

Dacian Goina
Universitatea de Vest din Timișoara
Facultatea de Matematică și Informatică
Informatică română
Anul 2

Deming

Programare pe dispozitive mobile

Data: 17.05.2021

Coordonator: lect. Dr. Liviu Octavian Mafteiu-Scai

Cuprins

Abstract	3
Introducere - scopul aplicației	3
Potențiali utilizatori	3
Contribuțiile autorului	4
Aplicații similare	4
Funcționalitate	5
Interfața grafică	9
Manualul utilizatorului	10
Documentație tehnică	14
Baza de date	23
Concluzii și direcții viitoare	25
Referințe	26

Abstract

Acest document conține documentația aplicației Deming, o aplicație de tip mobile pentru telefoanele cu sistem de operare Android. Aplicația a fost dezvoltată folosind Android Studio, este scrisă în limbajul Java și are scop lucrul cu sondajele de opinie. Documentul conține informații referitoare la scopul, funcționalitatea și utilizarea aplicației, contribuțiile autorului, structura aplicației, precum și direcțiile viitoare pentru dezvoltare aplicației.

Introducere - scopul aplicației

Scopul aplicației Deming este acela de a ajuta oamenii (firmele) care realizează sondaje de opinie. Concret: în ziua de azi sunt realizat sondaje de opinie pentru diverse subiecte, de exemplu alegeri: [1] ”cu cine ați vota dacă duminica viitoare...”, sondaje referitoare la fumat: [2] ”cât cheltuiți lunar în medie pe țigări”, sondaje referitoare la tendințe imobiliare: [3] ”având în vedere pandemia și carantina ați prefera în continuare să locuiți într-un apartament sau v-ar plăcea să aveți o casa” etc.

Colectarea datelor pentru astfel de sondaje nu se face prin mediul online [4] deoarece calitatea datelor obținute va fi foarte slabă – de ex. se poate abuza de multi-account (cineva răspunde cu mai multe conturi), sau nu sunt respectate condițiile pentru participarea la sondaj: de ex cineva sub 18 ani nu trebuie să răspundă la un sondaj referitor la alegeri.

Colectarea datelor se realizează prin mijloace fizice – ”interviu” direct cu o anumită persoană, fie cu persoane întâlnite în spații publice (ex. parcuri), fie prin vizită la domiciliu [1]. Practic, angajatul care merge pe teren să colecteze date vorbește cu mai multe persoane și notează răspunsurile date de acestea la întrebările din sondajul respectiv.

Pentru a nu mai nota răspunsurile primite pe hârtie, acestea sunt notate direct în aplicație, de acolo sunt trimise într-o bază de date centralizată, ulterior ele vor putea fi folosite pentru diverse prelucrări (de ex. generarea de grafice folosind datele colectate, verificare activității angajaților etc). Mai mult, pentru introducerea răspunsurilor (input) primite de la oameni, nu este nevoie de scrierea de la tastatura virtuală a telefonului, ci direct apăsarea pe butoanele (afișate pe ecran) corespunzătoare răspunsurilor, acest lucru oferind mai multă flexibilitate și economie de timp.

Potențiali utilizatori

Aplicația se adresează firmelor care realizează sondaje de opinie. Aplicație permite crearea de sondaje, iar apoi colectarea datelor pentru aceste sondaje. Pentru a colecta date pentru sondaje sunt trimiși pe teren mai mulți angajați care trebuie să discute cu cetățenii și să introducă răspunsurile primite în aplicație.

Pentru acest tip de activitate și pentru utilizarea aplicației, o persoană care dorește să facă aceste lucruri nu are nevoie neapărat de studii superioare, de multe ori fiind suficiente studiile medii [5]. Omul potrivit ar fi cineva cu vârsta între 18 și 55 de ani, care este prietenos, răbdător, și căruia îi place să discute cu oamenii (deoarece trebuie să comunice cu ei, să le adreseze întrebările din sondaj, să nu se enerveze ușor dacă trebuie să repete aceeași întrebare de mai multe ori).

Evident, persoana care realizează aceste activități (colectarea datelor) trebuie să știe să folosească un smartphone. De exemplu, o astfel de activitate (colectarea de date pentru sondaje) ar putea fi realizată chiar de elevii care muncesc part-time pe perioada verii pentru a câștiga câțiva bani în plus.

Contribuțiile autorului

În Google Play există deja aplicații similare cu Deming, prin urmare unele funcționalități ale acestei aplicații există deja în alte aplicații. Totuși, aplicația Deming conține unele funcționalități care sunt contribuții originale ale autorului.

Pe lângă codul aplicației, care a fost scris în întregime de către autor, alte contribuții originale sunt:

1. Metodă pentru a exporta datele colectate pentru sondaje sub forma unui CSV. Acest fișier este mult mai practic de folosit decât unul Excel deoarece este un fișier mai apropiat de formatul .txt, prin urmare poate fi citit cu ușurință de alte aplicații (în comparație cu un fișier excel care este destul de complex)
2. Metodă pentru calcularea pontajului angajaților – această funcție permite ca pentru un anumit angajat (persoană care merge să colecteze date) să se calculeze câte înregistrări a colectat într-o anumită lună, precum și sondajele pentru care a colectat.
3. Vizualizarea rezultatelor sondajelor folosind diagrame PieChart ("plăcintă"), care ilustrează mult mai bine acest tip de rezultate în comparație cu alte diagrame, de ex. BarChart sau LineChart
4. Structura stocării datelor – în baza de date avem tabele care stochează înregistrările (răspunsurile) primite pentru sondaje, totuși trebuie să stocăm și întrebările efective și opțiunile de răspuns pentru fiecare sondaj. În general, când folosim date care nu au o structură anume am putea foarte ușor să le stocăm folosind baze de date de tip NOSQL, totuși le putem grupa / concatena într-un anumit mod / după un anumit model care să permită stocarea lor în baze de date de tip SQL. Informațiile menționate anterior (întrebările, opțiunile de răspuns – acum nu mă refer la înregistrările efective) vor fi stocate tot într-o bază de date SQL dar trebuie grupate într-un mod specific care să permită ulterior și degрупarea lor pentru a putea fi folosite când avem nevoie de ele în aplicație. Pentru o aplicație de acest tip consider că este mai bine să folosesc o bază de date tip SQL deoarece avem nevoie ca datele să fie grupate / structurate în niște tabele specifice, să putem localiza ușor datele (să știm rapid din ce tabel să le luăm), astfel se obține mai multă performanță / eficiență, față de bazele de date NOSQL, care în general nu sunt structurate, ceea ce ar fi cam nepotrivit pentru această situație – pentru acest tip de aplicație (a se vedea **pagina 21**)

Aplicații similare

Așa cum am menționat mai sus, există deja aplicații care permite colectarea datelor.

KoboCollect [6] este aplicație care poate fi folosită pentru crearea sondajelor și colectarea datelor pentru acestea. Aplicația permite creare de întrebări cu opțiuni de răspuns, verificarea datelor colectate. Această aplicație permite de asemenea generarea de grafice de tip BarChart pe baza datelor colectate.

O altă aplicație similară cu KoboCollect este Teamscope [7], aceasta permite crearea de formulare, colectare de date, precum și partajarea datelor în diferite moduri. Răspunsurile sunt introduse în aplicație prin selectarea lor folosind checkbox-uri.

Funcționalitate

În continuare sunt prezentate câteva diagrame UML care descriu funcționalitățile aplicației.

Diagrama cazurilor de utilizare

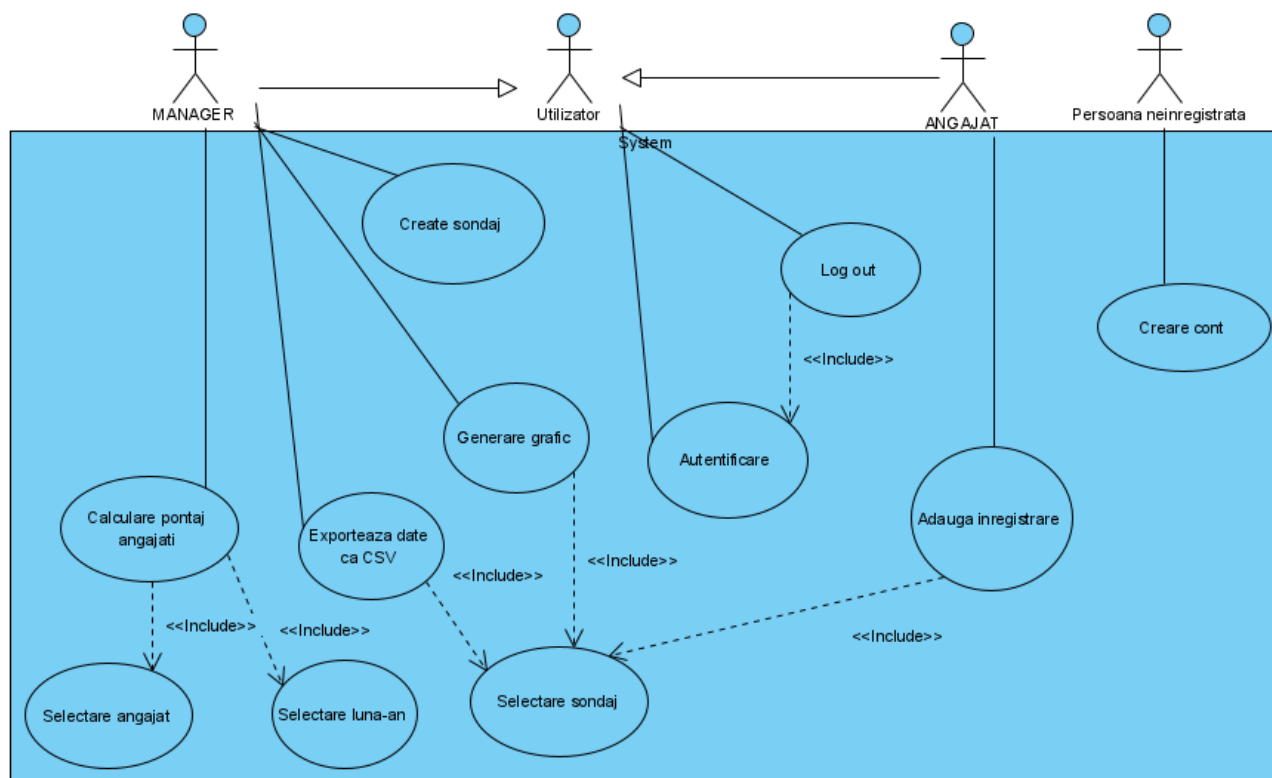
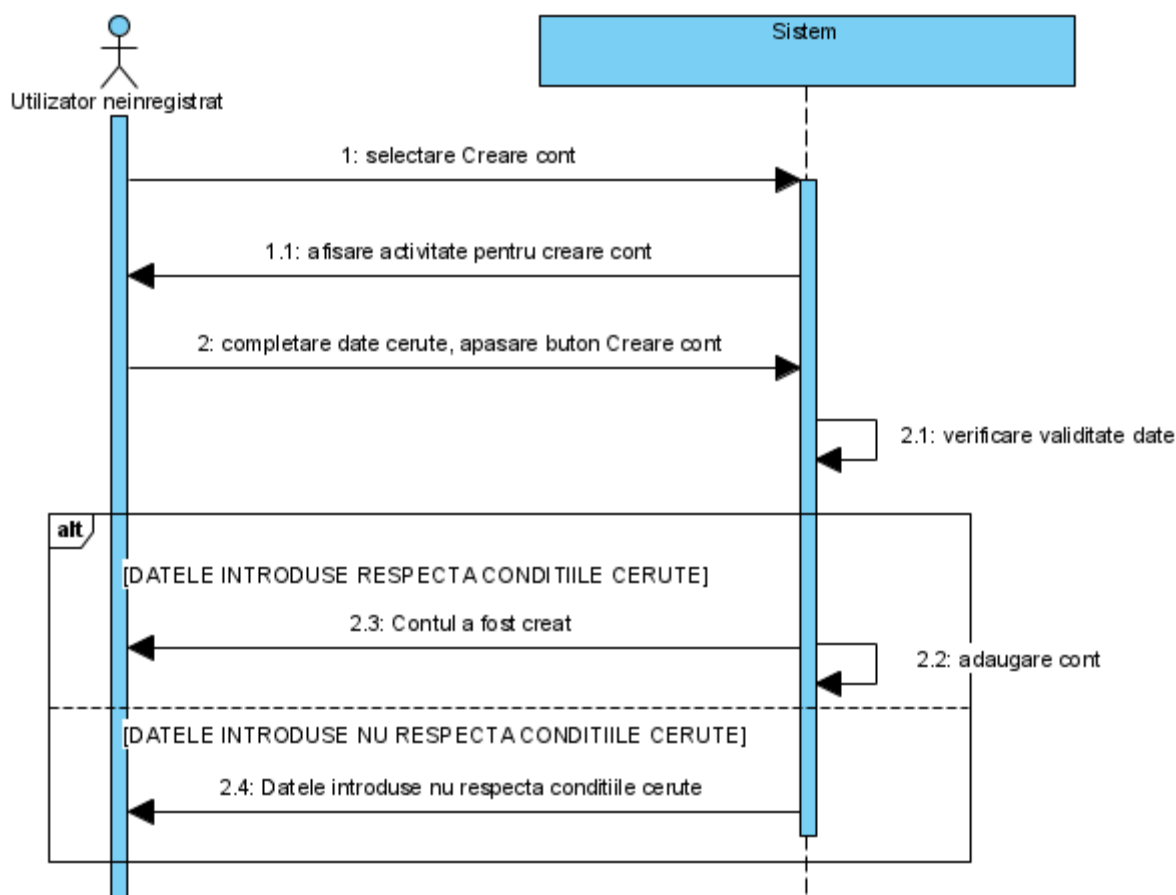
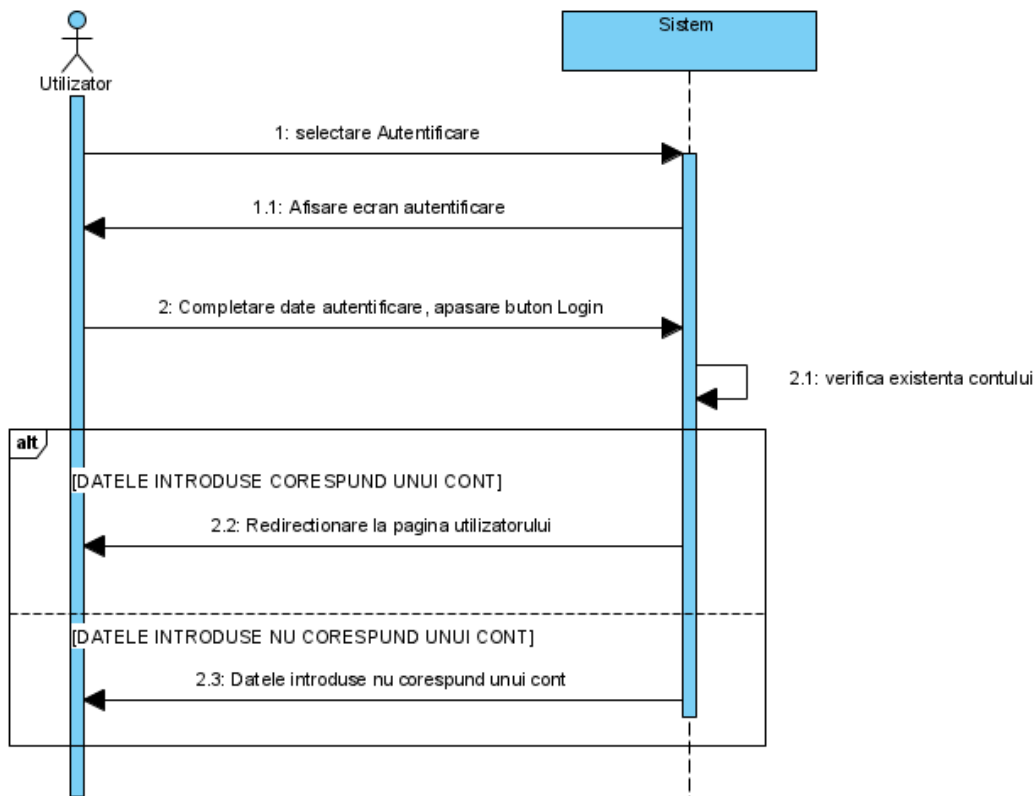


