

Colegiul Național “Moise Nicoară” Arad

LUCRARE DE ATESTAT
DISCIPLINA INFORMATICĂ

Candidat,

Bejan Andrei-Paul

Coordonator,

Prof. Cameniță Daniela

2021

Tema lucrării

Proiectul se intitulează "*Venit Zilnic*" și constă într-o aplicație de telefon menită să ofere o unealtă ușor de folosit de către muncitorii din fabrici, care sunt plătiți la bucată, pentru a-și nota eficient munca zilnică și a cunoaște în fiecare moment cât urmează să încaseze, pentru a nu exista erori în calculul de la sfârșitul lunii.



CUPRINS

Capitolul 1: Introducere (pagina 4)

1.1. Alegerea proiectului

1.2. Scopul aplicației

1.3. Necesități hardware/software

Capitolul 2: Baza teoretică (pagina 5)

2.1. Mediu de programare

2.2. Tratarea teoretică a subiectului

Capitolul 3: Implementarea practică (paginile 6-12)

3.1. Prezentarea generală a aplicației

3.2. Baza de date

3.3. Algoritmi utilizați

3.4. Probleme întâmpinate

Capitolul 4: Concluzii (pagina 13)

Capitolul 1: Introducere

1.1. Alegerea proiectului

Am ales acest proiect, deoarece voiam să îmi dezvolt cunoștințele de programare și să creez un program folositor în viața de zi cu zi. În acest sens, am încercat să rezolv o problemă cu care se confruntă mama mea, care lucrează într-o fabrică și este plătită pe unități produse. La sfârșitul fiecărei luni trebuia să-și calculeze numărul total de bucați realizate și venitul brut pentru a le compara cu informațiile de pe fluturașul primit de la locul de muncă.

1.2. Scopul aplicației

Scopul aplicației este să ferească muncitorii din fabrici, plătiți pe unități produse, de riscul de a li se oferi un salariu mai mic decât cel meritat cu adevărat. De asemenea, utilizatorii aplicației au acces la informații utile, precum statistici și grafice care vizează productivitatea acestuia, dar și sume totale precum venitul lunar brut sau numărul total de bucați din fiecare produs realizat. Aplicația este cel mai la îndemână loc în care putem salva informațiile despre munca depusă, deoarece reține local date din ultimele 12 luni și oferă acces instant la informații din fiecare zi a anului.

1.3. Necesități hardware/software

Aplicația este destinată utilizatorilor de telefoane inteligente cu sistem de operare Android 6.0 sau mai nou. Aplicația poate fi instalată direct de pe internet urmând acest [link](#). După instalare, aplicația nu are nevoie de internet, ci doar de acces la notificările și la data și ora telefonului pentru a putea trimite notificări și a seta ora exactă în cadrul aplicației. De asemenea, toate informațiile sunt salvate local, în memoria telefonului, pentru a avea o securitate sporită.

Capitolul 2: Baza teoretică

2.1. Mediul de programare

Pentru crearea aplicației am ales *Android Studio* ca mediu de programare, deoarece oferă acces la unelte foarte folosite pentru dezvoltatorii de aplicații pentru telefon precum un emulator pentru testarea aplicației și o documentație foarte ușor de urmărit, dar și forum-uri pe care există răspunsuri la diferite probleme întâmpinate de comunitate. Limbajul de programare pentru care am optat este *Java*, deoarece este unul dintre cele mai utilizate în crearea de aplicații și se aseamănă cu limbajul de baza *C++* pe care l-am studiat, iar pentru orice nelămurire există un răspuns pe internet de la specialiștii în acest limbaj.

2.2. Tratarea teoretică a subiectului

La început utilizatorul are posibilitatea de a adăuga diferite produse alături de costul lor. Odată îndeplinit acest lucru aplicația oferă posibilitatea înregistrării progresului sub forma unei liste de produse din cele înregistrate anterior, alături de numărul de bucăți corespunzător. Acest proces poate fi reluat de utilizator în fiecare zi, iar aplicația va memora toate informațiile acestuia. Pe baza informațiilor adunate se vor dezvolta statistici zilnice/săptămânale/lunare, dar și grafice în care se poate observa productivitatea acestuia. De asemenea, aplicația ar trebui să notifice utilizatorul zilnic/săptămânal pentru a-i aminti acestuia de necesitatea de a înregistra în aplicație progresul făcut.

