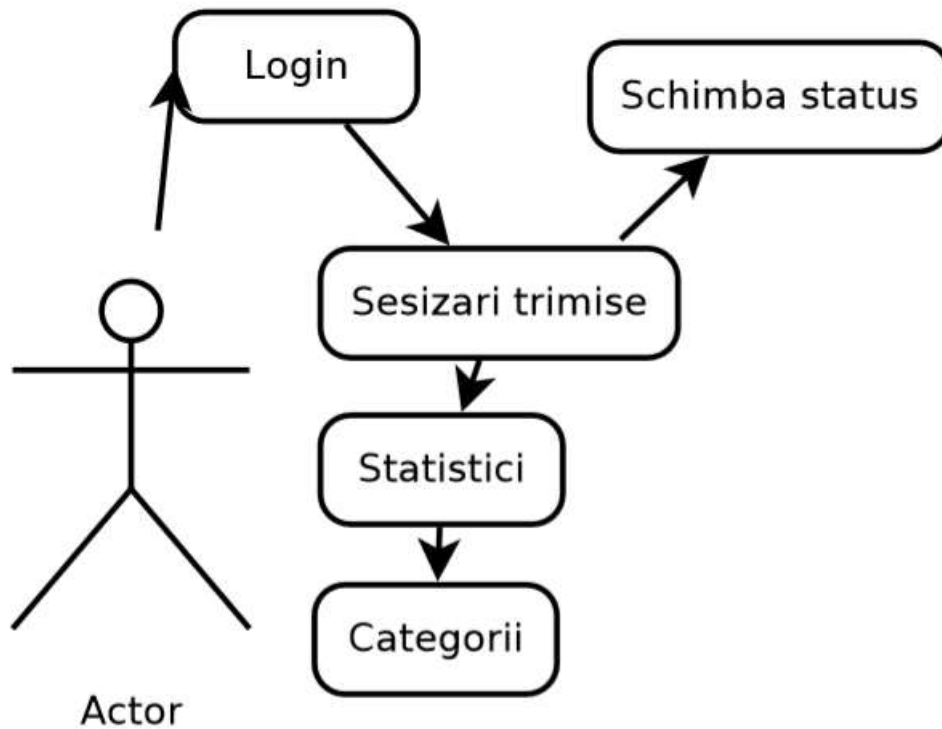


Figura 1.2: Diagrama use-case aferentă utilizării website-ului dedicat administratorului

Capitolul 2. Rezultate intermediare obținute

Mai jos, sunt prezentate în detaliu, funcționalitățile sistemului. De asemenea, acestea sunt însoțite de imagini pentru o înțelegere mai ușoară.

2.1. Aplicația Android

- Înregistrarea/autentificarea utilizatorului

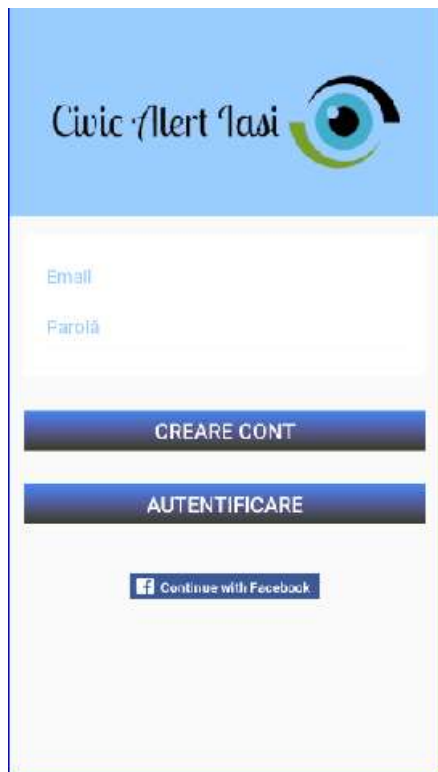


Figura 2.1: Autentificarea utilizatorului

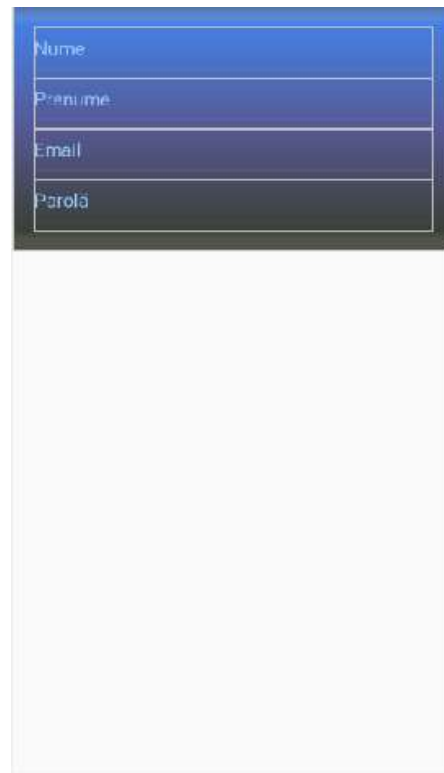


Figura 2.2: Înregistrarea utilizatorului

Pentru a putea utiliza aplicația, utilizatorul are nevoie de un cont, pentru a se putea autentifica. Pentru aceasta, utilizatorul trebuie să-și introducă numele, prenumele, adresa de email precum și o parolă. După introducerea acestor date, aplicația va face o cerere către aplicația server a sistemului pentru crearea unui cont. Dacă adresa de email nu a mai fost utilizată de către un al utilizator, contul se va crea cu succes.

După ce contul a fost creat, utilizatorul se va autentifica, pentru a putea folosi în continuare aplicația.

- Meniul principal



Figura 2.3: Meniul principal

Odată logat, utilizatorul are acces la meniul principal al aplicației, unde va putea trimite o sesizare nouă, își va putea vedea contul, sau sesizările trimise până atunci.

- Alegerea categoriei



Figura 2.4: Alegerea categoriei

Aici, utilizatorul va avea la dispoziție o listă de categorii, de unde va trebui să aleagă în funcție de sesizarea pe care dorește să o trimită.

Capitolul 3. Dificultăți/provocări întâmpinate și soluții de rezolvare

Una din cele mai mari provocări întâmpinate în realizarea aplicației Android, a fost mediul de lucru Android Studio deoarece este unul complex.

De asemenea, update-urile făcute destul de des asupra Android Studio, nu au fost întotdeauna utile pentru mine.

Aplicația să fie responsive pentru orice fel de dispozitiv Android, a fost de asemenea o provocare pentru mine.

Bibliografie

- [1] Cristian Stoica, Android Smart Presentation [Online], Disponibil la adresa: <https://aimas.cs.pub.ro/amicity/doc/GeorgeCristianStoica-AndroidSmartPresentationFeedback-document.pdf>, Accesat: 2012.
- [2] *, Android Tutorial-Concepte, activități și resurse ale unei aplicații android [Online], Disponibil la adresa: <http://www.itcsolutions.eu/2011/09/08/android-tutorial-concepte-activitati-si-resurse-ale-unei-aplicatii-android/>, Accesat: 2017.