Diagrama de secvențe pentru cazurile de utilizare:

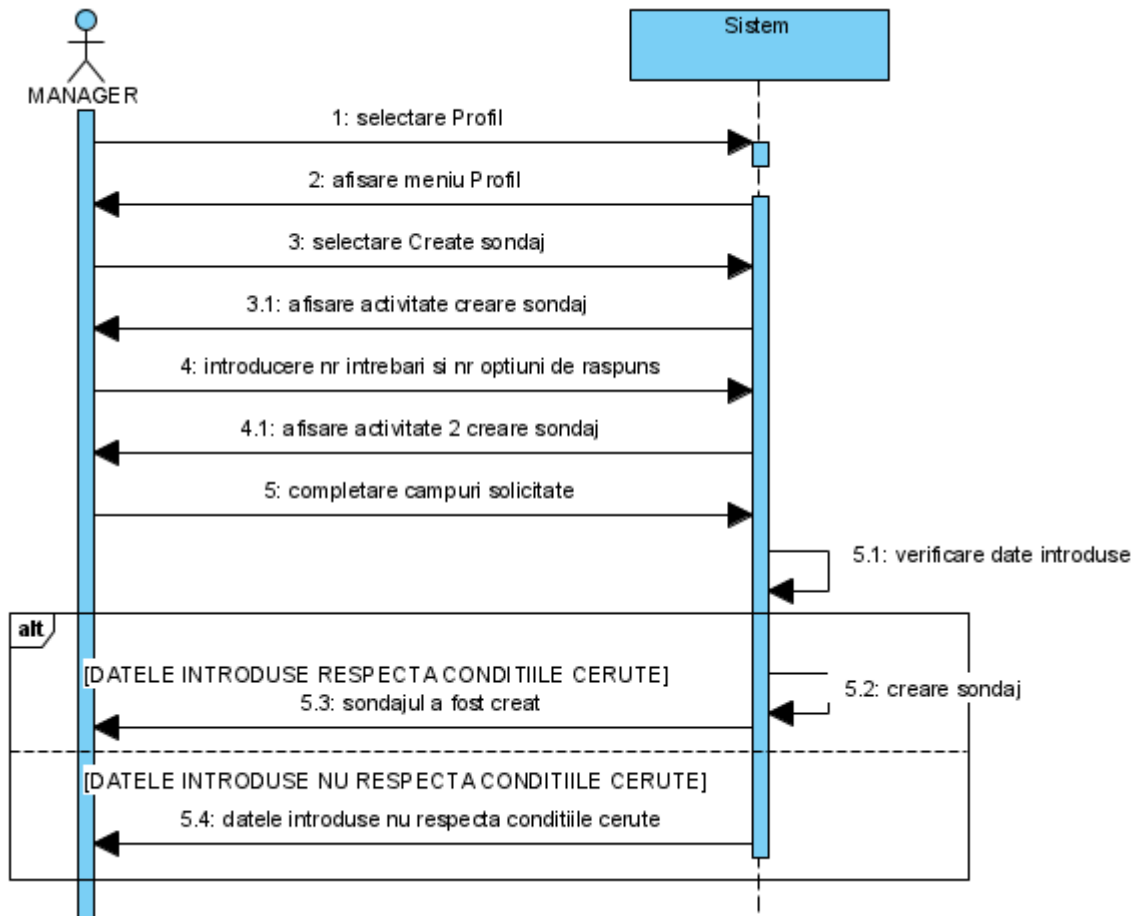
Creare cont:



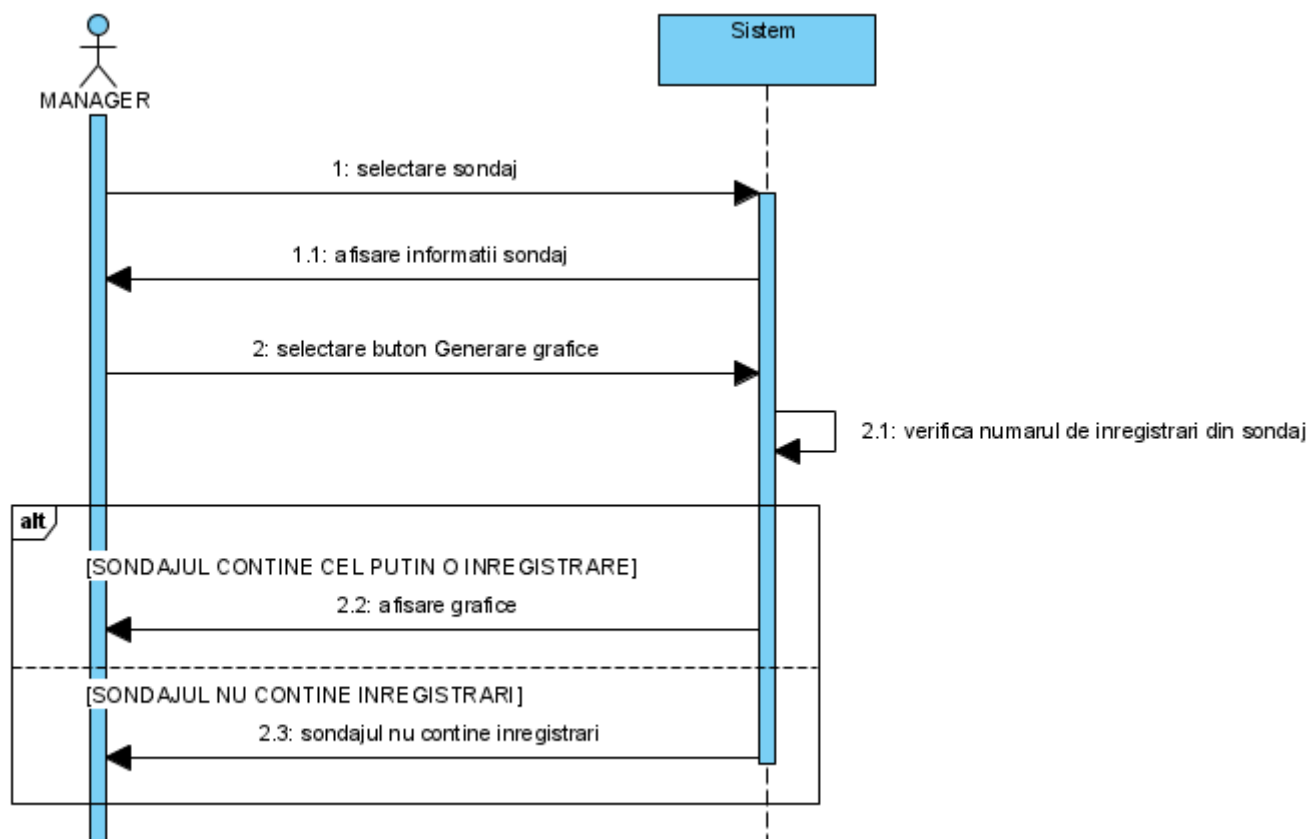
Autentificare:



