



**TECHNICAL
UNIVERSITY**
OF CLUJ-NAPOCA
ROMANIA

FACULTATEA: Automatică și Calculatoare
SPECIALIZAREA: Calculatoare și Tehnologia Informației
DISCIPLINA: Ingineria Sistemelor
PROIECT: Chat Application for Android

Îndrumător laborator:

Delia Aleandrina Mitrea

Student:

Galiș George-Laurențiu



Cuprins

1. Specificatii si analiza sistemului.....	3
2. Precizarea limbajului de programare ales, a sistemului de operare dupa care se face implementarea.....	4
3. Fundamente teoretice.....	7
4. Proiectarea aplicatiei.....	7
4.1. Descrierea arhitecturii sistemului	
4.2. Diagrama de clase	
4.3. Diagrame de interactiune: activitate, secventa colaborare	
4.4. Detalii de implementare	
4.5. Cazuri de testare	
5. Testarea propriu-zisă a aplicației	14
6. Manual de instalare și utilizare,.....	16
7. Concluzii și dezvoltări ulterioare	19
8. Bibliografie si webografie.....	20



1. Specificatii si analiza sistemului

Proiectul de fata reprezinta o aplicatie de chat pentru sistemul de operare Android. Aceasta este destinata fiecărei persoane care doreste sa comunice cu alti utilizatori aflatii într-o baza de date controlata de administratorul serverului. Utilizatorii pot interactiona cu aceasta aplicatie printr-o interfata grafica usor de folosit si foarte intuitiva.

Actor	Tip	Descriere
Administrator	Uman	Persoana care administrează/configurează sistemul si baza de date
User autorizat (client)	Uman	Persoana care se folosește de sistem

Desciere detaliata a administratorului:

Administratorul va fi cel care va avea grija de intretinerea sistemului, configurarea si imbunatatirea acestuia, mai exact – administrarea sistemului. Acesta va putea elimina profiluri de utilizatori, deci va avea acces asupra datelor stocate in baza de date. Pentru a putea folosi aplicatia, administratorul trebuie la randul lui sa se autentifice in aplicatie printr-un username si parola. O data autentificat, se va afisa numele da administrator si toti utiliatorii aplicatie, dupa care acesta poate sa adauge sau sa stearga utilizatori din aplicatie.

Unde e utilizat:

- eliminare utilizator;
- adaugare utilizatori in baza de date
- imbunatatirea aplicatiei si optimizarea ei
- comunice cu alti utilizatori din baza de date
- adaugare de mesaje in baza de date respectiva

**Descriere detaliata a utilizatorului:**

User-ul, adica utilizatorul obisnuit va putea sa se inregistreze printr-un username si parola, sa vada utilizatorii bazei de date si sa comunice cu acestia prin intermediul aplicatiei. Acesta nu va putea sa adauge sau sa stearga utilizatori din baza de date, doar sa adauge in baza de date in care sunt stocate mesajele si sa o foloseasca.

Unde e utilizat:

- înregistrare ca utilizator;
 - comunice cu alti utilizatori din baza de date
- adaugare de mesaje în baza de date respectiva

2. Precizarea limbajului de programare ales, a sistemului de operare dupa care se face implementarea

2.1. Java

Java este un limbaj de programare orientat-obiect, puternic tipizat, conceput de către James Gosling la Sun Microsystems (acum filială Oracle) la începutul anilor '90, fiind lansat în 1995. Cele mai multe aplicații distribuite sunt scrise în Java, iar noile evoluții tehnologice permit utilizarea sa și pe dispozitive mobile, spre exemplu telefon, agenda electronică, palmtop etc. În felul acesta se creează o platformă unică, la nivelul programatorului, deasupra unui mediu eterogen extrem de diversificat. Acesta este utilizat în prezent cu succes și pentru programarea aplicațiilor destinate intranet-urilor.

Limbajul împrumută o mare parte din sintaxă de la C și C++, dar are un model al obiectelor mai simplu și prezintă mai puține facilități de nivel jos. Un program Java compilat, corect scris, poate fi rulat fără modificări pe orice platformă care e instalată o mașină virtuală Java (engleză Java Virtual Machine, prescurtat JVM). Acest nivel de portabilitate (inexistent pentru limbaje mai vechi cum ar fi C) este posibil deoarece sursele Java sunt compilate într-un format standard numit cod de octeți (engleză byte-code) care este intermediar între codul mașină (dependent de tipul calculatorului) și codul sursă.



2.2. SQLite

SQLite este o mică bibliotecă C care implementează un motor de baze de date SQL încapsulat, oferă posibilitatea de a-l introduce în diverse sisteme și necesită zero-configurare.

