

ORAR

LUCRARE PENTRU ATESTAREA COMPETENȚELOR PROFESIONALE

Elev : Panaitescu Constantin

Clasa : a XII - a C

Profesor coordonator : Iuscinschi Simona

CUPRINS

1. Motivul alegerii temei	pag. 3
2. Structura aplicației	pag. 4
3. Detalii tehnice	pag. 8
4. Resurse de hard și soft necesare	pag. 10

Motivul alegerii temei

Tema a fost aleasă cu scopul de a facilita accesul elevilor la orarul școlii și de a le oferi posibilitatea de a-și gestiona evenimentele importante.

Structura aplicației

Rolul aplicației este de a-i oferi utilizatorului accesul la orarul clasei și posibilitatea de a gestiona evenimente.

Aplicația este structurată în două secțiuni. Prima secțiune este responsabilă cu afișarea informației, iar a doua secțiune are rolul de a ajuta utilizatorul să introducă informația în bază de date. Fiecare secțiune reprezintă câte o activitate.

Prima activitate este alcătuită din două liste: orarul și evenimentele.

ORAR Liis		+	⋮
ORAR		EVENIMENTE	
LUNI, 5 MAI			
8:00 8:50	Matematica		
9:00 9:50	Romana		
10:00 10:50	Franceza		
12:10 13:00	Fizica		
11:10 12:00	Aplicatii		
MARTI, 6 MAI			
13:10			
CLASA :		12C	

ORAR Liis

+

⋮

ORAR

EVENIMENTE

Teza Matematica

7 Mai, 2014 08:00

Simulare Matematica

8 Mai, 2014 08:00

Navigare dintre cele două liste se face cu ajutorul tab-urilor prezente în bara de navigație sau prin intermediul gesturilor (swipe). Utilizatorul are posibilitatea de a alege clasa prin intermediul selectorului din partea de jos a ecranului.

În momentul în care lista este în mișcare, selectorul este scos de pe ecran prin intermediul unei animații, cu scopul de a maximiza numărul de elemente ce pot fi vizionate în același timp.

ORAR Liis		+	⋮
ORAR		EVENIMENTE	
LUNI, 5 MAI			
8:00 8:50	Matematica		
9:00 9:50	Romana		
10:00 10:50	Franceza		
12:10 13:00	Fizica		
11:10 12:00	Aplicatii		
MARTI, 6 MAI			
13:10			
CLASA :		12C	

ORAR Liis		+	⋮
ORAR		EVENIMENTE	
10:00 10:50	Franceza		
12:10 13:00	Fizica		
11:10 12:00	Aplicatii		
MARTI, 6 MAI			
13:10 14:00	Engleza		
8:00 8:50	Aplicatii		
9:00 9:50	Aplicatii		

Pentru a putea introduce un eveniment nou, utilizatorul este nevoit să acționeze butonul “+” situat în bară de navigație, acțiune ce va declanșa a două activitate.

Un alt element prezent în bară de navigație este butonul de informații. La acționarea acestuia, pe ecran va apărea un dialog ce precizează realizatorul aplicației și profesorul coordonator.