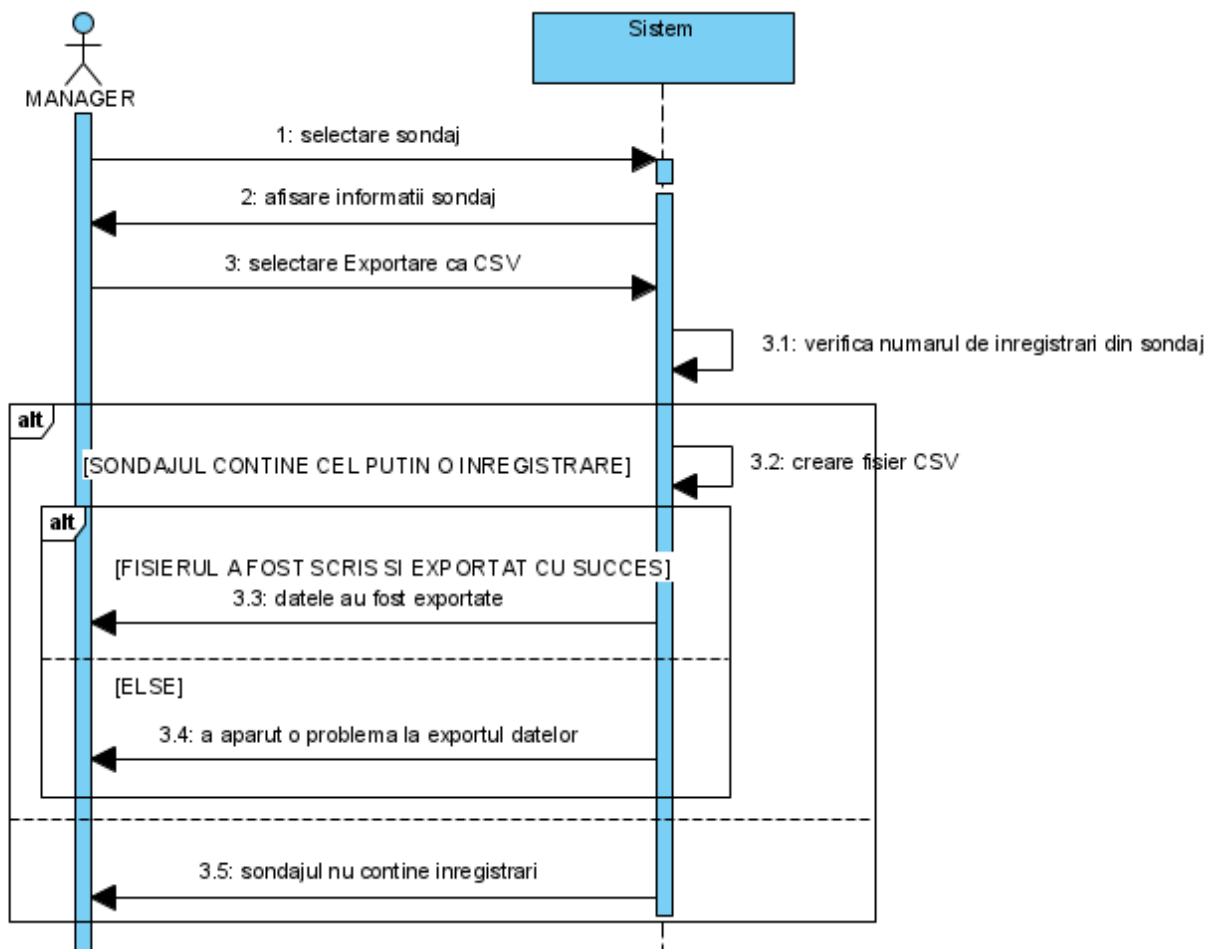
Creare sondaj:



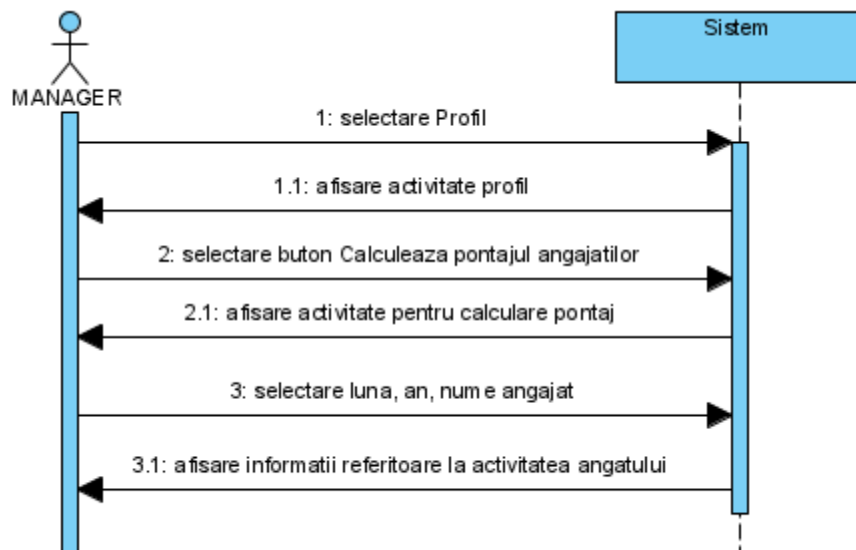
Generare grafice



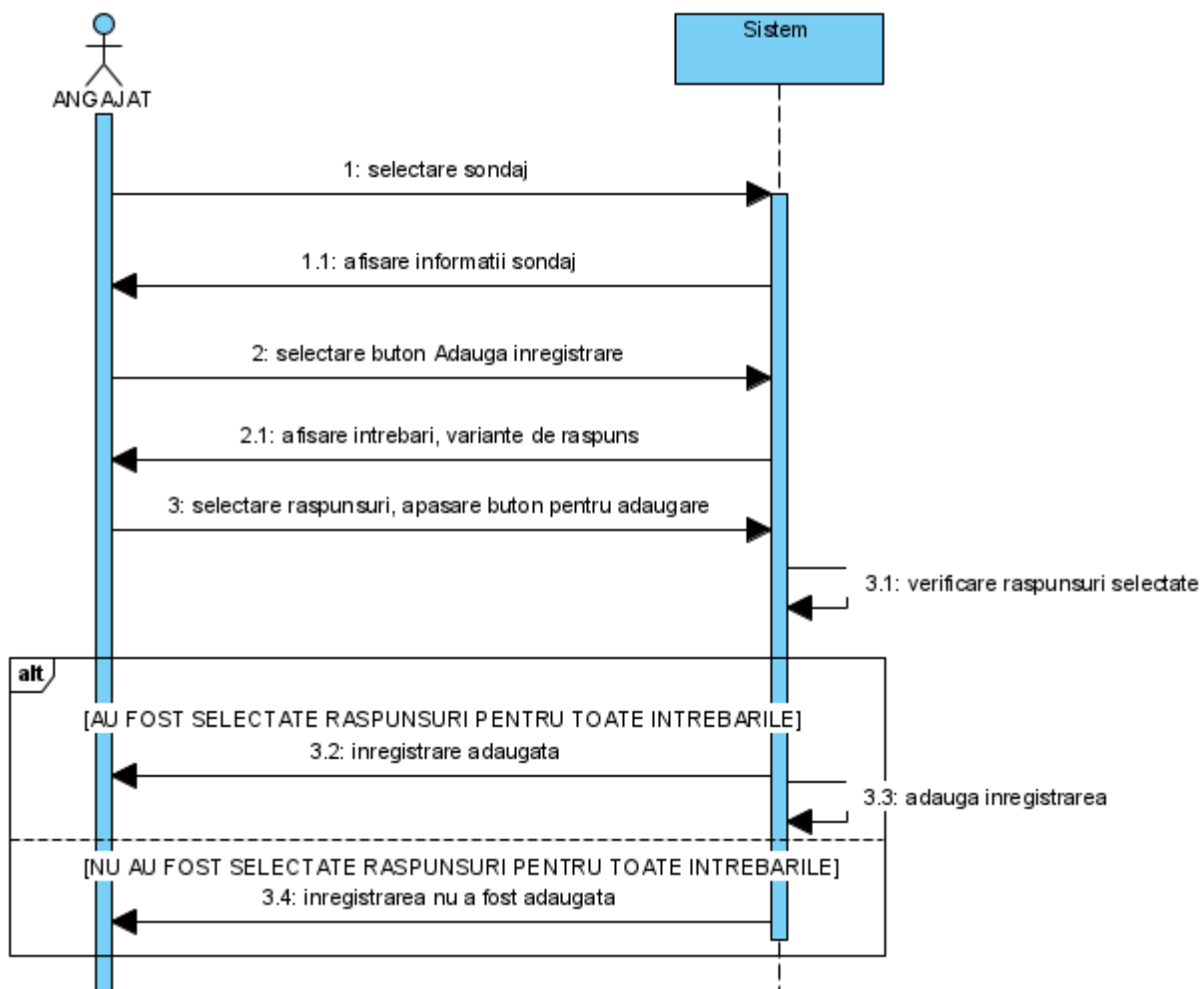
Exportare date în fișier CSV



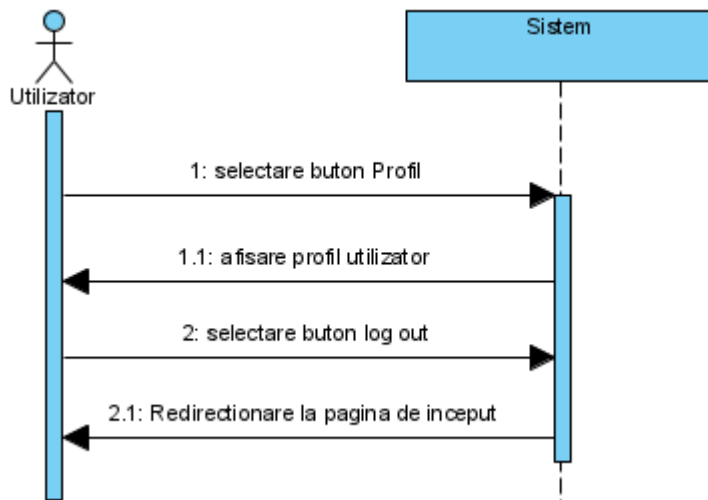
Calculare pontaj angajați



Adăugare înregistrare



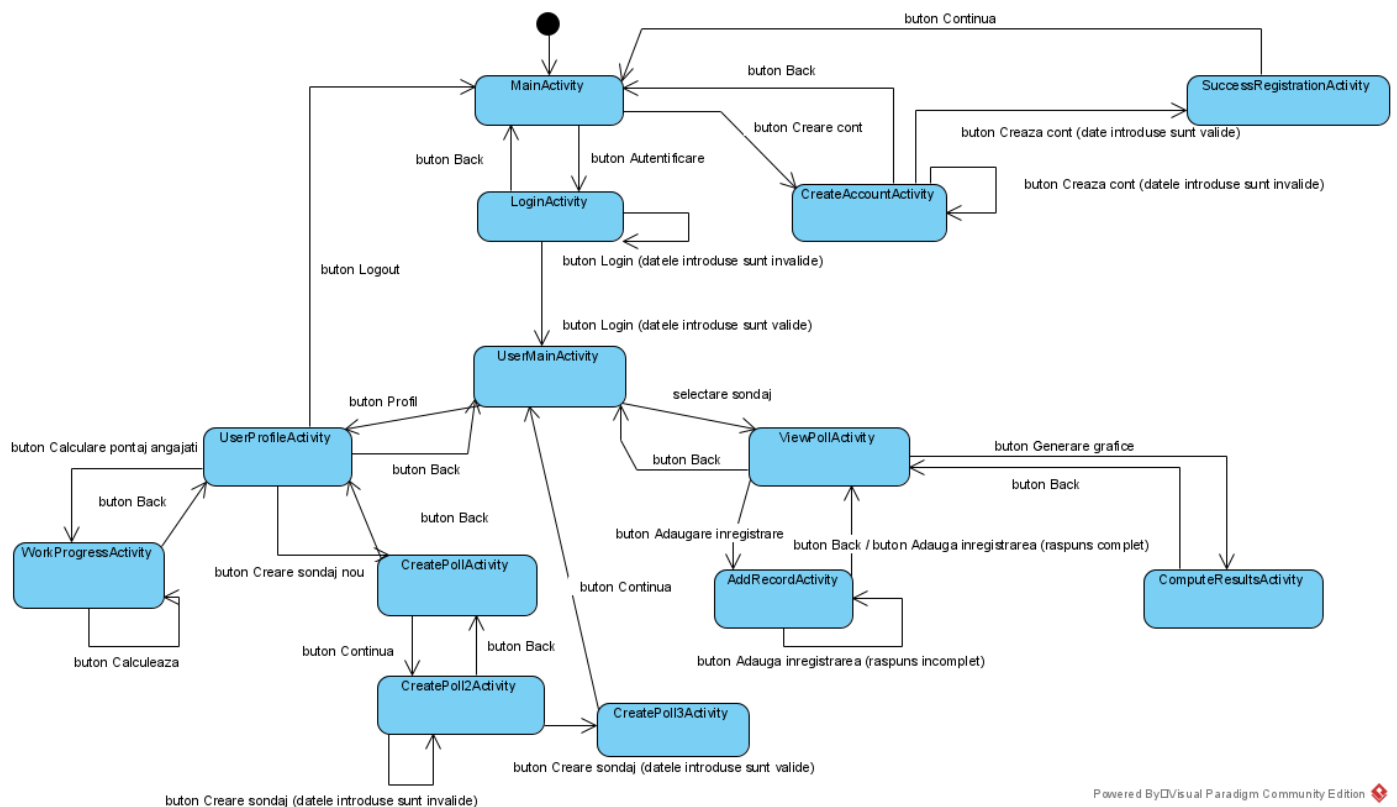
Log out:



Interfața grafică

În continuare este ilustrată diagrama de stări care prezintă parcurgerea ferestrelor (activităților).

