***Documentație Proiect Android-Aplicație suport pentru pregătirea examenului auto***

Student: Stoian Maria Alina

Grupa:1083, seria D

**1.Introducere:**

*Definirea problemei:*

Aplicația este menită să vină în ajutorul persoanelor care vor să susțină examenul scris pentru obținerea permisului de conducere (categoria B).

*Definire obiective:*

Proiectul își dorește să pună bazele generale ale codului rutier, venind în ajutor cu diverse funcționalități precum: o secțiune specială de învățare ( indicatoare,întrebări ), o secțiune pentru a putea vizualiza întrebările de top preluate de pe internet și o secțiune care să-l obișnuiască pe utilizator cu întrebările, definită printr-un test cronometrat. Secțiunea ce oferă partea de verificare este cronometrată, deci utilizatorul poate vedea cât de rapid răspunde la întrebări, de asemena după fiecare întrebare i se afișează răspunsul corect și în final punctajul acumulat. Secțiunea de învățare dorește să-l treacă pe utilizator prin partea de teorie.

*Definire cadru general:*

Aplicația poate fi folosită oriunde, în orice mediu, este gratuită și nu necesită crearea unui cont de utilizator. Ea poate fi utilizată în orice situație și ca obiect de relaxare în pauză, sau intensiv pentru învățare.

**2.Descriere soluție:**

Obiectivul principal al proiectului este acela de învățare și verificare a cunoștințelor, astfel încât pentru atingerea acestuia aplicația se împarte în două secțiuni principale: mediul de învățare și mediul de verificare. Secundar, când există acces la internet, utilizatorul poate vedea întrebările de top regăsite cel mai des în examenul auto. Aplicația nu necesită crearea unui cont, se deschide direct având butoane specifice fiecărei secțiuni: buton pentru începerea testului de verificare, buton pentru a intra în mediul de învățare și un buton pentru accesul la întrebările de pe internet.

În secțiunea de verificare este afișat timpul rămas, punctajul acumulat până în prezent cât și întrebările din test. După fiecare răspuns trimis la întrebare ni se afișează cu verde pe ecran care era răspunsul corect și apoi se trece la următoarea întrebare. Dacă ieșim mai devreme din test ni se afișează punctajul acumulat până la dorința noastră de părăsire a testului, iar dacă ne-a expirat timpul aplicația va opri automat testul. De asemenea, este păstrat mereu punctajul maxim acumulat pentru a știi cât am făcut cel mai mult și să motiveze autodepășirea.

În secțiunea de învățare este prezent un meniu ce oferă mai multe avantaje. Dacă utilizatorul face click pe “indicatoare” în meniul aplicației găsește o varietate de imagini ale indicatoarelor rutiere împreună cu denumirea acestora și explicația. Apăsând pe iconiță de plus, utilizatorul are posibilitatea de a-și adaugă singur întrebări care i se par mai grele sau mai inedite în baza de date și apoi se poate verifica deoarece ele vor apărea automat și în secțiunea de învățare. Apăsând pe “întrebări” din meniu vom găsi afișate toate întrebările din baza de date a aplicației. Făcând click pe întrebare va apărea un meniu suplimentar care îi oferă posibilitatea de a șterge întrebări pe care le consideră învățate/știute  sau să modifice întrebările pe care acesta vrea să le modifice. “ReportOne” și “ReportTwo” oferă posibilitatea utilizatorului de a-și salva fișiere ce descriu liste cu întrebări. În “charts” găsește niște grafice pentru a verifica numărul de întrebări din fiecare categorie. Aceste informații sunt utile dacă utilizatorul își dorește să-și mai adauge întrebări din categorii care se află în minoritate. “Home” îl întoarce în ecranul principal al aplicației, unde găsește și butonul de acces al întrebărilor de top.

**3.Implementare soluție:**

În vederea creării celor două secțiuni principale a fost creată o bază de date cu două tabele ce conțin întrebări și categoriile din care fac parte acestea. Întrebările sunt extrase din baza de date și afișate cu ajutorul unor diferite controale de tip textview în secțiunea de test. Cu ajutorul controalelor radio utilizatorul poate bifa răspunsul pe care acesta îl consideră corect și cu ajutorul butoanelor se poate deplasa înainte cu testul. Alte controale de tip textview ajută în vederea punctajului și numărului de întrebări parcurse. De asemenea implementăm un countdowntimer care va cronometra și își va schimbă culoarea atunci când mai rămâne doar un minut. Apelându-se apoi metoda onDestroy pentru a închide cronometrul când activitatea a luat sfârșit.