SQLite este o mică bibliotecă C care implementează un motor de baze de date SQL încapsulat, oferă posibilitatea de a-l introduce în diverse sisteme și necesită zero-configurare. Caracteristicile includ:

- Tranzacțiile sunt atomice, consistente, izolate și durabile (ACID) chiar după căderi de sistem și pene de current.
- Zero-configurare - nici o setare sau administrare necesară.
- Implementează o mare parte a SQL92. (sunt unele caracteristici nesuportate)
- O bază de date întreagă este stocată într-un singur fișier de pe HDD.
- Fișierele bază de date pot fi liber partajate între mașini cu ordine diferită de octeți.
- Suportă baze de date de până la 2 teraocteți (2^{41} octeți) în mărime.
- Mărimea șirurilor și BLOB-urile limitate doar de memoria disponibilă.
- Cod de bază mic: mai puțin de 250KiO configurat complet sau mai puțin de 150KiO cu caracteristici opționale omise.
- Mai rapid decât baze de date client/server cunoscute pentru majoritatea operațiilor obișnuite.
- API simplu, ușor de folosit.
- Legături TCL incluse. Legături către alte multe limbaje disponibile separat.
- Cod sursă foarte bine comentat cu peste 95% acoperire prin teste.
- Încapsulat: nu are dependențe externe.
- Sursele sunt în domeniul public. Poate fi folosit pentru orice scop.



2.3. XML

Extensible Markup Language (XML) este un limbaj pentru crearea altor limbaje, are o structura bine definita si oate fi utilizat pentru a crea limbaje de marcare precum HTML, XHTML.

XML a fost elaborat pentru:

- separarea sintaxei de semantica pentru a furniza un cadru comun de structurare a informatiei
- construirea de limbaje de mark-up pentru aplicatii din orice domeniu
- structurarea informatiei in viitor
- asigurarea independentei de platforma si suport pentru internationalizare

Avantajele

- Extensibilitate - se pot defini noi indicatori dacă este nevoie;
- Validitate -se verifică corectitudinea structurală a datelor;
- Oferă utilizatorilor posibilitatea de a-și reprezenta datele întrun mod independent de aplicație;
- XML este simplu și accesibil (sunt fișiere text create pentru a structura, stoca și a transporta informația);
- Poate fi editat, modificat foarte ușor, necesită doar un editor text simplu precum notepad, wordpad etc.

Dezavantaje

- Oferă un model de date ne-normalizat;
- Dificil de decis ce sa fie continut si ce sa fie atribut;
- Flexibilitate (uneori dezavantaj);
- Lipsa de suport in browser-e sau alte aplicatii enduser;
- Mai strict decit HTML (daca exista o eroare in document nu se opreste rularea scriptuului).



3.Fundamente teoretice

Pentru realizarea acestui proiect este nevoie de cunotstiinte legate de POO(Programare Orientata pe Obiect). Limbajul caracteristic care poate aplica acest tip de programare este Java. Pe lanaga acest lucru, este nevoie si de cunostiinte legate de baze de date. Mai exact, creerea sau manipularea tabelelor. Mai mult de atat este necesara si introducerea, stergerea si modificarea datelor din tabelele respective.

4. Proiectarea aplicatiei

4.1. Descrierea arhitecturii sistemului

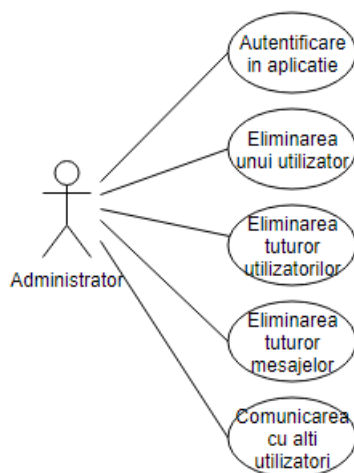
- **Administratorul:**

Administratorul va fi cel care va avea grija de intretinerea sistemului, configurarea si imbunatatirea acestuia, mai exact – admnistrarea sistemului. Acesta va putea adauga sau elimina profiluri de utilizatori, deci va avea acces asupra datelor stocate in baza de date. Pentru a putea folosii aplicatia, administratorul trebuie la randul lui sa se autentifice in aplicatie printr-un username si parola. O data autentificat, se va afisa numele da administrator si toti utiliatorii aplicatie, dupa care acesta poate sa adauge sau sa stearga utilizatori din aplicatie.