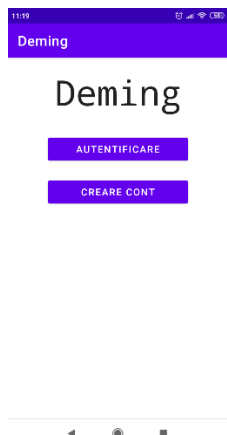
Diagrama de stări:



Manualul utilizatorului

Această secțiune conține instrucțiuni referitoare la utilizarea aplicației din perspectiva unui utilizator comun (normal), cineva care nu are cunoștințe în programare.

La deschiderea aplicației este afișată activitatea principală - ecranul principal



Creare cont: pentru a utiliza efectiv aplicația este nevoie de un cont care se poate crea gratuit. Pentru a crea un cont se apasă butonul Creare cont ca va deschide activitatea corespunzătoare, în această activitate trebuie complementate datele solicitate :

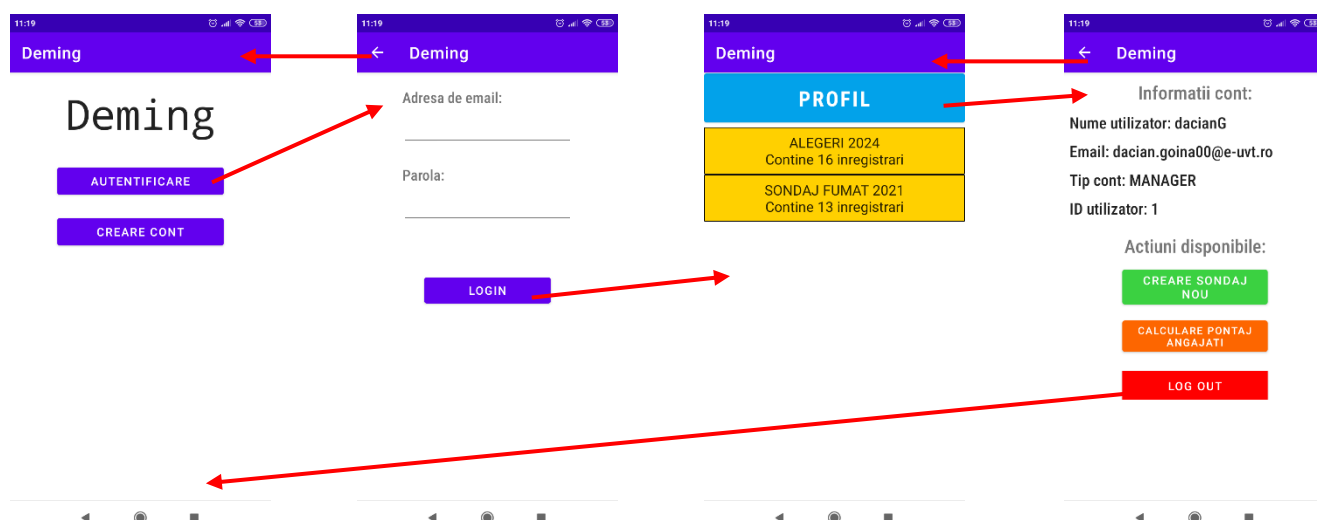
- adresă de email
- nume de utilizator
- parolă
- repetare parolă

Datele introduse în câmpuri trebuie să respecte anumite condiții: de ex. pentru numele de utilizator și parolă, trebuie să conțină minim 3 caractere fiecare, parola trebuie să conțină atât litere mari cât și litere mici. În cazul în care condițiile nu vor fi îndeplinite, vor fi afișate mesaje corespunzătoare care să ajute utilizatorul să editeze datele introduse astfel încât acestea să fie acceptate. În mod normal (default) contul creat va fi de tip ANGAJAT. Pentru a modifica contul să fie de tip MANAGER trebuie contactat administratorul pentru a seta acest lucru folosind baza de date. Mai jos este ilustrat interacțiunea dintre activitățile implicate în procesul de creare a unui cont.



Notă: pentru a realiza trecerea între unele activități trebuie îndeplinite anumite condiții – de ex. în cazul creării unui cont, pentru a crea contul cu succes și a ne întoarce la activitatea principală trebuie să completăm câmpurile cerute; aceste condiții nu au mai fost scrise pe săgețile roșii deoarece ele apar în diagrama de stări de mai sus și vor mai fi explicate în partea de documentație tehnică

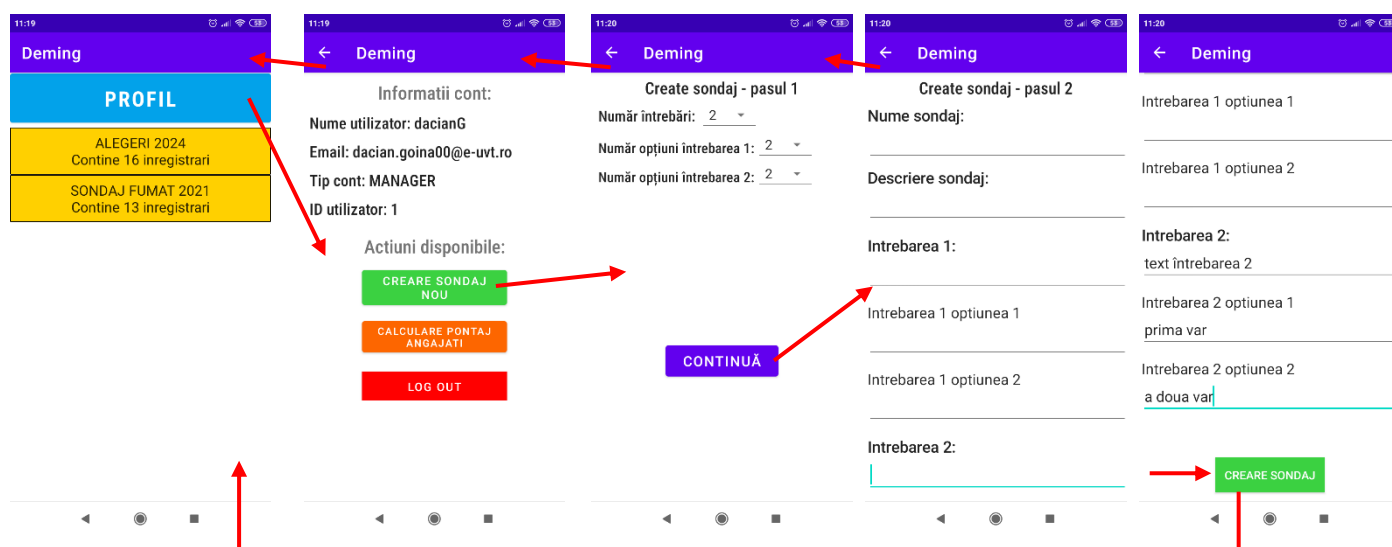
Autentificare: dacă avem un cont ne putem autentifica și putem folosi efectiv aplicația. Se apasă butonul Autentificare, utilizatorul este redirecționat la pagina de logare, se completează datele cerute, apoi se apasă butonul Login. În cazul în care datele introduse corespund unui cont existent, atunci logarea se va realiza și se va deschide o nouă activitate. În cazul în care datele introduse nu corespund unui cont existent se va afișa un mesaj care să informeze utilizatorul despre acest lucru. Dacă suntem logați în aplicație putem efectua diverse acțiuni în funcție de tipul contului (ANGAJAT sau MANAGER). Toți utilizatorii, indiferent de tipul contului, vor putea vedea numele sondajelor existente precum și numărul de înregistrări (records) ale acestor sondaje. Mai jos este un exemplu care ilustrează pași pentru autentificare și deconectare din aplicație.



Creare sondaj: utilizatorii care au cont de tip MANAGER pot crea sondaje noi. Pentru a face acest lucru, se apasă butonul Profil, acest lucru va deschide o nouă activitate iar în această activitate se apasă butonul Creare sondaj nou. În continuare trebuie specificat numărul de întrebări (între 2 și 7) pe care vrem să îl conțină noul sondaj, precum și numărul de opțiuni de răspuns pentru fiecare întrebare (între 2 și 7). După specificarea informațiilor anterioare se apasă butonul Continua, aceste deschide o nouă activitate unde vor trebuie completate datele despre noul sondaj:

- titlul sondajului
- descrierea sondajului
- corpul (enunțul) pentru toate întrebările din sondaj
- opțiunile de răspuns pentru toate întrebările din sondaj

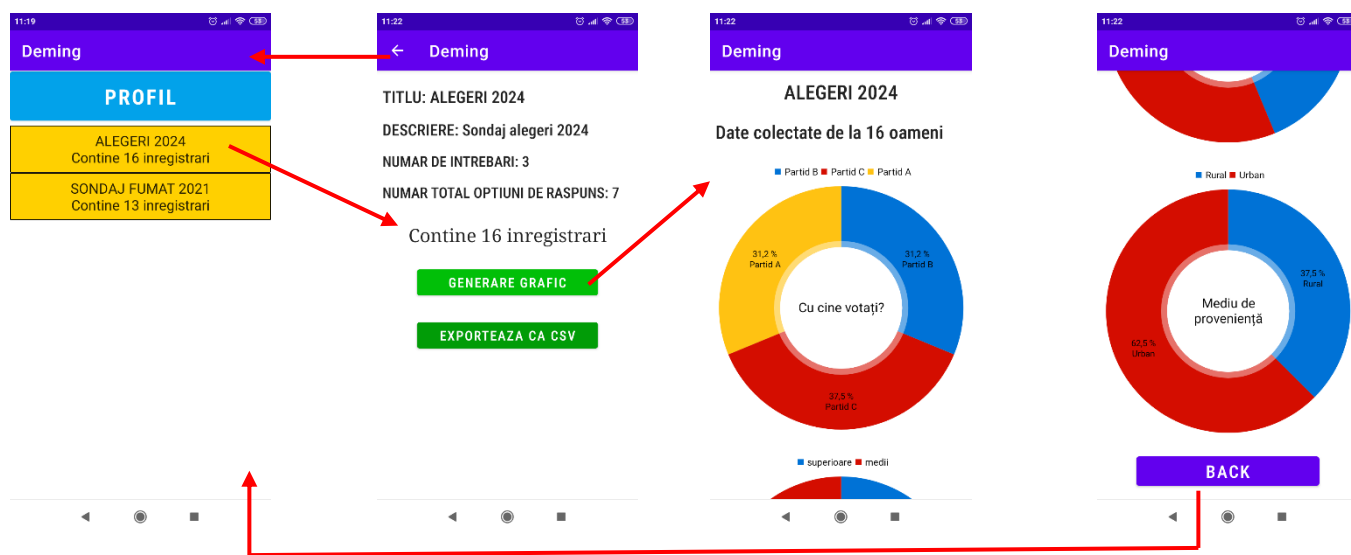
Aceste date trebuie să respecte anumite condiții, de ex. numele sondajului trebuie să conțină cel puțin două cuvinte. În cazul în care datele introduse nu respectă condițiile cerute, se va afișa un mesaj care va informa utilizatorul despre greșelile din textul introdus și îl va ghida pentru a le îndrepta și pentru a completa procesul de creare a unui sondaj. Mai jos este ilustrat procesul de creare al unui sondaj



Generare grafice: utilizatorii care au cont de tip MANAGER pot efectua diverse operații asupra datelor colectate pentru sondajele existente. Se selectează sondajul dorit, iar dacă sondajul conține cel puțin o înregistrare atunci se pot efectua următoarele două acțiuni:

- Export ca CSV: exportează datelor colectate sub forma unui fișier CSV
- Generare grafic : generarea de grafice folosind datele colectate pentru sondaj, astfel se pot vizualiza rezultatele sondajului.