Capitolul 3: Implementarea practică

3.1. Prezentarea generală a aplicației



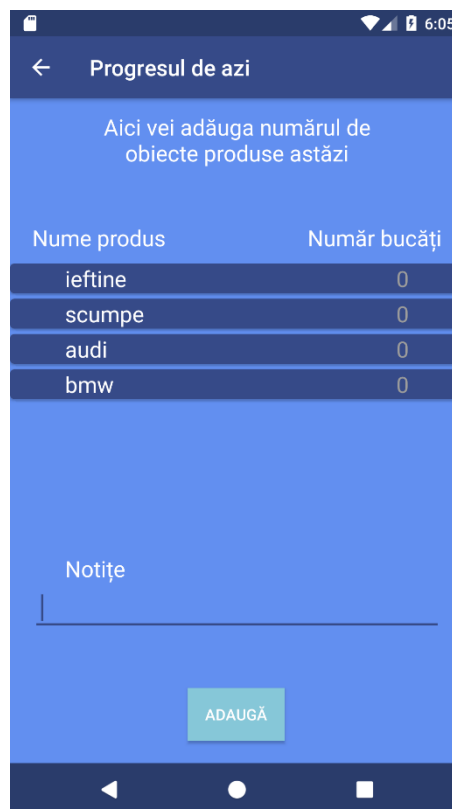
Pagina principală a aplicației

Pagina conține un citat motivațional care se schimbă de fiecare dată când revenim la această pagină, pentru a induce o productivitate sporită. Acest lucru este realizat prin alegerea aleatorie a unui citat dintr-o listă de 50 de citate.

În partea de sus se află un meniu care conține alte 5 tab-uri intitulate în ordine: *Progresul de azi*, *Total lunar*, *Statistici*, *Grafice*, *Calendar*, *Setări*. Acestea oferă acces ușor la toate facilitățile aplicației.

Primul tab – Progresul de azi

Această pagină oferă utilizatorului o interfață sugestivă pentru a adăuga cât mai ușor progresul zilei curente, în cazul unei zile lungi și stresante la locul de muncă. De asemenea, se oferă posibilitatea adăugării unei notițe care să descrie modul în care a decurs ziua curentă.



Al doilea tab – Total lunar

În această pagină apar toate produsele realizate de utilizator, însumate în funcție de luna aleasă.

În partea de jos apar numărul total de obiecte și venitul brut în funcție de costul înregistrat al fiecărui obiect în tab-ul *Setări*.



Tab-ul Setări

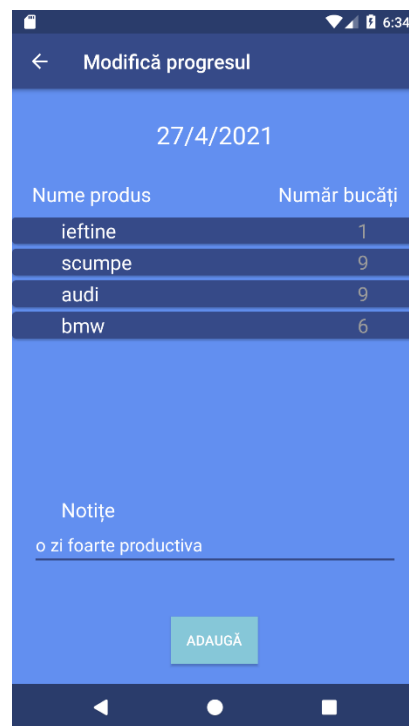
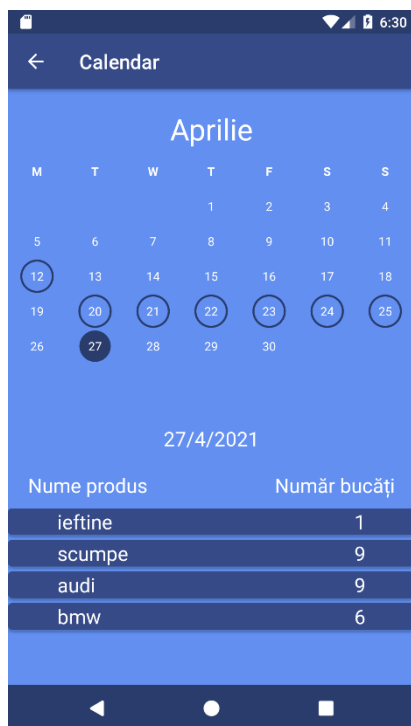
Pagina aceasta oferă posibilitatea de a alege tipul de notificări preferat de utilizator, dar oferă și o interfață sugestivă pentru a adăuga tipurile de produse și prețul lor în lei, acceptând și valori cu virgulă. De asemenea, există posibilitatea de a le elimina pe cele pe care nu le mai folosește sau schimbarea prețului fiecărui obiect în parte.