Unde e utilizat:

- eliminare utilizator;
- adaugare utilizatori in baza de date
- imbunatatirea aplicatiei si optimizarea ei
- comunice cu alti utilizatori din baza de date
- adaugare de mesaje in baza de date respectiva
-

Diagrama use-case a administratorului:



Descriere:

Initial administratorul se va autentifica prin intermediul unui username si parola. Daca se greseste introducerea acestor componente se va afisa un mesaj corespunzator. O data ce administratorul s-a autentificat, acesta poate vedea o lista in care sunt toate conturile de utilizatori ai aplicatiei. Acesta poate sa comunice cu fiecare si sa elimine din baza de date oricare din acestia cu autorul unei iconite corespunzatoare. Mai mult de atat acesta poate sa elimine toate mesajele sau ulitizatorii care se afla in baza de date.

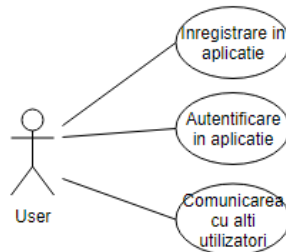
• User-ul:

Un user – utilizator obișnuit poate fi o persoana care se afla deja in baza de date a aplicatiei. Acest lucru poate fi posibil doar dupa ce acesta si-a creat un cont tot prin intermediul aplicatiei. Activitatea acestuia este determinata de administrator adica el poate fi sters din baza de date oricand doreste administratorul.

Unde e utilizat:

- eliminare utilizator;
- adaugare utilizatori in baza de date
- imbunatatirea aplicatiei si optimizarea ei
- comunice cu alti utilizatori din baza de date
- adaugare de mesaje in baza de date respectiva

Diagrama use-case a user-ului:

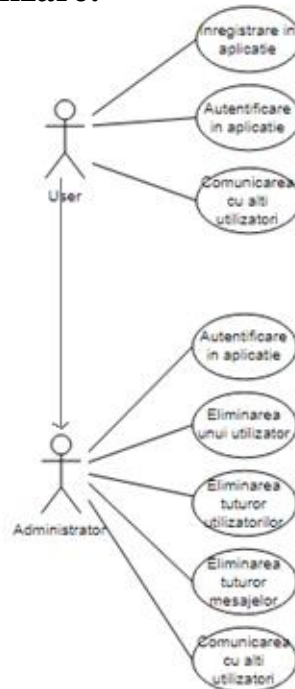


Descriere:

Initial user-ul autentificat cu un username si parola, acestea fiind introduse in baza de date. Dupa care acesta va folosi datele cu care s-a autentificat si si se va inregistra si va vedea lista utilizatorilor care folosesc aplicatia. Acesta va putea comunica cu oricare dintre acestia.

4.2. Identificarea functionalitatilor aplicatiei

Diagrama de cazuri de utilizare:

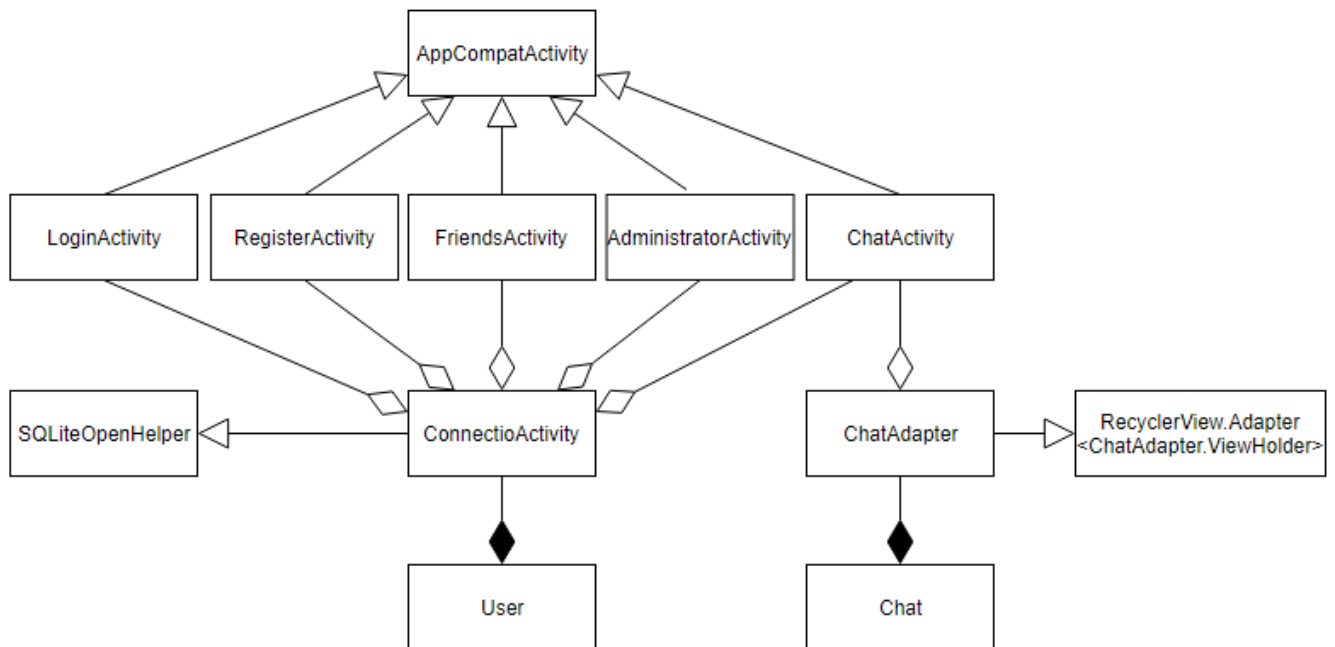




Descriere:

In general, activitatea utilizatorului este determinata de administrator. Cu alte cuvinte, utilizatorul poate fi sters oricand de administrator din cauza diferitor motive. O data ce acesta este sters, utilizatorul poate sa faca din nou un cont nou, dar acesta nu va avea salvate nici un mesaj cu ceilalti din baza de date deoarece acesta este un user nou.