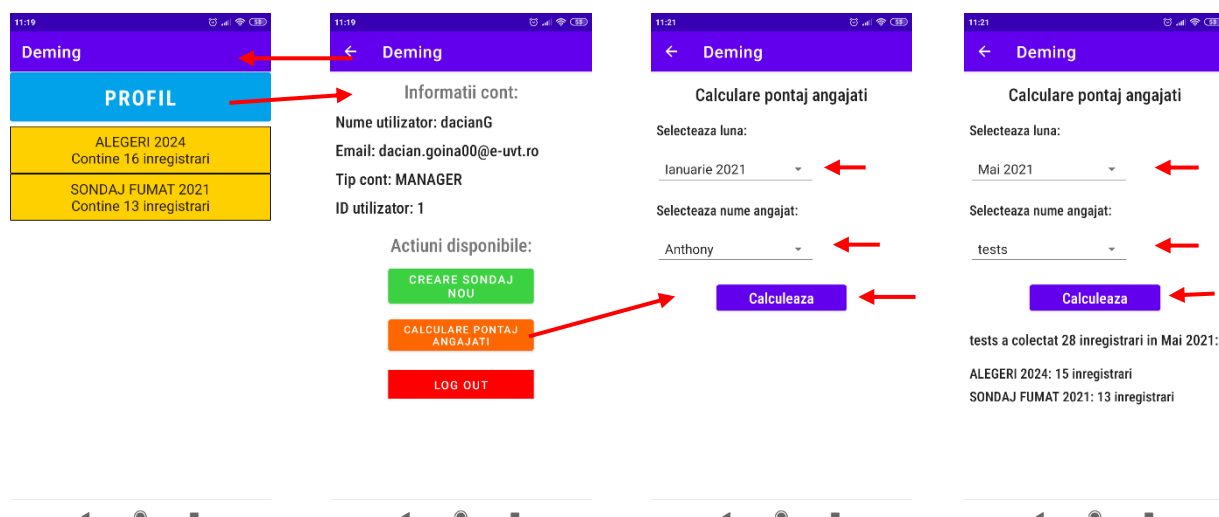
În cazul în care sondajul selectat conține zero înregistrări, se va afișa un mesaj corespunzător care va informa utilizatorul că nu poate efectua cele două acțiuni descrise mai sus. Mai jos sunt ilustrați pașii care trebuie urmați pentru a genera graficele care ilustrează rezultatele unui anumit sondaj.



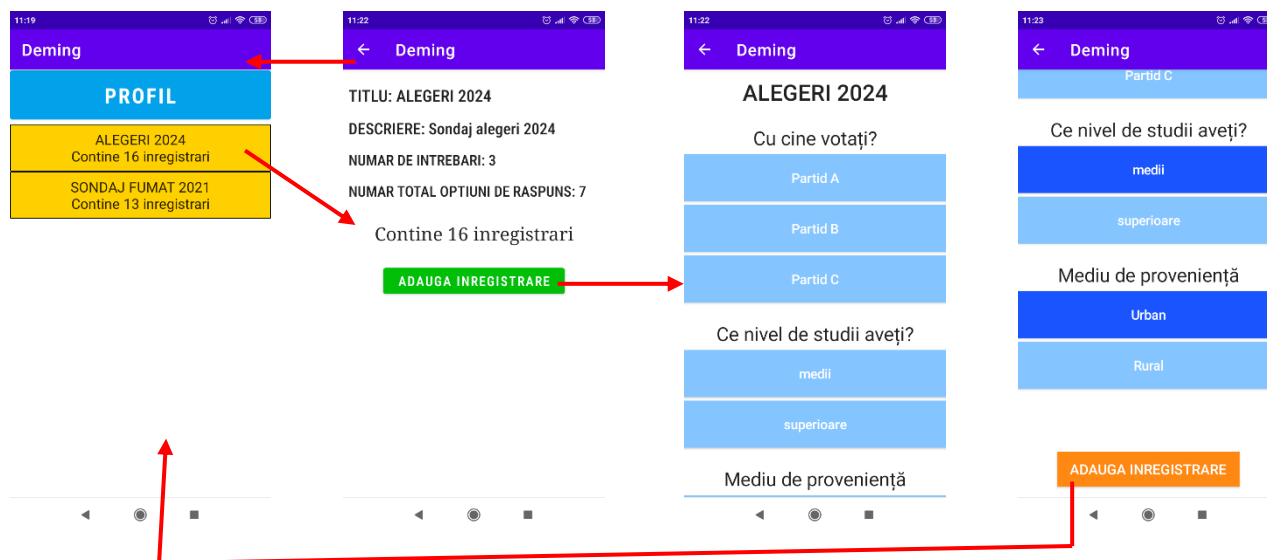
Calculare pontaj angajați: conturile de tip MANAGER pot verifica câte înregistrări a colectat fiecare angajat (utilizatorii care au cont de tip ANGAJAT). Pentru a realiza acest lucru se apasă butonul Profil, se deschide o nouă activitate iar în această activitate se apasă butonul Calculare pontaj angajați. În continuare se va deschide o nouă activitate, în această activitate trebuie specificate următoarele informații:

- luna pentru care se vrea verificarea pontajului
- numele angajatului

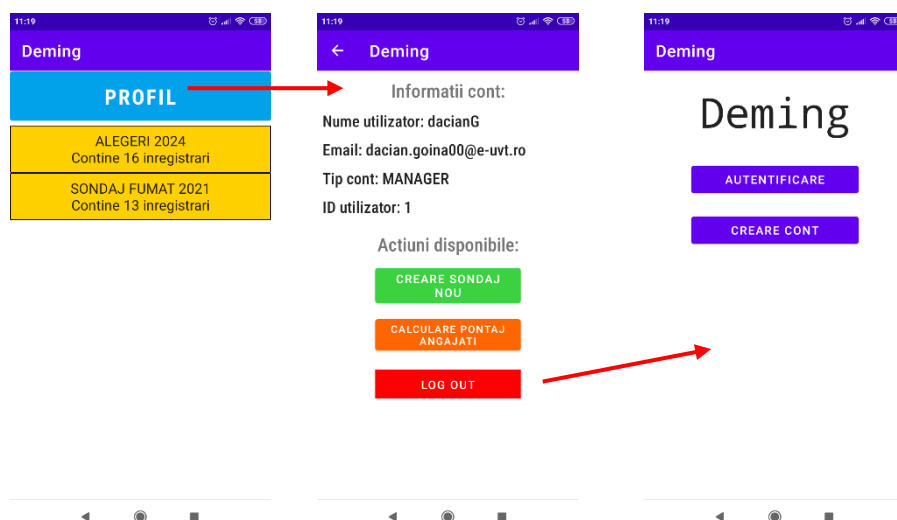
După specificarea acestor informații se apasă butonul Calculează iar pe ecran vor fi afișate informațiile corespunzătoare: numărul de înregistrări colectate de angajatul indicat, în luna indicată, precum și sondajele pentru care a colectat înregistrări. Mai jos sunt ilustrați pașii explicați anterior:



Adaugă înregistrare: utilizatorii care au cont de tip ANGAJAT sunt cei merg să discute cu oamenii și să colecteze date. Pentru a adauga o înregistrare într-un anumit sondaj se selectează sondajul dorit, se apasă butonul Adaugă înregistrare – acest lucru va deschide o fereastră care conține întrebările și opțiunile de răspuns ale sondajului selectat. În continua trebuie întrose răspunsurile primite la persoana cu care se discută, pentru a introduce un răspuns la o anumită întrebare se nagivează vertical (scroll) la fiecare întrebare și se selectează răspunsurile primite. Pentru a selectat un răspuns se face click pe el, răspunsul selectat va primi o culoare mult mai albastră decât răspunsurile neselectate. Dacă am selectat deja un răspuns și dorim să schimbăm răspunsul atunci selectăm alt răspuns, noul răspuns selectat va fi colorat în albastru închis, restul răspunsurilor din întrebarea respectivă vor fi colorate în albastru deschis. Dacă s-au selectat răspunsuri pentru toate întrebările atunci se apasă butonul Adauga inregistrare, utilizatorul va fi informat că înregistrarea a fost adăugată și va fi redirecționat la activitate principală a utilizatorului. Mai jos sunt ilustrați pașii pentru a adăuga o nouă înregistrare într-un anumit sondaj.



Deconectare: pentru deconectarea din aplicație (atât pentru conturile de tip ANGAJAT cât și pentru conturile de tip MANAGER) se apasă butonul Profil, se va deschide o nouă activitate iar acolo trebuie trebuie apăsat butonul Log out. Mai jos sunt ilustrați pașii pentru deconectarea din aplicație (în imagini s-a folosit un cont tip MANAGER, pentru un cont ANGAJAT, procedura este identică).



Documentație tehnică

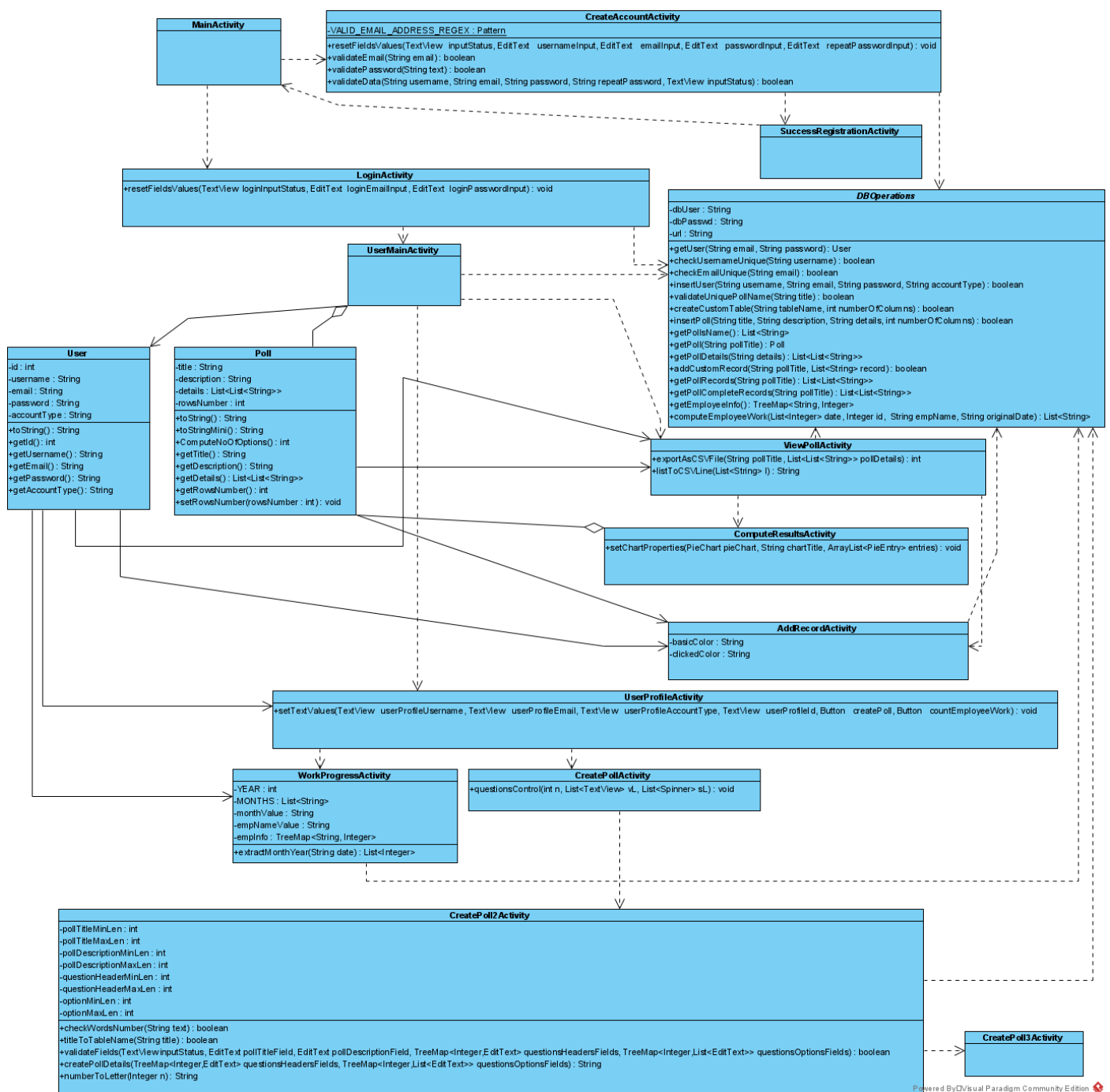
În continuare este explicată structura aplicației, implementările, funcționalitățile, explicații la nivel tehnic – pentru a fi înțelese de programatori.

