Proiect de curs Aplicatii Mobile

Aplicatie de management al produselor

Lucaci Gabriel

Cuprins

Capitolul 1: Introducere ….................................................................................................. 3

Capitolul 2: Tehnologii Aplicate …...................................................................................... 4

Capitolul 3: Proiectarea Aplicatiei ….................................................................................. 5

Capitolul 4: Aspecte de Implementare …........................................................................... 7

Bibliografie …...................................................................................................................... 11

Capitolul 1: Introducere

Telefoanele mobile, mai specific smartphoneurile, sunt probabil una dintre cele mai importante inventii recente, majoritatea populatiei avand macar unul in ziua de azi. Pe langa functionalitatea de la care au inceput telefoanele, aceea de a putea sa comunici cu cineva la o distanta mare, smartphoneurile pe care le avem in prezent au o multitudine de alte functionalitati datorita pe de o parte numeroaselor aplicatii dezvoltate pentru ele.

Pe un subiect complet diferit la prima vedere, magazinele sunt la fel o parte foarte importanta a vietii noastre de zi cu zi, iar cu mai multe produse din diverse domenii existand pe piata care trebuie organizate si inregistrate intr-un fel sau altul de catre magazine pentru ca acestea sa poata functiona eficient, metode de organizare mai eficiente sau mai usor de utilizat sunt necesare.

Considerand astfel aceasta nevoie de a organiza mai bine/mai usor diferitele produse ale unui magazin si faptul ca smartphoneurile sunt atat de comune in zilele noastre, eu m-am decis sa creez o aplicatie care incearca sa ajute organizarea mai usoara a produselor si inregistrearea lor.

Aplicatia mea are scopul de a face treaba angajatiilor ce se ocupa cu orgnaizarea produselor din magazine mai usoare lasandu-i sa poata adauga, elimina si edita produse intr-o lista ce poate fi trimisa mai apoi la un server ce mentine o baza de date pentru a putea fi accesate aceste date de mai multi utilizatori sau ca si un backup in caz ca aplicatia locala isi pierde datele.

Capitolul 2: Tehnologii Aplicate

2.1 Android Studio – Mediul de dezvoltare aplicatie client

Android Studio este principalul si oficialul mediu de dezvoltare al aplicatiilor pentru platforma Android si este oferit gratuit de Google. Platforma se bazeaza pe editia de comunitate IntelliJ IDEA open source si include o gama larga de instrumente si functii pentru a sprijini intregul ciclu de viata al dezvoltarii aplicatiilor Android. Acesta include editarea codului, depanarea, testarea si implementarea.

2.2.1 Visul Studio – Mediul de dezvoltare aplicatie server

Visual Studio este un mediu de dezvoltare integrat (IDE) puternic utilizat pentru dezvoltarea unei game largi de aplicatii. Este dezvoltat de Microsoft si este utilizat pe scara larga. Interfata sa intuitiva, suportul pentru mai multe limbaje de programare si setul robust dee instrumente de dezvoltare il fac o alegere populara pentru dezvoltatorii din intreaga lume.

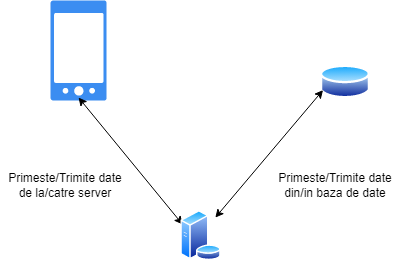
2.2.2 Entity Framework

Entity Framework este un cadru de cartografiere obiect-relationala (ORM) dezvoltat de Microsoft pentru aplicatii .NET. Acesta permite dezvoltatorilor sa lucreze cu date sub forma de obiecte .NET, mai degraba decat sa scrie interogari SQL traditionale pentru a interactiona cu o baza de date. Entity Framework simplifica procesul de acces si de gestionare a datelor, permitand dezvoltatorilor sa scrie cod care interactioneaza cu datele folosind interogari LINQ si clase C# sau VB.NET. Elimina necesitatea dezvoltatorilor de a scrie interogari SQL complexe sau de a interactiona direct cu schema bazei de date.