4.3. Diagrama de clase



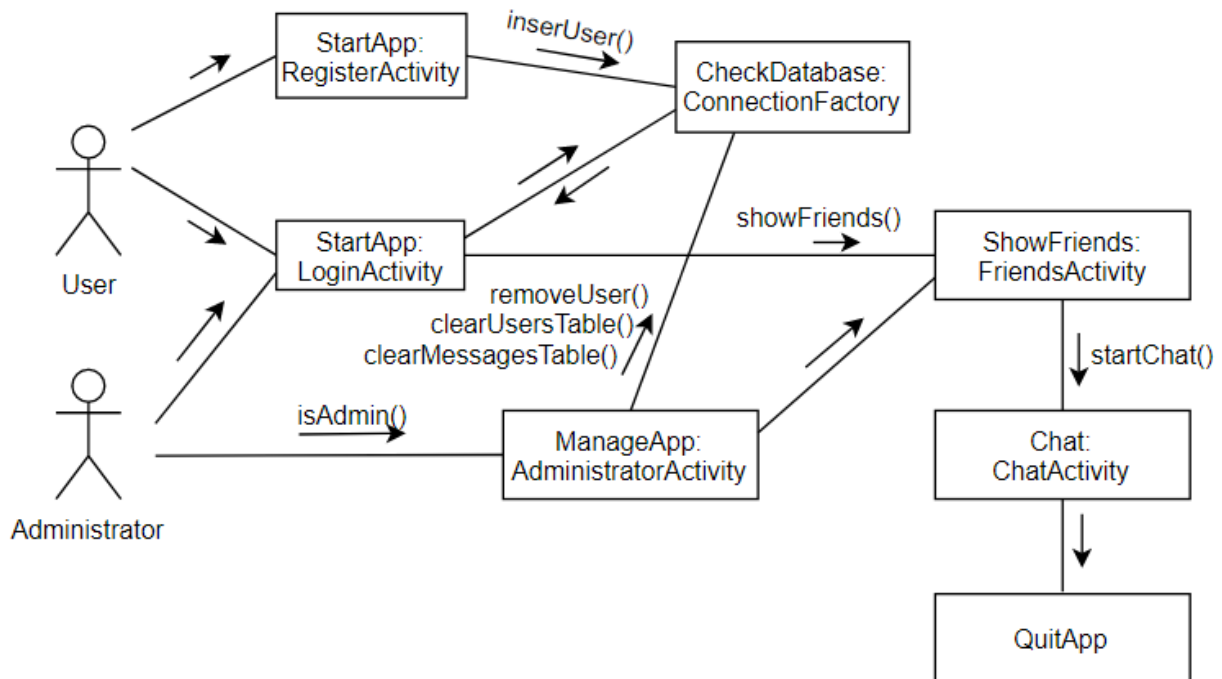


- **Activitate**



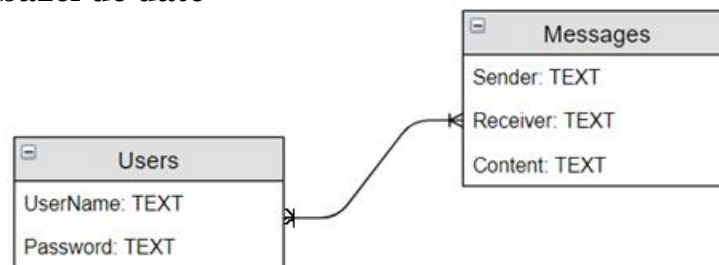


- Colaborare



4.5. Detalii de implementare

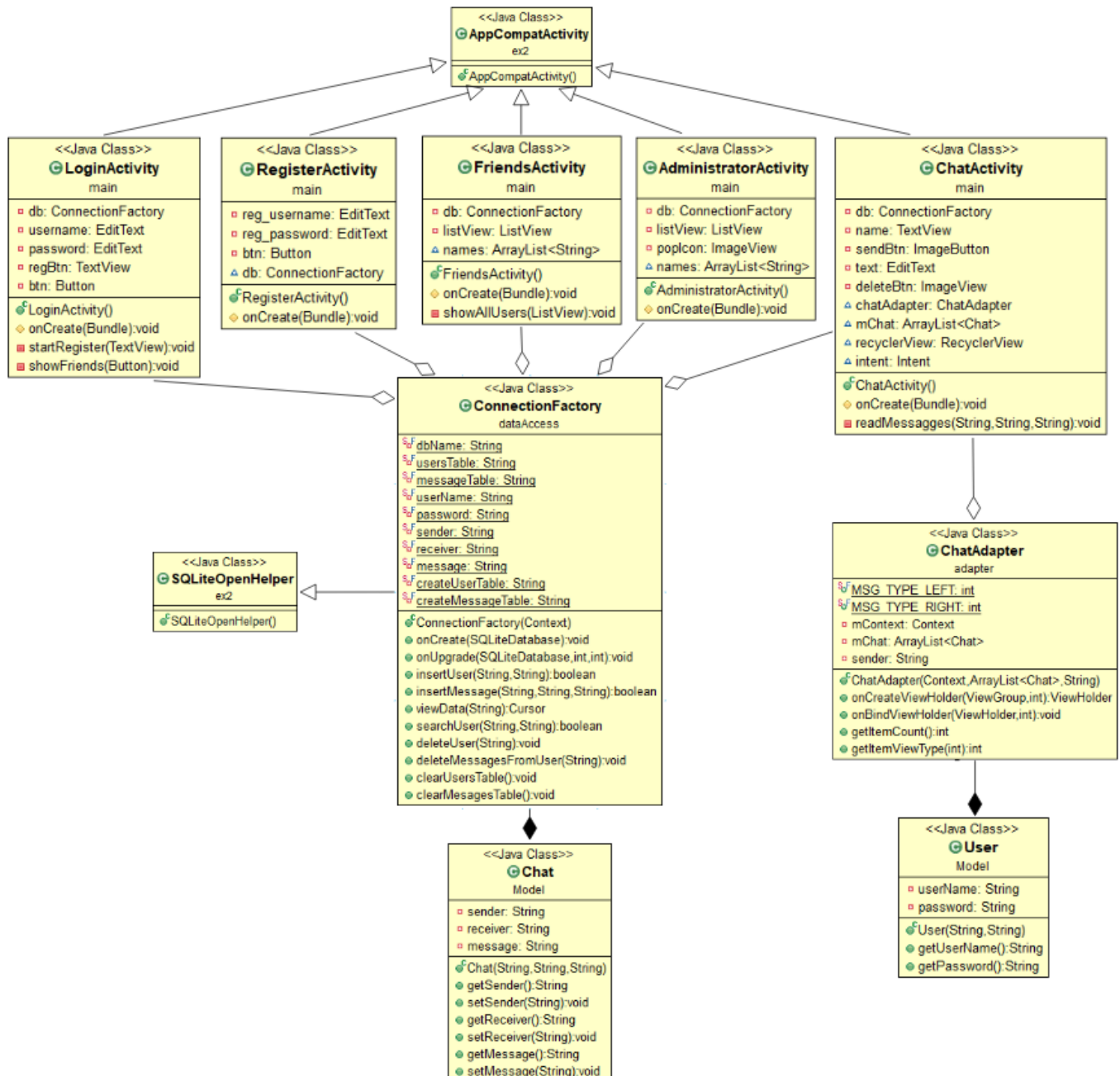
- Diagrama bazei de date



Baza de date folosita se numeste Chat.db si este formata din doua table: Users si Messages. Relatia dintre acestea este una de recurenta de tipul MannyToManny deoarece fiecare mesaj este format dintr-un sender. Deci, fiecare user poate fi si Sender si Receiver in acelasi timp. Tabela mesaje contine coloanele Sender(user-ul care trimite mesajele), Receiver(cel care primeste mesajele) si Content(mesajul trimis).