Informații de bază

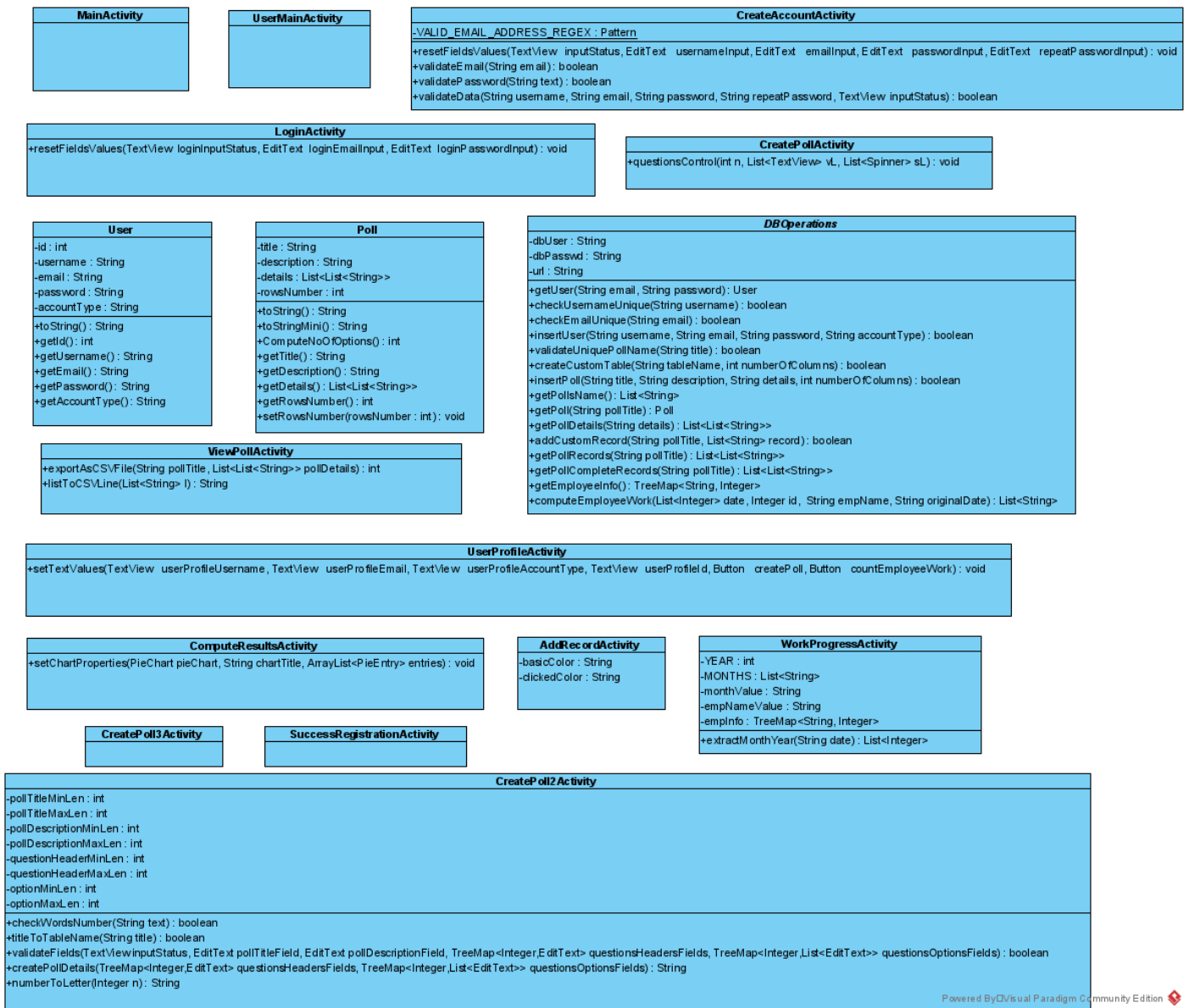
Aplicația este scrisă în limbajul Java, a fost creată folosind Android Studio. Pe lângă limbajul Java mai este folosit și limbajul XML. Aplicația poate fi rulată pe telefoanele (smartphones) cu sistem de operare Android, minim versiunea 8. Pentru a putea fi folosită efectiv, aplicația comunică cu o bază de date globală (adică nu este locală pe telefon). Ca sistem de gestiune a bazei de date se folosește H2, limbajul folosind fiind SQL. Comunicarea cu baza de date se realizează folosind conexiunea la internet.

Codul aplicației

La nivel de cod aplicația este formată din 16 clase, din care 13 clase sunt folosite pentru **activități**.



Varianta simplificată – conține doar clasele, fără relațiile dintre ele



Notă: toate clasele care conțin Activity în nume moștenesc (extends) clasa AppCompatActivity, acest lucru nu a mai fost specificat în diagramă deoarece o făcea să fie mai greu de vizualizat și citit. De asemenea, toate clasele cu Activity în nume conțin metoda onCreate cu semnătura:

protected void onCreate(Bundle savedInstanceState)

Explicarea claselor și a metodelor

Clasa MainActivity: este clasa folosită pentru activitatea care se afișează la deschiderea aplicației; în metoda onCreate din această clasă se creează butoanele care sunt folosite pentru trecerea la activitățile LoginActivity și CreateAccountActivity

Clasa CreateAccountActivity: clasa folosită pentru activitate care are ca scop pagina pentru a crea un cont. Conține diverse elemente GUI: câmpuri (fields) pentru a introduce adresa de email, numele de utilizator, parola, repetarea parolei, conține buton pentru a realiza crearea contului. Această clasă conține metodele:

- resetFieldsValues(Textview inputStatus, EditText usernameInput, EditText emailInput, EditText passwordInput, EditText repeatPasswordInput) care se folosește pentru a goli textul din câmpurile menționate anterior
- validateEmail(String email) - validează corectitudinea adresei de email introdus în câmpul text, adresa de email trebuie să respecte șablonul cunoscut (să conțină @, etc), acest șablon este indicat de variabila Pattern VALID_EMAIL_ADDRESS_REGEX (o expresie regex), membră a clasei
- validatePassword(String text) – verifică ca parola introdusă să conțină atât litere mari cât și litere mici
- validateData(String username, String email, String password, String repeatPassword, Textview inputStatus) – validează toate datele introduse în câmpurile text:
 - numele de utilizator trebuie să conțină cel puțin 3 caractere, să nu fie folosit deja de alt utilizator
 - parola trebuie să conțină cel puțin 3 caractere, să conțină atât litere mari cât și litere mici, textul pentru repetarea parolei să fie egal cu cel din câmpul pentru parolă
 - adresa de email trebuie să fie validă (validatePassword), trebuie să conțină cel puțin 5 caractere și să nu fie folosită deja de alt utilizator

Clasa SuccessfulRegistrationActivity: clasa folosită pentru activitate care indică faptul că contul a fost creat cu succes iar acum se poate folosi pentru autentificare; conține un buton continuă care trimite la MainActivity

Clasa LoginActivity: se folosește pentru activitatea LoginActivity, conține câmpuri în care trebuie introduse adresa de email și parola necesare logării; după completarea câmpurilor se apasă butonul Login, iar în cazul în care datele introduse sunt valide (contul există) se face trecerea la activitatea UserMainActivity

Clasa UserMainActivity: se folosește pentru activitate care reprezintă pagina principală pe care o vede un utilizator logat în aplicație; această pagină (activitate) conține un buton Profil care deschide activitatea UserProfileActivity, pe lângă acest buton în această activitate sunt afișate sondajele existente, mai exact sunt afișate pentru fiecare sondaj următoarele informații:

- titlul sondajului
- numărul de înregistrări pe care le are sondajul

Făcând click pe un sondaj se deschide activitatea ViewPollActivity.

Clasa ViewPollActivity: clasa folosită pentru activitatea care afișează informații despre sondajul selectat, sunt afișate următoarele informații:

- titlul sondajului
- descrierea sondajului
- numărul de întrebări din sondaj
- numărul total de opțiuni de răspuns din sondaj
- numărul de înregistrări (records) pe care le are sondajul

În funcție de tipul utilizatorului (ANGAJAT sau MANAGER), această activitate conține anumite butoane.

Dacă utilizatorul are cont de tip ANGAJAT atunci:

se afișează butonul Adauga înregistrare, apăsarea acestui buton deschide activitatea AddRecordActivity

Clasa AddRecordActivity: folosită pentru activitatea care permite adăugarea unei noi înregistrări în sondajul selectat, conține doi membri de tip String: basicColor și clickedColor care reprezintă culorile pe care le au butoanele cu variantele de răspuns când nu sunt / sunt apăsat. În această activitate sunt afișate toate întrebările din sondajul selectat precum și opțiunile de răspuns pentru fiecare întrebare; dacă este selectat un răspuns pentru fiecare întrebare atunci se apasă butonul Adaugă înregistrare pentru a trimite înregistrarea

Dacă utilizatorul are cont de tip MANAGER atunci:

se afișează butoanele Generare grafic și Exporțeaza ca CSV

butonul **Exporțează ca CSV**: dacă sondajul conține cel puțin o înregistrare atunci înregistrările colectate se pot exporta sub forma unui fișier CSV – se folosește metoda **exportAsCSVFile(String pollTitle, List<List<String>> pollDetails)** care primește titlul sondajului și alte informații despre sondaj, această metodă creează pe telefon un fișier cu extensie CSV în care scrie datele colectate pentru sondaj după ce preia aceste înregistrări din baza de date; ca metodă auxiliară se folosește de **listToCSVLine(List<String> l)** – metodă care primește o listă de String și returnează valorile din listă sub forma unei linii dintr-un fișier CSV, ex: pentru l = [”azi”, ”este”, ”luni”] se returnează:””azi”,”este”,”luni”\n”

butonul **Generare grafic**: dacă sondajul conține cel puțin o înregistrare atunci se pot genera grafice pentru a vizualiza rezultatele sondajului, apăsarea acestui buton deschide ComputeResultsActivity

Clasa ComputeResultsActivity : se folosește pentru activitatea în care sunt afișate graficele create pe baza înregistrărilor colectate pentru sondaj; clasa conține un array int COLORS care conține valori pentru culori în format RGB, culori folosite pentru culorile din graficele generate. Graficele sunt create cu ajutorul librăriei **MPAndroidChart [8]**, acestea sunt de tip PieChart, se creează un grafic pentru fiecare întrebare din sondaj, se folosește metoda **setChartProperties(PieChart pieChart, String chartTitle, ArrayList<PieEntry> entries)** pentru a configura proprietățile graficelor (de ex. intrările – entries din grafic, proprietăți referitoare la legendă – dimensiunea textului, aliniere, afișare în stil orizontal sau vertical etc)

Clasa User: folosită pentru a ține / gestiona (ca o structură de date) informații despre utilizatorul autentificat în aplicație. Membrii acestei clase sunt:

- int id – reprezintă ID - ul utilizatorului care este unic raportat la restul utilizatorilor
- String username – numele utilizatorului, unic raport la restul utilizatorilor
- String email – adresa de email a utilizatorului, unică raportat la restul utilizatorilor
- String password – parola utilizatorului
- String accountType – tipul contului utilizatorului (ANGAJAT sau MANAGER)

Clasa **User** conține constructor cu toate cele 5 atribute explicate mai sus, conține metodă `toString()` pentru afișare, conține metode de `get` și `set` pentru atributele explicate anterior.