Capitolul 3: Proiectarea Aplicatiei

3.1 Descriere

Aplicatia de management a produselor are rolul de a ajuta angajatii unui magazin cu organizarea si inregistrarea produselor acelui magazin. Aceasta lasa utilizatorul sa acceseze datele dintr-o baza de date accesata prin intermediul serverului care vor fi incarcate in aplicatie dupa care utilizatorul poate lucra cu aceste, adaugand noi produse in lista sau eliminand produse existente, aceste schimbari putand fi apoi trimise la server pentru a actualiza baza de date cu schimbarile facute de utilizator.



3.2 Diagrama de clase

In imaginea de mai jos sunt reprezentate clasele folosite in aplicatie si legaturile lor.

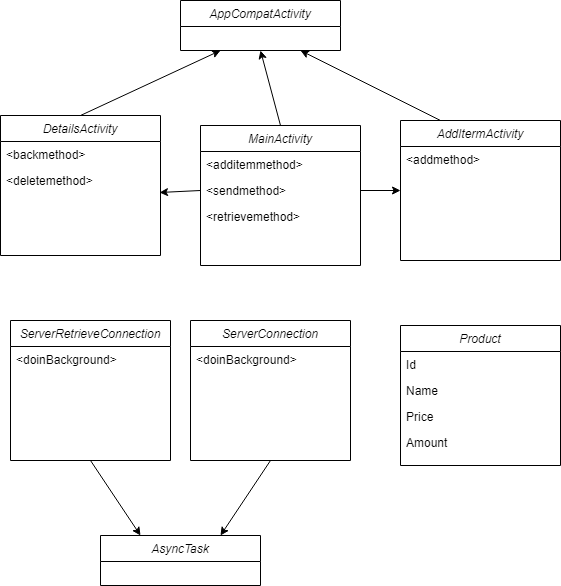
Clasa MainActivity este responsabila de ecranul principal ala aplicatiei, unde se afiseaza lista de produse si se pot adauga noi produse, trimite date la server sau cere datele de la server.

Clasa DetailsActivity este responsabila de ecranul cu detalii care se deschide cand utilizatorul da click pe unul din produsele din lista si stergerea acelui produs.

Clasa AddItemActivity este responsabila de adaugarea produselor noi in lista.

Clasa Product reprezinta structura de date a unui produs.

Clasa ServerConnection si ServerRetrieveConnection sunt responsabile de trimiterea de date si cererea de date de la server.



Capitolul 4: Aspecte de implementare

4.1 Lista de produse si adaugare/stergere

Lista ce apare pe ecranul principal este realizata folosind un arrayadapter ce se leaga de listviewul ce se ocupa cu afisarea listei si practic afiseaza doar o lista a numelor produselor, nu lista cu toate datele. Doar cand se da click pe unul din produsele din aceasta lista se va apela ecranul cu detalii unde bazat pe ce item din lista a fost apasat, se va accesa intrarea in lista cu produse cu acel nume si vor fi afisate detaliile acestuia.

Ecranul cu detalii este in mod normal doar un sablon, cu textviewuri fara text propriu zis in ele, care vorfi setate la valorile corecte doar la accesare bazat pe campurile produsului accesat. Butonul de stergere din aceste ecran trimite inapoi la activitatea principala numele si id-ul acelui produs, pentru a putea identifica corect care produs trebuie mai apoi sters din lista.

Butonul de adaugare la fel ca si cu ecranul de detalii va fi apela un ecran sablon in care utilizatorul va completa campurile iar apoi aceste campuri vor fi de unde se iau valorile ce vor compune noul produs care va fi adaugat in lista.