- Diagrama de clase cu attribute și metode



4.6. Cazuri de testare

Cateva dintre cazurile posibile a aplicatiei, am luat in considerare urmatoarele cazuri de testare a aplicatiei:

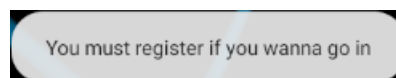
- Utilizatorul incearca sa se autentifice cu un Username si Password care nu se afla in baza de date.
- Utilizatorul nu introduce niciun caracter in campurile Username si Password, la autentificare sau la inregistrare in aplicatie.
- Utilizatorul doreste sa-si faca un cont cu numele "Admin".
- Utilizatorul foloseste un Username care se afla deja in baza de date.
- Utilizatorul nu introduce nici un caracter in campul in care se scriu mesajele.

5. Testarea propriu-zisa a aplicatiei

Pentru a testa buna functionare a aplicatiei trebuie luata fiecare situatie pe care un utilizator obisnuit o poate crea. Pentru a testa aceasta aplicatie, am luat in considerare mai multe scenarii posibile cum ar fi:

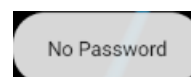
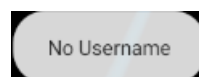
- **Utilizatorul incearca sa se autentifice cu un Username si Password care nu se afla in baza de date.**

Fiind o aplicatie in care este nevoie de o parte de intrregistrare pentru a o putea folosi, este nevoie ca utilizatorul sa se inregistreze mai intai daca doreste sa foloseasca acea aplicatie. Cu alte cuvinte, administratorul sau cel care a creat aplicatia sa stie de prezenta acestuia in mediul respectiv. Daca utilizatorul doreste sa foloseasca aplicatia cu un Username si Password care nu se afla in baza de date a aplicatiei, se va afisa un mesaj respectiv.



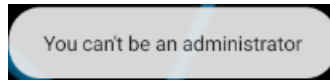
- **Utilizatorul nu introduce niciun caracter in campurile Username si Password, la autentificare sau la inregistrare in aplicatie.**

Pentru tratarea acestui caz, am decis sa afisez mesajul "No username", daca utilizatorul nu a introdus niciun caracter in campul Username, respectiv mesajul "No Password" daca nu a introdus nici un caracter in campul Password.



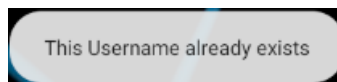
- **Utilizatorul doreste sa-si faca un cont cu numele "Admin".**

Deoarece exista un singur administrator a aplicatiei, acesta are un nume unic pe care un utilizator obisnuit nu poate sa-l foloseasca. Administratorul are un layer special in care are mai mult control asupra aplicatirii si a bazei dedate. Daca utilizatorul doreste sa-si faca un cont cu numele "Admin", va aparea un mesaj respectiv.



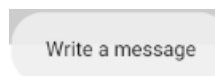
- **Utilizatorul foloseste un Username care se afla deja in baza de date.**

Ca orice alta aplicatie in care este nevoie de autentificare, trebuie luat in considerare si cazul in care utilizatorul foloseste un Username deja existent in baza de date a acelei aplicatii deoarece se creaza confuzie. Daca utilizatorul doreste sa-si faca un cont cu un nume deja existent, va aparea un mesaj repsectiv.



- **Utilizatorul nu introduce nici un caracter in campul in care se scriu mesajele.**

Pentru a nu crea confuzie sau introduce in baza de date a mesajelor date care nu au nici un continut la coloana "Content", am decis sa afisez un mesaj respectiv pentru a avertiza utilizatorul.



6. Manual de instalare si utilizare

6.1. Manual de instalare

Ca orice aplicatie care foloseste sistemul de operare Android, aceasta aplicatie poate fi instalata la fel de usor ca orice alta aplicatie. Generarea fisierului ".apk" poate fi facuta foarte usor de catre mediul de programare folosit(InteliJ). Dupa care fisierul generat poate fi instalat pe orice telefon cu sistemul de operare Android prin selectarea fisierului.

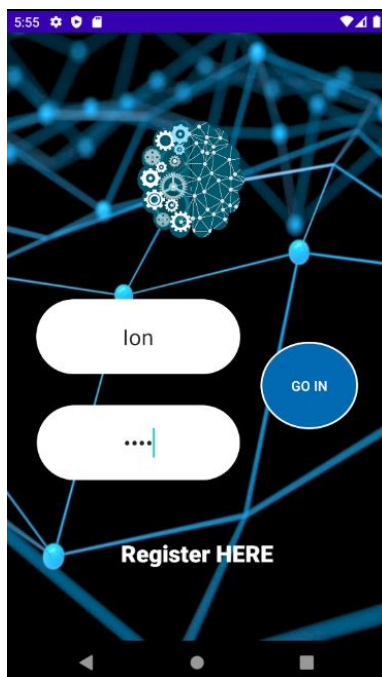
O alta varianta de instalare a aplicatiei poate fi cautarea acesteia in aplicatia Play Store si apasarea butonului "Install".