Clasa Poll: reprezintă un sondaj, se folosește pentru a gestiona mai ușor informații despre sondaje după ce sunt preluate din baza de date. Această clasă conține următorii membrii:

- `String title` – reprezintă titlul sondajului, este unic raportat la restul sondajelor
- `String description` – reprezintă descriere sondajului
- `List<List<String>> details` – reprezintă datele despre sondaj în sine: enunțul întrebărilor, opțiunile de răspuns
- `int rowsNumber` – reprezintă numărul de înregistrări ale sondajului

Clasa conține 2 constructori:

- unul care conține doar membrii **title** și **rowsNumber**, se folosește în `UserMainActivity` când este nevoie să se afișeze doar titlul și numărul de înregistrări din sondaj
- unul care conține toți cei 4 membrii prezentați mai sus

Clasa conține diverse metode:

- `toString()` pentru a afișa un obiect format cu toți cei 4 membri.
- `toStringMini()` pentru a afișa un obiect format cu constructorul care conține doar 2 membri
- metode de `get` și `set` pentru membrii clasei
- metoda **ComputeNoOfOptions()** care calculează și returnează numărul total de opțiuni de răspuns din sondaj, ex. dacă în sondaj sunt 2 întrebări, una cu 2 opțiuni de răspuns iar alta cu 3 opțiuni atunci rezultat returnat de metodă va fi 5

Din activitatea **UserMainActivity**, prin apăsarea butonul **Profil** se deschide activitatea **UserProfileActivity**.

Clasa UserProfileActivity: este folosită pentru activitate care reprezintă pagina pentru profilul utilizatorului, pe această pagină se afișează informații despre utilizator:

- `id` – id-ul utilizatorului
- `tipul contului`
- `numele de utilizator`
- `adresa de email`

În această activitate se găsesc anumite butoane în funcție de tipul utilizatorului:

Dacă utilizatorul este **ANGAJAT** atunci în această activitate se găsește doar butonul `Log out` care realizează deconectarea din aplicație, redirectionarea la `MainActivity`

Dacă utilizatorul este **MANAGER** atunci această activitate mai conține încă două butoane:

- `Creare sondaj nou`
- `Calculare pontaj angajați`

Apăsarea butonului `Calculare pontaj angajați` deschide activitatea `WorkProgressActivity`

Clasa WorkProgressActivity: folosită pentru activitatea în care se poate calcula pontajul angajaților – practic, pentru un anumit angajat identificat prin nume se poate verifica câte înregistrări a adunat într-o anumită lună, precum și sondajele pentru care a colectat.

Această clasă conține următorii membrii:

- `int YEAR` – pentru a stoca valoarea anului curent
- `List<String> MONTHS` – listă de `String` care conține numele lunilor
- `String monthValue` – pentru valoarea care va fi selectată pentru lună
- `String empNameInfo` – pentru valoarea numelui angajatului care va fi selectat
- `TreeMap<String,Integer> empInfo` – pentru a stoca perechi de tip nume – id, care vor reprezintă numele și id-ul angajaților

Selectarea angajatului pentru care se calculează numărul de înregistrări colectate se face prin numele de utilizator al acestuia, dar căutarea efectivă în tabelele din baza de date se face folosind ID -ul angajatului. Selectarea numelui angajatului precum și a lunii se realizează folosind elemente grafice `Spinner`.

Această clasă conține metoda **`extractMonthYear(String date)`** care primește o dată (lună, an) sub format text și extrage din ea luna și anul, returnează rezultatul sub forma unei liste de `Integer`, de ex.: pentru "Ianuarie 2021" va returna `[1, 2021]`

Clasa `CreatePollActivity`: se folosește pentru activitatea care permite crearea unui nou sondaj; în această activitate se regăsesc elemente `Spinner`, folosind aceste obiecte grafice, utilizatorul introduce numărul de întrebări pentru sondajul care se va crea, precum și numărul de opțiuni de răspuns pentru fiecare întrebare, după introducerea acestor informații se apasă butonul continuă care deschide activitatea `CreatePoll2Activity`

Clasa `CreatePoll2Activity`: folosită pentru a activitatea în care se introduc datele pentru sondajului care va fi creat. Activitatea conține câmpuri text pentru:

- titlul sondajului
- descrierea sondajului
- enunțul fiecărei întrebări din sondaj
- opțiunile de răspuns pentru fiecare întrebare din sondaj

După completarea acestor date se apasă butonul Creare sondaj. Această clasă conține următorii membrii:

- `int pollTitleMinLen, pollTitleMaxLen` : valori folosite pentru a indica numărul de caractere pe care îl poate conține titlul sondajului pentru a putea fi creat
- `int pollDescriptionMinLen, pollDescriptionMaxLen` : valori folosite pentru a indica numărul de caractere pe care îl poate conține descrierea sondajului
- `int questionHeaderMinLen, questionHeaderMaxLen` : valori folosite pentru a indica numărul de caractere care îl poate conține fiecare întrebare din sondaj
- `int optionMinLen, int optionMaxLen` : valori folosite pentru a indica numărul de caractere care îl poate conține fiecare opțiune de răspuns din sondaj

La apăsarea butonului Creare sondaj, datele introduse în câmpuri sunt validate folosind mai multe metode:

- `checkWordsNumber(String text)` – primește un `String` și verifică ca acesta să conțină cel puțin două cuvinte, se folosește pentru a valida titlul sondajului
- `titleToTableName(String title)` – primește un `String` – titlul sondajului și returnează un alt `String` care reprezintă numele care va fi folosit pentru tabelul din baza de date în care vor fi stocate înregistrările pentru noul sondaj, leagă cuvintele din titlul sondajului folosind " _ ", de ex.: pentru titlul SONDAJ ALEGERI 2024, numele tabelului va fi SONDAJ_ALEGERI_2024
- `numberToLetter(Integer n)` – primește un număr între 1 și 7 și returnează identificatorul corespunzător – litera din alfabet de la poziția `n`, ex. 1 => A, 4 => D, se va folosi pentru numele

coloanelor din tabelul în care se vor stoca înregistrările pentru noul sondaj (mai multe explicații în paginile următoare – partea referitoare la baza de date)

- `validateFields(TextView inputStatus, EditText pollTitleField, EditText pollDescriptionField, TreeMap<Integer, EditText> questionsHeadersFields, TreeMap<Integer, List<EditText>> questionsOptionsFields)` - validează toate textele introduse în câmpuri text în această activitate: titlul sondajului, descrierea sondajului, întrebările din sondaj, opțiunile de răspuns. Condițiile de validare sunt:
 - lungimea corespunzătoare a textului (conform valorilor membrilor clasei explicați mai sus)
 - numărul de cuvinte din titlul sondajului
 - titlul sondajului să nu conțină caracterul ” _ ”
 - fiecare text din câmp să nu conțină caracterul ” | ” (bară verticală)

Clasa `CreatePoll2Activity` mai conține metoda `createPollDetails(TreeMap<Integer, EditText> questionsHeadersFields, TreeMap<Integer, List<EditText>> questionsOptionsFields)`, aceasta primește datele introduse în câmpurile pentru întrebări și opțiunile de răspuns, returnează un String reprezintă gruparea / ”codificarea” acestor informații pentru a putea fi puse în baza de date.

Context: pentru fiecare sondaj este necesar să ținem înregistrările colectate în baza de date, fiecare sondaj va avea tabelul lui în care vor fi puse înregistrări. De ex. fie sondajul ilustrat prin următorul tabel:

Identificator întrebare	Întrebare	Variante de răspuns
A	Cu ce partid votați?	<ul style="list-style-type: none">▪ Partid A▪ Partid B▪ Partid C
B	Ce nivel de studii aveți?	<ul style="list-style-type: none">▪ Medii▪ Superioare
C	Mediu de proveniență	<ul style="list-style-type: none">▪ Rural▪ Urban

Pentru acest sondaj putem crea un tabel SQL în care să ținem înregistrările colectate de la oameni (răspunsurile primite), de ex.

ID	A	B	C	DATA	ID_ANGAJAT
1	Partid A	medii	Urban	2021-05-04	2
2	Partid B	medii	Rural	2021-05-04	2
3	Partid C	medii	Urban	2021-05-04	2
4	Partid C	superioare	Urban	2021-05-04	2
5	Partid A	medii	Urban	2021-05-04	2
6	Partid B	superioare	Rural	2021-05-04	2

Astfel putem stoca înregistrările pentru un sondaj, dar trebuie să stocăm și **informațiile efective despre sondaj (titlul, descriere, întrebări, opțiuni de răspuns)**. Pentru acest lucru se folosește un tabel numit **SONDAJE**. Se pot folosi două coloane pentru **titlu** și **descriere**, dar trebuie stocate și întrebările precum și variantele de răspuns. Pentru a stoca aceste ultime informații (întrebări, opțiuni de răspuns) se folosește încă o coloană numită **detalii** în care se stochează o valoare text care este ”codificarea” informațiilor menționate anterior.

Metoda folosită pentru ”codificare” / lipirea / gruparea întrebărilor, a opțiunilor de răspuns într-un singur text (pentru a putea fi pus doar pe o coloană) se face în următorul mod:

- identificatorul unei întrebări (A, B, C ...), împreună cu enunțul întrebării și opțiunile de răspuns se concatenează într-un String folosind ” | ” ca separator, acest String rezultat se va numi mini-grup (termen folosit doar pentru a explica mai departe)
- toate mini-grupurile se concatenează într-un String final, se folosește ” || ” (două linii verticale) ca separator

Notă: identificatorul întrebării (A, B, C ..) este pus automat

De ex, pentru sondajul ilustrat prin tabelul de mai jos:

Identificator întrebare	Întrebare	Variante de răspuns
A	Cu ce partid votați?	<ul style="list-style-type: none">▪ Partid A▪ Partid B▪ Partid C
B	Ce nivel de studii aveți?	<ul style="list-style-type: none">▪ Medii▪ Superioare
C	Mediu de proveniență	<ul style="list-style-type: none">▪ Rural▪ Urban

Pentru prima întrebare, concatenarea textului într-un String este următoarea:

A | Cu ce partid votați | Partid A | Partid B | Partid C

Astfel, concatenarea / gruparea informațiilor din tabelul de mai sus este următoarea:

A |Cu ce partid votați | Partid A | Partid B | Partid C ||

B | Ce nivel de studii aveți? | Medii | Superioare ||

C |Mediu de provenienta | Rural |Urban

Astfel, mai sus există doar un String, aceasta poate fi stocat într-o singură celulă (adică doar pe o coloană) în baza de date. Când se aduce acest String în aplicație, degрупarea / operație inversă concatenării se face folosind metoda **split** care este deja implementată în limbajul Java

Clasa DBOperations: conține metode care reprezintă operații care asigură comunicarea dintre aplicație și baza de date (înserarea de date, interogarea bazei de date etc)

Metode:

- `getUser(String email, String password)` – interoghează baza de date, extrage datele despre un anumit utilizator identificat prin email și password, se folosește pentru autentificare
- `checkUsernameUnique(String username)` - primește un String care reprezintă un nume de utilizator și verifică dacă există deja (în baza de date) un utilizator cu acest nume, se folosește la crearea unui cont, pentru a ne asigura că fiecare nume de utilizator este unic
- `checkEmailUnique(String email)` – similar cu `checkUsernameUnique`, dar verifică unicitatea pentru adresa de email
- `insertUser(String username, String email, String password, String accountType)` – înserează un nou utilizator în baza de date, în tabelul corespunzător.