ADAUGA EVENIMENT

Informatii

Aplicatie realizata de:
Panaiteescu Constantin 12 C

Profesor coordonator:
Iuscinski Simona

INAPOI

ADAUGA

ADAUGA EVENIMENT

Nume

Data

Ziua Ora

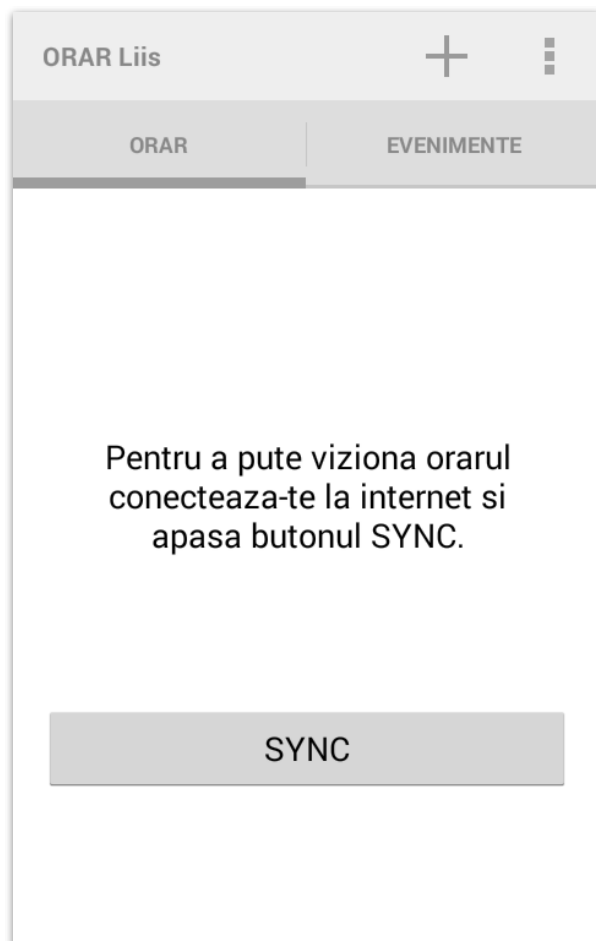
ADAUGA

A două activitate, cea responsabilă cu introducerea datelor, conține câmpurile ce sunt necesare pentru crearea unui nou eveniment. În momentul acționării butonului “ADAUGA” datele introduse sunt verificate, fiind apoi introduse în baza de date.

Pentru reîntoarcerea la activitatea principală, utilizatorul poate acționa titlul barei de navigație ce prezintă, în mod sugestiv, simbolul “<”.

Evenimentele adăugate sunt introduse în lista din activitatea principală.

Pentru a șterge un eveniment, este necesar că utilizatorul să acționeze evenimentul respectiv timp de o secundă. În urmă acestei acțiuni, pe ecran va apărea un dialog cerând confirmarea acțiunii.



Utilizatorul are posibilitatea de a modifica lista de evenimente, adăugând și ștergând evenimente dar nu și orarul. Orarul este sincronizat cu o bază de date situată pe un server. Astfel aplicația verifică de fiecare dată când este deschisă și are acces la internet, versiunea locală cu cea de pe server având grijă că cele două să coincidă.

În cazul în care nu este stocată nici o versiune local (aplicația este deschisă pentru prima dată), și o conexiune la internet nu se poate realiza, un mesaj sugestiv este afișat.

Detalii tehnice

Datele ce alcătuiesc orarul și lista de evenimente sunt stocate într-o bază de date SQLite. Orarul este stocat pe un server, accesul la aceste date făcându-se prin intermediul unui API.

Pentru server este folosit serviciul <https://parse.com> ce oferă accesul la date prin intermediul unui API REST. Astfel putem obține datele stocate cu ajutorul unui request HTTP.

De exemplu, pentru obținerea datelor folosim următoarele url-uri:

- tabelul CLASSES <https://api.parse.com/1/classes/classes>
- tabelul BLOCKS <https://api.parse.com/1/classes/blocks>
- tabelul VERSIONS <https://api.parse.com/1/classes/versions>

Pentru a utiliza cu ușurință informațiile din bază de date avem definite următoarele clase:

```
public class Block {  
  
    public String objectId ;  
    public String name ;  
    public String classId ;  
    public String day;  
    public String start ;  
    public String end ;  
  
    public Block(String id, String name, String _class, String day, String start, String end) {  
        this.objectId = id;  
        this.name = name;  
        this.day = day;  
        this.classId = _class;  
        this.start = start;  
        this.end = end;  
    }  
    public ContentValues getContentValues() {...}  
    public static Block fromCursor(Cursor cursor) {...}  
}
```



```

public class Class {

    public String objectId ;
    public String name ;

    public Class(String id, String name) {
        this.objectId = id;
        this.name = name;
    }

    public ContentValues getContentValues() {...}
    public static Class fromCursor(Cursor cursor) {...}
}

public class Version {

    private String objectId;
    private int data_version;

    public Version(int data_version){...}
    public Version(String id, int data_version){...}

    public ContentValues getContentValues() {...}
    public static Version fromCursor(Cursor cursor) {...}
}

public class Event {

    public int id ;
    public String name ;
    public String date ;

    public Event(String name, String date) {...}
    public Event(int id, String name, String date) {...}

    public ContentValues getContentValues() {...}
    public static Event fromCursor(Cursor cursor) {...}
}

```

Funcțiile `getContentValues()` și `fromCursor()` ajută la introducerea și scoaterea obiectelor din bază de date prin intermediul `ContentProvider`-ului.

Resurse de hard și soft necesare

Aplicația a fost scrisă în java cu scopul de a fi utilizată pe sistemul de operare Android.

Resurse minime :

- sistem de operare : Android - 4.0
- spatiu liber : 2.87 MB
- conexiune internet