6.2. Manual de utilizare

În continuare voi prezenta pașii și posibilele situații cu care utilizatorul se poate întâlni

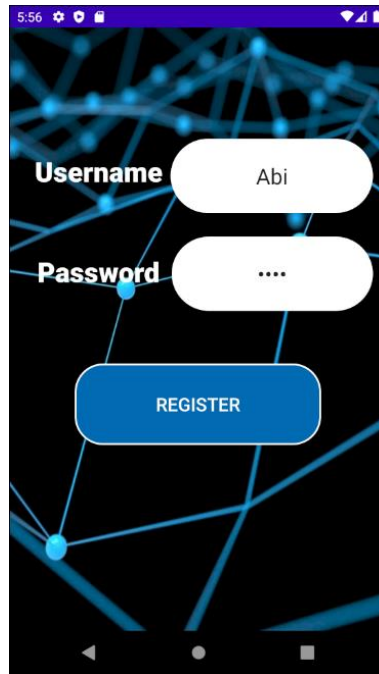
1. Fereastra de înregistrare:

Primul lucru ce se poate vedea la deschiderea aplicației este ecranul de înregistrare unde se află casutele unde se introduc numele și parola user-ului. Dacă acesta este înregistrat și se află în baza de date acesta va putea ”întra” în aplicație.



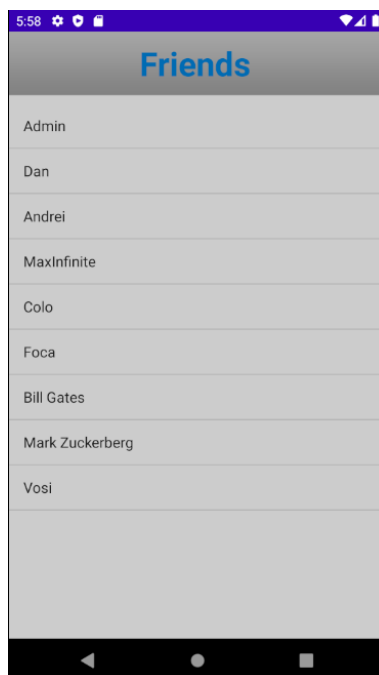
2. Fereastra de autentificare:

Dacă utilizatorul nu are un cont ce se află în baza de date, acesta poate să-și creeze unul apăsând pe textul ”Register HERE” din fereastra de înregistrare. După apăsarea acesteia se deschide fereastra de autentificare unde utilizatorul își poate introduce un nume unic și parola. Dacă acesta introduce un username care este deja folosit, se va afișa un mesaj pe ecran.



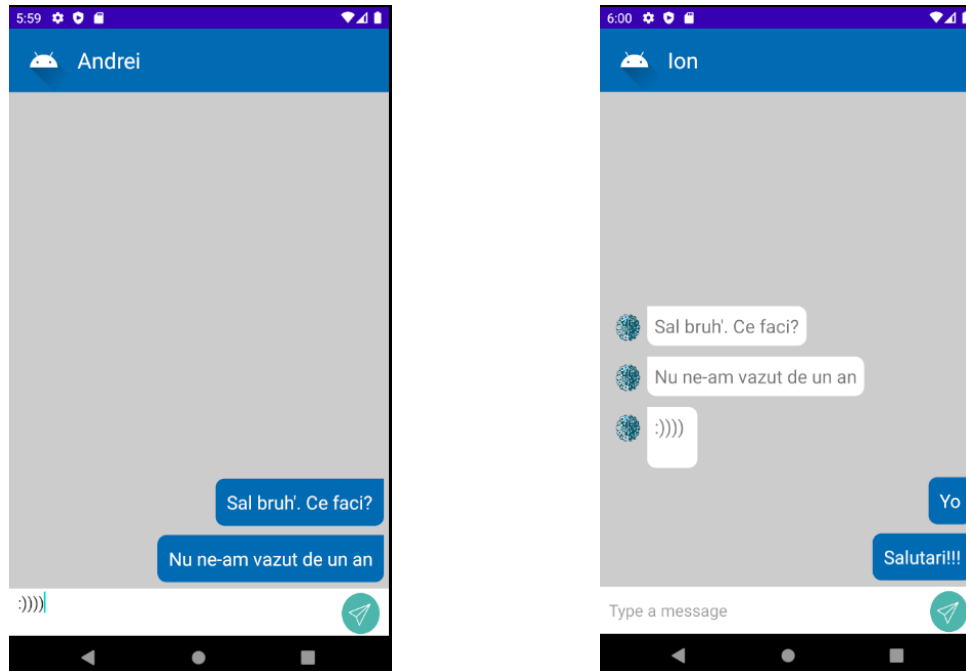
3. Fereastra cu lista de prieteni:

O data ce utilizatorul a introdus datele in campurile de Username si Password, acesta poate vedea toti utilizatorii din baza de date dupa care poate selecta unul dintre acestia si sa incepe sa comunice cu el.