The screenshot shows the 'Setări' (Settings) screen with a blue background. The title bar has a back arrow and the text 'Setări'. The first section is 'Schimbă tipul notificărilor' with three radio buttons: 'Zilnic' (selected), 'Săptămânal', and 'Niciodată'. The second section is 'Înregistrează un nou obiect' with two input fields: 'Denumire' and 'Valoare (lei)'. Below these are three buttons: 'ADAUGĂ', 'SCHIMBĂ', and 'ȘTERGE'. The third section is 'Obiecte adăugate până acum' followed by a table with two columns: product name and value. The table lists: 'ieftine' (10.2), 'scumpe' (15.0), 'audi' (13.4), and 'bmw' (14.0). The Android navigation bar is at the bottom.

| Obiecte adăugate până acum | |
|----------------------------|------|
| ieftine | 10.2 |
| scumpe | 15.0 |
| audi | 13.4 |
| bmw | 14.0 |

Tab-ul Calendar

În această pagină observăm calendarul implementat în aplicație care reține pentru fiecare zi progresul corespunzător și oferă posibilitatea modificării tuturor informațiilor folosind butonul “modifică”, și “adaugă” odată ce am completat noile informații.



Observație: Navigarea între paginile oferite de aplicație se poate face prin intermediul săgeții din stânga sus, dar și prin butoanele explicite din josul paginii.



Tab-urile Statistici și Grafice

Pagina despre statistici oferă informații concrete despre productivitatea utilizatorului în raport cu ultimele date cunoscute.

Pagina care implementează grafice are un meniu prin care putem parcurge diferitele grafice, cu informații despre ultimele 14 zile/ultimele 12 săptămâni/ultimele 12 luni



3.2. Baza de date

Pentru a salva datele despre progresul zilnic al unui persoane este necesară folosirea unei baze de date. Primul tabel memorează lista de obiecte adăugată într-o anumită zi și are următoarea structură: ZI / LUNĂ / AN / LISTA_DE_OBIECTE / NOTIȚE.

De asemenea, pentru înregistrarea tuturor produselor este folosit un tabel cu structura: ID_OBIECT / NUME_OBIECT / PREȚ.

Implementarea funcțiilor care gestionează baza de date este prezentată în următoarele imagini:

```
public class DatabaseHelper extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "workregister121.db";
    public static final String TABLE_NAME = "everydaywork_table";
    public static final String COL_1 = "DAY";
    public static final String COL_2 = "MONTH";
    public static final String COL_3 = "YEAR";
    public static final String COL_4 = "ITEMS";
    public static final String COL_5 = "MESAJ";

    public DatabaseHelper(@Nullable Context context) { super(context, DATABASE_NAME, null, version: 1); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TABLE_NAME + " (DAY INT, MONTH INT, YEAR INT, ITEMS TEXT, MESAJ TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int i1) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }

    public void insertData(int Day, int Month, int Year, String Items, String Mesaj)
    {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_1, Day);
        contentValues.put(COL_2, Month);
        contentValues.put(COL_3, Year);
        contentValues.put(COL_4, Items);
        contentValues.put(COL_5, Mesaj);

        db.insert(TABLE_NAME, nullColumnHack: null, contentValues);
        db.close();
    }
}
```