S-au creat două clase model pentru instanțele din baza noastră de date (questions, categorie)  și o clasa finală (quizcontract) ce va conține constantele pentru numele coloanelor și tabelelor. Mai departe, am creat o clasa (quizdbhelper) ce reprezintă baza de date. Tot aici sunt create tabelele și sunt populate cu ajutorul unor metode definite în clasa. Mai departe pentru a accesa înregistrările din baza de date, se fac query-uri cu ajutorul unui cursor și apelând metoda getReadableDatabase urmată de rawQuery, trecând prin fiecare coloană din tabel cu metoda getColumnIndex pentru a accesa informațiile din fiecare linie. Vom returna aceste date sub formă de arrayList/List  de obiecte specifice. Acestea vor fi mai târziu utilizate pentru afișarea în aplicație a informațiilor, de exemplu pentru întocmirea rapoartelor, afișarea întrebărilor în mediul de învățare, afișarea întrebărilor în mediul de verificare. Cu ajutorul metodei shuffle întrebările vor fi mereu amestecate, se compară butonul radio selectat (selectedRadioButton) cu numărul răspunsului corect pentru verificare, se compară mereu numărul de întrebări parcurse cu dimensiunea array-ului, când nu ne mai rămân întrebări apelăm metoda finish() pentru a închide activitatea. Vom transmite obiecte și parametrii între activități și fragmente. Pentru salvarea punctajului  vom începe activitatea cu metoda startActivityForResult și apoi apoi vom seta scorul rezultat în QuizActivity cu setResult. Acesta va fi afișat în StartingScreenActivity și va fi salvat și în fișierele de preferințe. Fișierele de preferințe au ca scop reținerea informațiilor pentru a evita pierderea progresului făcut de utilizator.

*Structura tabelelor din baza de date:*

a. quiz\_questions:

- id: cheie primară; se autoincrementează la fiecare înregistrare introdusă.

- question: textul întrebării.

- option1: text răspuns A.

- option2: text răspuns B.

- option3: text răspuns C.

- answer\_nr: răspunsul corect.

- categorie\_id: id-ul categoriei corecte, cheie externă.

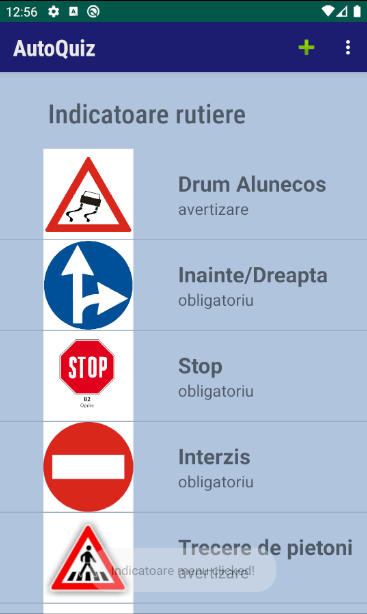
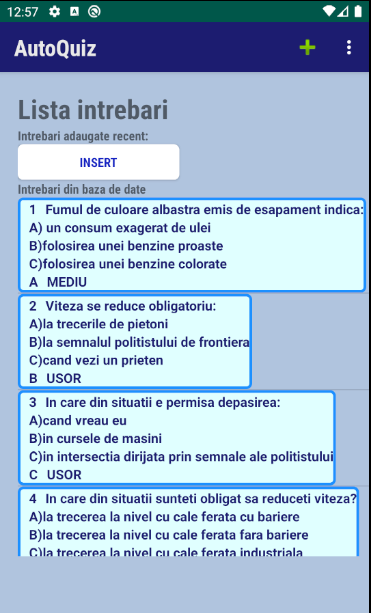
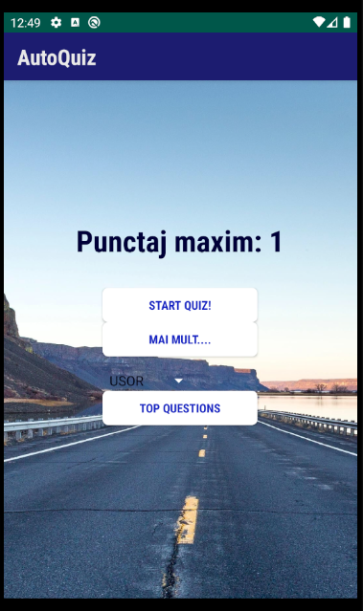
b. quiz\_ategorii:

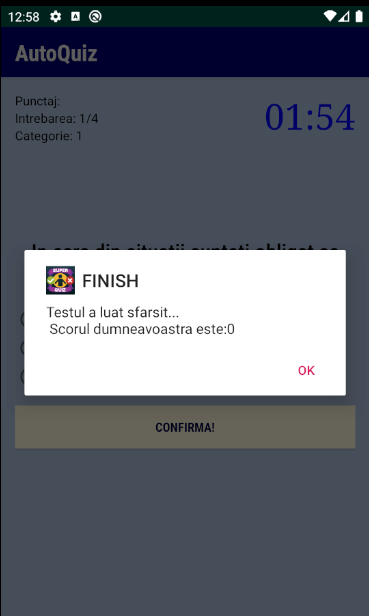
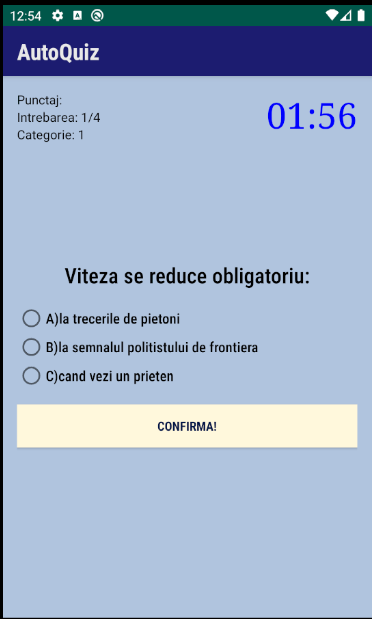
* id: cheie primară; se autoincrementează la fiecare înregistrare introdusă.
* denumire: reprezintă tipul categoriei.

Ca userul să nu iasă din greșeală din test am implemenat o funcționalitate care îl avertizează pe utilizator să apese de două ori pentru a ieși din activitate. Metoda onBackPressed iese din activitate doar dacă apăsăm de două ori înapoi. Pentru afișarea informațiilor despre întrebări și indicatoare s-au creat adaptoare customizate.

Clasa Question va fi Parceble pentru a putea transmite obiecte de tip întrebare în bundle. Bundle ajută la transferul de parametrii între fragmente (putParcelable).De exemplu când dorim să înseram o întrebare nouă. Se fac tranzacții între fragmente pentru a putea accesa diferitele câmpuri din meniu. Asincron făcând http request accesăm internetul și parcurgem un fișier json cu întrebări top pentru a fi la dispoziția utilizatorului. În vederea obținerii graficelor, am folosit o clasa MyChart derivată din view în care cu ajutorul mai multor metode am desenat pe canvas și am setat culorile să fie afișate mereu diferit pe dreptunghiuri. Am utilizat mai multe permisiuni pentru a putea salva rapoartele sub formă de fișiere txt/csv.

Platforma android oferă multe avantaje și implementarea acestei aplicații este foarte manevrabilă, având deja tot ceea ce este necesar pentru a oferi un produs final. Sqlite este încorporat, partea de interfață se realizează foarte ușor, având nenumărate controale și multe moduri de a le customiza cât mai frumos. Ce este și mai plăcut e faptul că poți face și partea din spatele aplicației tot aici, rulând rapid pe emulator. Principalul avantaj consider că este debug-ul foarte specific pentru aplicații. Ce consider un dezavantaj ar fi layouturile care sunt foarte fine la modelare și faptul că trebuie mereu să iei în considerare poziția telefonului pentru aranjarea controalelor astfel încât să nu existe suprapuneri.





Pentru stilizarea aplicației am schimbat culorile temei default și am creat stiluri speciale pentru butoane si textview-uri.

**4.Concluzii:**

O aplicație de învățare și verificare se pliază cel mai bine ca o aplicație mobilă, fiind ușor de utilizat și de manevrat pentru oameni. Dacă proiectul ar fi fost web nu ar fi fost așa practic și la indemâna oricui, mai ales că majoritatea oamenilor dețin un smartphone în ziua de azi cu sistem de operare predominant android. Consider că implementarea mea nu este cea mai optimă din punct de vedere tehnic, existând metode mult mai eficiente de a face ceea ce am construit eu, aplicația mea face mult prea multe lucruri în firul principal fiind necesară mânuirea mai bună a firelor de execuție.

Aplicația ar putea deveni și mai practică dar s-ar dezvolta mai multe tipuri de teste sau dacă ar exista un mediu de învățare pentru și mai multe categorii: A1, A2, A, B, C etc. Un alt lucru interesant ar fi dacă s-ar crea diferite șcenarii cu situații din trafic și userul să-și introducă propriul răspuns, iar acel răspuns să fie analizat pe baza unui algoritm. Desigur, aplicația poate fi extinsă în multe moduri pentru a reprezenta cu adevărat un instrument cat mai bun de învățare și verificare.

**Bibliografie:**

[**https://developer.android.com/**](https://developer.android.com/)

[**https://codinginflow.com/tutorials/android**](https://codinginflow.com/tutorials/android)