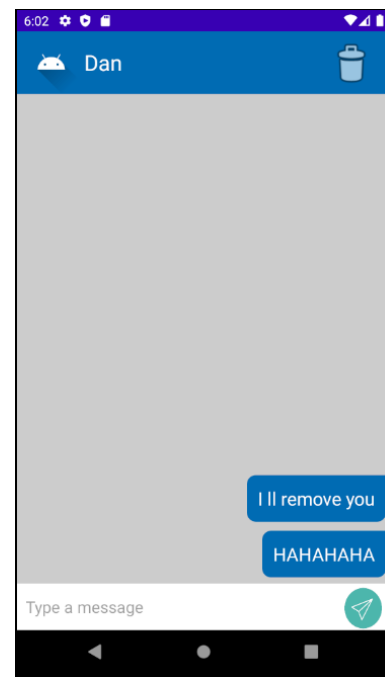
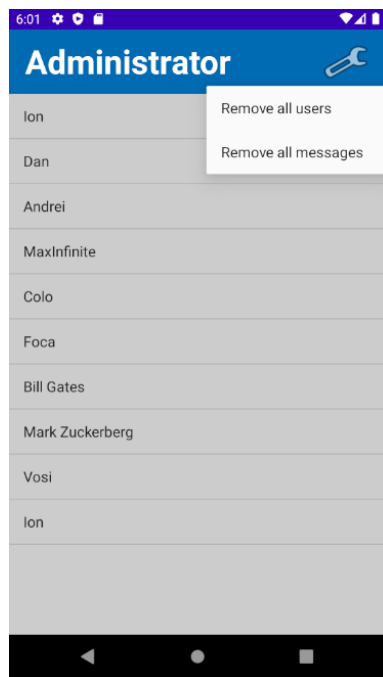
4. Fereastra in care utilizatorul trimite mesajele:

Dupa selectarea user-ului cu care se doreste comunicarea, se deschide fereastra unde se scriu si se trimit mesajele. Utilizatorul poate scrie orice mesaj. Mesajul respectiv este introdus in baza de date si poate fi vazut doar de cel care a primit mesajul. In stanga Andrei tocmai a trimis mesaje lui Ion si in dreapta se poate vedea cum vede Ion mesajele primite



5. Fereastra cu lista de prieteni si de mesaje a administratorului:

Administratorul este unic in aplicatie. Acesta se poate inregistra cu numele "Admin" si parola "1234". Acesta are ferestre diferite fata de utilizatorii obisnuiti, mai exact in fereastra in care se afiseaza utilizatorii(stanga) acesta are o iconita in plus. Acesta are optiunile de stergere a tuturor utilizatorilor din baza de date, cu exceptia lui, si de a sterge toate mesajele trimise intre utilizatori. In fereastra in care se trimit mesajele, administratorul mai are posibilitatea de a sterge userul respectiv din baza de date prin apasarea iconitei din dreapta sus.



7. Concluzii si dezvoltari ulterioare

In concluzie, alegerea acestui proiect adica dezvoltarea unei aplicatii a avut scopul de a intelege care sunt bazele tuturor aplicatiilor de pe telefoane si care sunt pasii care pot fi urmati pentru a putea dezvolta o asemenea aplicatii. Mai mult de atat am putut aprofunda folosirea limbajului de programare Java in majoritatea aplicatiilor existente si librariile existente care pot dezvolta asemenea aplicatii. Majoritatea aplicatiilor folosesc si limbajul de programare XML pentru a putea da o forma aplicatiei si a putea face interactiunea dintre utilizator si aplicatie mult mai uoara.

Cateva dintre dezvoltarile ulterioare la care poate fi supusa aplicatia sunt:

- Stergerea unui anumit mesaj selectat de utilizator
- Posibilitatea de a avea anumiti utilizatori din baza de date in lista de prieteni
- Posibilitatea de a avea o fotografie de profil
- Introducerea mai multor date la autentificarea in aplicatie
- Tinirea minte a utilizatorului care a folosit aplicatia ultima data(Remind me)
- Posibilitatea de a trimite si fotografii sa vieoclipuri ca mesaje
- Crearea de grupuri
- Posibilitatea de a suna si a sustine convorbiri telefonice cu persoana respectiva



8. Bibliografie si webografie

<https://www.codejava.net>

<https://www.w3schools.com/>

<https://www.geeksforgeeks.org/>

<https://www.youtube.com/>