4.2 Trimiterea de date la server

Cand se da click pe butonul de trimitere date prima data se va converti lista de produse intr-un string json care va putea apoi sa fie trimis folosind un protocol tcp la server, care va procesa acest string si apoi va adauga produsele ce nu sunt deja in baza de date sau va sterge pe cele ce sunt prezente in baza de date dar nu sunt prezente in lista.

//partea ce conversie in json

public void sendmethod(View view) throws JsonProcessingException {  
 if(!items.isEmpty()) {  
 ObjectMapper mapper = new ObjectMapper();  
 String jsonstring = "[";  
 for (Product p : items) {  
 jsonstring = jsonstring + mapper.writeValueAsString(p) + ",";  
 }  
 jsonstring = jsonstring.substring(0, jsonstring.length() - 1) + "]";  
 System.*out*.println(jsonstring);  
 ServerConnection conn = new ServerConnection();  
 conn.execute(jsonstring);  
 }  
 else Toast.*makeText*(this, "List of items empty", Toast.*LENGTH\_SHORT*).show();  
}

//partea de conectare

@Override  
protected Void doInBackground(String...strings) {  
  
 for(String string:strings)  
 {  
 message=message+string;  
 }  
 try {  
 String received="";  
 s = new Socket(serverip, 8989);  
 pw = new PrintWriter(s.getOutputStream());  
 pw.write(message);  
 pw.flush();  
 pw.close();  
 s.close();  
 }  
 catch (IOException e) {  
 e.printStackTrace();  
 }  
 return null;  
}

4.3 Cererea de date la server

Respectiv cand se da click pe butonul de cerere date se vor converti datele din baza de date intr-un string json in acelasi stil care va fi trimis printr-un protocol tcp in acelasi stil la client care va procesa acest string si va salva datele ce rezulta in lista de produse a clientului.

//partea ce conversie in json

public void retrievemethod(View view) throws JsonProcessingException {  
 ServerRetrieveConnection conn = new ServerRetrieveConnection();  
 String connstring="Retrieve";  
 conn.execute(connstring);  
 String result="";  
 try {  
 result = conn.get();  
 } catch (ExecutionException | InterruptedException e) {  
 e.printStackTrace();  
 }  
 System.*out*.println(result);  
 Product[] auxlist;  
 ObjectMapper mapper = new ObjectMapper();  
 auxlist=mapper.readValue(result,Product[].class);  
 for(Product aux:auxlist) {  
 int oktoadd = 1;  
 for (Product p : items) {  
 if (Objects.*equals*(aux.Name, p.Name) && Objects.*equals*(aux.Id, p.Id)) {  
 oktoadd = 0;  
 }  
 }  
 if (oktoadd == 1) {  
 items.add(aux);  
 itemnames.add(aux.Name);  
 arrayAdapter.notifyDataSetChanged();  
 }  
 }

//partea de conectare

protected String doInBackground(String...strings) {  
  
 for(String string:strings)  
 {  
 message=message+string;  
 }  
 try {  
 s = new Socket(serverip, 8989);  
 pw = new PrintWriter(s.getOutputStream());  
 pw.write(message);  
 pw.flush();  
 InputStream inputStream = s.getInputStream();  
 BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream));  
 received=reader.readLine();  
 s.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return received;  
}

Bibliografie

1. Documentatie Android Studio <https://developer.android.com/docs>
2. Documentatie Visual Studio <https://learn.microsoft.com/en-us/visualstudio/windows/?view=vs-2022>
3. Documentatie Entity Framework <https://learn.microsoft.com/en-us/ef>
4. Librarie folosita pentru conversia in/din json <https://github.com/FasterXML/jackson>
5. Wikipedia Android Studio <https://en.wikipedia.org/wiki/Android_Studio>
6. Wikipedia Visual Studio <https://en.wikipedia.org/wiki/Visual_Studio>
7. Wikipedia Entity Framework <https://en.wikipedia.org/wiki/Entity_Framework>