Crearea tabelului ce se ocupă cu memorarea progresului zilnic al utilizatorului

```

public class DatabaseHelperForItems extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "itemregister421.db";
    public static final String TABLE_NAME = "items_table";
    public static final String COL_1 = "ID";
    public static final String COL_2 = "ITEM";
    public static final String COL_3 = "PRICE";

    public DatabaseHelperForItems(@Nullable Context context) {
        super(context, DATABASE_NAME, factory: null, version: 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TABLE_NAME + " (ID INTEGER PRIMARY KEY AUTOINCREMENT, ITEM TEXT, PRICE REAL)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int i1) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }

    public void insertData(model_adapter model_adapter)
    {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues contentValues = new ContentValues();

        contentValues.put(COL_2, model_adapter.getItemName());
        contentValues.put(COL_3, model_adapter.getPrice());

        db.insert(TABLE_NAME, nullColumnHack: null, contentValues);
        db.close();
    }
}

```

Crearea tabelului ce se ocupă cu înregistrarea obiectelor folosite de utilizator

```

myDb.deleteData(intDay, intMonth, intYear);
myDb.insertData(intDay, intMonth, intYear, NewItems, Notita);

Toast.makeText(context: this, text: "Obiecte modificate cu succes", Toast.LENGTH_SHORT).show();

```

Modificarea datelor dintr-o zi anume și afișarea unui mesaj corespunzător

```

createExampleList(myDb.getItems(intDay, intMonth, intYear));
buildRecyclerView();

```

Preluarea obiectelor care corespund zilei căutate și afișarea lor într-o listă

3.3. Algoritmi utilizați

Aplicația folosește memoria foarte eficient utilizând containere predefinite. Algoritmii folosiți constau în sortarea obiectelor în liste în complexitate $O(n \log n)$, dar și algoritmi de căutare binară în listele sortate precedent. Alți algoritmi utilizați sunt cel de interclasare, cel de verificare al input-ului oferit de utilizator, dar și cei folosiți în determinarea statisticilor utilizatorului.

3.4. Probleme întâmpinate

Cea mai mare problemă întâmpinată în finisarea aplicației a fost necesitatea de a adăuga notițe în fiecare zi de către utilizator, funcție care a determinat transformarea completă a bazei de date inițiale. Așadar, am fost nevoit să o redefinesc, oferindu-i mai multe proprietăți și totodată să adaug noi funcții prin care aplicația să poată gestiona baza de date.

Altă problemă a fost cererea de a adăuga valori reale în loc de cele naturale în faza de input, fapt ce a determinat schimbarea multor variabile mai ales în tab-ul de statistici. Alte probleme întâmpinate au fost rezolvate rapid prin intermediu forum-urilor pe care erau precizate diferite soluții foarte folositoare în totalitatea lor.

Capitolul 4: Concluzii

În concluzie proiectul a fost un succes, deoarece și-a atins scopul în ceea ce o privește pe mama mea, a ajutat-o să își noteze în cel mai ușor mod informațiile despre progresul său și chiar să observe o diferență la sfârșitul lunii. În opinia mea, proiectul ar putea fi dezvoltat prin adăugarea a mai multe facilități precum afișarea venitului net (scăzând taxa de asigurare și TVA-ul), iar la final aplicația ar putea fi adăugată pe o platformă specială, la care să aibă acces tot publicul.

WEBOGRAFIE

<https://stackoverflow.com>

<https://docs.oracle.com/en/java>

https://www.w3schools.com/sql/sql_create_db.asp

<https://developer.android.com/docs>

https://youtube.com/playlist?list=PLS1QulWo1RIbVgr0GRuQW5q_K0zb3rrct