- `validateUniquePollName(String title)` – verifică ca titlul unui sondaj să fie unic – caută în baza de date să nu existe deja un sondaj cu titlul dat prin `title`
- `createCustomTable(String tableName, int numberOfColumns)` – creează un nou tabel care se va folosi pentru a stoca înregistrările unui sondaj, folosind `String`-ul `tableName`, concatenează cuvintele folosind separatorul ”_” pentru numele noului tabel, ex. SONDAJ FUMAT 2021 => SONDAJ_FUMAT_2021; acest tabel nou creat poate avea un număr variabil de coloane în funcție de numărul de întrebări pe care îl va avea noul sondaj (între 2 și 7 întrebări), structura acestui tabel nou creat este:
ID | A | B | G | DATA | ID_ANGAJAT adică prima coloană va conține ID – ul înregistrării, următoarele coloane vor conține efectiv răspunsurile primite la întrebările din sondaj, penultima coloană va stoca data la care a fost trimisă înregistrarea, ultima coloană va stoca ID -ul angajatului care a trimis înregistrarea
- `insertPoll(String title, String description, String details, int numberOfColumns)` – inserează un nou sondaj în baza de date (în tabelul corespunzător)
- `List<Poll> getPollsName()` – returnează o listă care conține obiecte de tip `Poll`, pentru aceste obiecte se folosește constructorul cu 2 parametri – nume sondaj și număr de înregistrări
- `Poll getPoll(String pollTitle)` – returnează un obiect de tip `Poll`, obiect format cu valorile extrase din baza de date, sondajul este identificat în tabelul corespunzător prin `pollTitle`
- `addCustomRecord(String pollTitle, List<String> record)` – adaugă o nouă înregistrare în baza de date, numele sondajului este identificat prin `pollTitle`, conținutul noii înregistrări (răspunsuri, data, id angajat) se găsesc în lista **record**
- `List<List<String>> getPollRecords(String pollTitle)` – returnează o listă de liste care conține toate înregistrările pentru un anumit sondaj identificat prin `pollTitle`, fiecare listă din lista mare conține datele de pe o coloană din tabelul folosit pentru a stoca înregistrările, de ex. în prima listă din lista mare se vor regăsi valorile de pe prima coloană din tabelul cu înregistrări
- `List<List<String>> getPollCompleteRecords(String pollTitle)` – asemănătoare cu metoda de mai sus, dar acum se aduc în aplicație toate datele despre înregistrările unui anumit sondaj, la metoda anterioară se aduceau doar răspunsurile efective la întrebări (deci nu se aducea ID – ul înregistrării, data trimerii, ID – ul angajatului)
- `TreeMap<String,Integer> getEmployeeInfo()` – extrage din baza de date datele despre angajați (nume de utilizator, ID), le aduce în aplicație, creează un `TreeMap` în care valorile cheilor vor fi numele angajaților iar valorile datelor vor fi ID – urile angajaților, se folosește `TreeMap` deoarece vrem ca ordinea cheilor (numele angajaților) să apară în ordine alfabetică
- `computeEmployeeWork(List<Integer> date, Integer id, String empName, String originalDate)` – se folosește pentru a calcula câte înregistrări a colectat fiecare angajat (precum și sondaje le pentru care a colectat) într-o anumită lună (adică pontajul unui angajat)

Baza de date

Pentru baza de date se folosește sistemul de gestiune H2, limbajul de interogare fiind SQL. Baza de date este un globală (adică nu este una locală – nu este ținută doar pe un telefonul individual).

În baza de date sunt 2 tabele principale: UTILIZATORI și SONDAJE, în acestea sunt ținute date despre utilizatori, precum și date generale despre sondaje (nume, descriere, detalii). Structura acestor două tabele:

(PK înseamnă că coloana respectivă este cheie primară, UNIQUE înseamnă că coloana respectivă are această proprietate – fiecare celulă din acest tip de coloană are valori unice raportat la coloană)

UTILIZATORI	SONDAJE
PK int ID UNIQUE text USERNAME UNIQUE text EMAIL text PASSWORD text ACCOUNTTYPE	PK text NUME text DESCRIERE text DETALII

Pe lângă aceste tabele principale, baza de date de mai conține și alte tabele, mai exact fiecare sondaj are un tabel propriu în care se stochează înregistrările colectate pentru acel sondaj. Aceste tabele suplimentare se creează la crearea fiecărui sondaj nou. Numărul de coloane din aceste tabele variază în funcție de numărul de întrebări din sondajul respectiv. Fiecare din aceste tabele suplimentare au în comun următoarele coloane:

ID – cheie primară, coloană de tip int, reprezintă ID – ul unei înregistrări

DATA – coloană de tip Date, reprezintă data la care a fost trimisă înregistrarea

ID_ANGAJAT – coloană de tip int, reprezintă ID – ul angajatului care a colectat înregistrare

Un astfel de tabel are următoarea structură

TITLU_SONDAJ
PK int ID text A text B text C text D text E text F text G Date DATA int ID_ANGAJAT

Pentru coloanele A, B, C, ... F: fiecare tabel conține A, B, restul coloanelor de la D la G există / nu există în tabel în funcție de numărul de întrebări din sondaj. Ex: dacă sondajul conține 3 întrebări atunci vor exista coloanele A, B, C, dacă conține 4 atunci vor exista coloanele A, B, C, D. Notă: fiecare sondaj conține între 2 și 7 întrebări.

Numele tabelului corespunzător unui sondaj se formează pornind de la numele sondajului, cuvintele din numele sondajului se concatenează folosind ” _ ” ca separator, ex:

nume sondaj: ACESTA ESTE UN SONDAJ, nume tabel corespunzător: ACESTA_ESTE_UN_SONDAJ

De asemenea numele sondajelor sunt puse automat cu litere mari, nu pot conține caracterul ” _ ”

Câteva exemple cu tabele din baza de date:
Tabelul UTILIZATORI:

SELECT * FROM UTILIZATORI;

ID	USERNAME	EMAIL	PASSWORD	ACCOUNTTYPE
1	dacianG	dacian.goina00@e-uvt.ro	Abcde	MANAGER
2	tests	b	b	ANGAJAT
65	Marius	marius.marius@ma.ro	Marius	ANGAJAT
97	Anthony	anthony.aaa@gmail.com	Abcde	ANGAJAT

(6 rows, 31 ms)

Tabelul SONDAJE:

SELECT * FROM SONDAJE;

NUME	DESCRIERE	DETALII
ALEGERI 2024	Sondaj alegeri 2024	A Cu cine votezi? Partid A Partid B Partid C B Ce nivel de studii aveți? medii superioare C Mediu de proveniență Urban Rural
SONDAJ FUMAT 2021	Un sondaj despre fumat	A Ce marcă de țigări fumați? Marca A Marca B Marca C Marca D Marca E Marca F B Câte țigări fumați pe zi? maxim 2 între 3 și 5 între 5 și 7 peste 7 C Cât cheltuiți lunar pe țigări? maxim 100 lei între 100 și 250 lei între 250 și 500 lei între 500 și 1000 lei peste 1000 lei D Vă gândiți să vă lăsați? Da Nu

Tabele a stoca înregistrările pentru sondaje:

SELECT * FROM ALEGERI_2024;

ID	A	B	C	DATA	ID_ANGAJAT
1	Partid A	medii	Urban	2021-05-04	2
2	Partid B	medii	Rural	2021-05-04	2
3	Partid C	medii	Urban	2021-05-04	2
4	Partid C	superioare	Urban	2021-05-04	2
5	Partid A	medii	Urban	2021-05-04	2
6	Partid B	superioare	Rural	2021-05-04	2
7	Partid A	superioare	Rural	2021-05-04	2
8	Partid C	superioare	Urban	2021-05-04	2
9	Partid A	medii	Rural	2021-05-04	2
10	Partid B	medii	Urban	2021-05-04	2

SELECT * FROM SONDAJ_FUMAT_2021;

ID	A	B	C	D	DATA	ID_ANGAJAT
1	Marca A	peste 7	între 100 și 250 lei	Da	2021-05-08	2
2	Marca A	peste 7	peste 1000 lei	Nu	2021-05-08	2
3	Marca C	între 3 și 5	între 500 și 1000 lei	Da	2021-05-08	2
4	Marca E	maxim 2	între 250 și 500 lei	Nu	2021-05-08	2
5	Marca A	maxim 2	maxim 100 lei	Da	2021-05-08	2
6	Marca B	între 3 și 5	între 100 și 250 lei	Da	2021-05-08	2
7	Marca C	între 5 și 7	între 250 și 500 lei	Nu	2021-05-08	2
8	Marca D	peste 7	între 500 și 1000 lei	Da	2021-05-08	2

Concluzii și direcții viitoare

Deming este o aplicație folosită pentru a crea și gestiona sondajele de opinie. Aplicația a fost dezvoltată folosind limbajul de programare Java și mediul de dezvoltare Android Studio. Ea poate fi folosită pe telefoanele (smartphones) cu sistem de operare Android, minim versiunea 8 (Android Oreo).

Aplicația este destinată oamenilor / în general firmelor care realizează sondaje de opinie, aplicația facilitând colectarea datelor dar și prelucrarea lor. Angajații care merg să colecteze date în urma discuțiilor cu oamenii nu vor mai trebuie să noteze răspunsurile pe hârtie, ci direct în aplicație. Mai mult, pentru introducerea răspunsurilor (input) primite de la oameni, nu este nevoie de scrierea de la tastatura virtuală a telefonului, ci direct apăsarea pe butoanele (afișate pe ecran) corespunzătoare răspunsurilor, acest lucru oferind mai multă flexibilitate și economie de timp.

Printre funcțiile principale aplicației se numără: crearea sondajelor de opinie, colectarea datelor pentru acestea, exportul datelor, generarea de grafice pe baza rezultatelor, precum și calcularea activității angajaților (persoanele care merg să colecteze înregistrări pentru sondaje prin discuții cu cetățenii). Toate datele colectate sunt trimise într-o bază de date globală, de acolo ele se pot prelucra în diferite feluri – generare sondaje pe baza lor pentru a vizualiza rezultatele sondajului, calcularea pontajului angajaților pentru a vedea câte înregistrări a colectat fiecare, exportul datele sub forma unui fișier CSV – în cazul în care se dorește folosirea datelor colectate pentru alte scopuri.

Aplicații similare

Așa cum s-a menționat la începutul acestui document, există aplicații similare cu cea prezentată aici: KoboCollect, TeamScope. O comparație generală între aplicațiile menționate anterior și Deming se poate citi în tabelul de mai jos:

	Deming	KoboCollect	TeamScope
Permite generarea de grafice	DA	DA	DA
Tip de grafic folosit pentru reprezentare date	PieChart ("plăcintă")	BarChart (bări verticale)	LineChart (linii)
Permite export date în fișier CSV	DA	NU	DA
Calculare activitate pentru colectori	DA	NU	DA
Costă (necesită plată pentru utilizare)	NU	NU	DA
Tip bază de date folosită	SQL	NOSQL	NOSQL

Direcții viitoare pentru dezvoltarea aplicației

În general nu se poate spune despre o aplicație că este finalizată, întotdeauna se vor putea adăuga sau îmbunătăți lucruri în funcționalitatea ei. Aplicația Deming poate fi dezvoltată în continuare, pe viitor aş vrea să mai adaug noi facilități:

- posibilitatea de a modifica culorile graficelor – momentan graficele au un set static de culori : prima culoare care apare în grafic (în legendă) este roșie, apoi albastră, galbenă etc; pe viitor aş dori să adaug o opțiune care să permită utilizatorului să aleagă ordinea culorilor
- adăugare unei funcții pentru a determina cel mai activ angajat – cel care a colectat cele mai multe înregistrări într-o anumită perioadă de timp

Referințe

1. **Tatiana Pascovschi.** *SONDAJ. Alegeri parlamentare: PNL conduce în preferințele electorale. PSD suflă puternic în spatele liberalilor. Cum stau USR PLUS, PMP, PRO Romania și UDMR.* Mediafax.ro
2. **Petre Dobrescu.** *Sondaj: Ce părere au românii despre fumatul în locuri publice.* Libertatea.ro
3. **Dan Popa.** *Perspectiva consumatorilor asupra pieței imobiliare: 62% dintre români nu și-ar achiziționa o locuință în următoarele 6 luni, în timp ce aproape 10% dintre aceștia își caută o a doua locuință în afara orașului (sondaj).* Hotnews.ro
4. **Maria Tufan.** *Ce ascunde un sondaj de opinie? Ghidul alegătorului circumscript.* Pressone.ro
5. Mod de lucru IRES (Institutul Român pentru Evaluare și Strategie) : *ALĂTURĂ-TE REȚELEI NAȚIONALE DE OPERATORI DE INTERVIU IRES!*. Ires.ro
6. KoboCollect pe Google Play :
<https://play.google.com/store/apps/details?id=org.koboc.collect.android&hl=ro&gl=US>
7. Teamscope pe Google Play :
<https://play.google.com/store/apps/details?id=com.teamscopeapp&hl=ro&gl=US>
8. MPAndroidChart – librărie dezvoltată de PhilJay : <https://github.com/PhilJay/MPAndroidChart>