



Facultatea de  
Automatică și  
Calculatoare



---

# **SPOT MY ACNE - APLICAȚIE MOBILĂ PENTRU MONITORIZAREA ACNEEI**

**Proiect realizat la *Codesign Hardware-Software***

Student:  
**Claudia BURCEA**

Timisoara  
Ianuarie, 2024



# Cuprins

<b>1 Introducere</b>	<b>3</b>
1.1 Context . . . . .	3
<b>2 Analiza domeniului/ State-of-the-art</b>	<b>5</b>
2.1 Aplicații mobile asemănătoare . . . . .	5
<b>3 Tehnologii folosite</b>	<b>7</b>
3.1 Baza de date . . . . .	7
3.1.1 Firebase Cloud Storage . . . . .	7
3.2 Java . . . . .	7
3.3 Versionarea codului - Git . . . . .	7
3.4 Android Studio . . . . .	7
<b>4 Proiectarea sistemului</b>	<b>9</b>
4.1 Diagrama UML a cazurilor de utilizare . . . . .	9
4.1.1 Sistemul . . . . .	10
4.1.2 Actorii . . . . .	10
4.1.3 Asocierile dintre un actor și un caz de utilizare . . . . .	10
4.1.4 Asocierile dintre două cazuri de utilizare . . . . .	10
4.2 Proiectarea bazei de date . . . . .	10
4.2.1 Diagrama UML a bazei de date . . . . .	10
4.2.2 Securitatea bazei de date . . . . .	10
<b>5 Implementarea funcționalităților</b>	<b>11</b>
5.1 Înregistrarea în aplicație . . . . .	11
5.2 Monitorizarei acneei cu ajutorul calendarului . . . . .	12
<b>6 Utilizarea aplicației</b>	<b>15</b>
6.1 Înregistrarea în aplicație . . . . .	15
6.2 Ecranul Home . . . . .	15
6.2.1 Monitorizarea evoluției acneei cu ajutorul calendarului . . . . .	18
6.2.2 Deschiderea camerei telefonului . . . . .	18
6.2.3 Efectuarea fotografiei . . . . .	18



# **Capitolul 1**

## **Introducere**

### **1.1 Context**

"Spot My Acne" este o aplicație mobilă concepută pentru monitorizarea acneei și facilitarea accesului la serviciile dermatologice. Această inițiativă a luat nastere dintr-o perspectivă personală, bazată pe experiența mea directă cu acneea, motivându-mă să dezvolt o soluție mai amplă de gestionare a acestei probleme cu care se confruntă mulți oameni în viața de zi cu zi. Am observat că gestionarea acneei poate fi adesea dificilă și, uneori, copleșitoare, iar nevoia de a găsi specialiști calificați pentru îngrijirea pielii este crucial.

În esență, scopul aplicatiei este de a transforma procesul de îngrijire a pielii într-o experiență simplă, accesibilă, convenabilă și personalizată pentru fiecare utilizator.

"Spot My Acne" aduce în prim plan două funcționalități esențiale pentru a facilita gestionarea acneei.

Funcționalitatea de calendar permite utilizatorilor să înregistreze și să monitorizeze cu ușurință rutina lor zilnică de îngrijire a pielii. Această caracteristică nu doar evidențiază obiceiurile zilnice, ci oferă și o privire detaliată asupra progresului pielii în timp real, prin fotografierea zilnică a suprafetei afectate.

Funcționalitatea de localizare a medicilor dermatologi facilitează accesul rapid la îngrijirea specializată. Utilizatorii pot găsi cu ușurință experți în dermatologie, eliminând procesul lung și adesea complicat de căutare a medicilor potriviti.



# **Capitolul 2**

## **Analiza domeniului/ State-of-the-art**

### **2.1 Aplicații mobile asemănătoare**

Pentru a compara sistemul dezvoltat, am căutat două aplicații pentru Android ce pun la dispoziția utilizatorilor funcționalități asemănătoare cu "Spot My Acne": **Acné Monitor** și **Rutina de îngrijire a pielii**, prima depășind 100 000 de descărcari, iar cea de-a doua, 500.

Tabelul 2.1 compară sistemul "Spot My Acne" cu cele două sisteme și putem observa ușor că majoritatea funcționalităților ce se regăsesc în acestea sunt disponibile și utilizatorilor aplicației dezvoltate de mine, mai puțin acestea trei: obținerea sfaturilor pentru tratament, analiza și gestionarea produselor de îngrijire a pielii și temporizatoare și mementouri. După ce s-au conectat în aplicație, utilizatorii pot accesa calendarul în care acestia pot crea și modifica o rutină zilnică proprie, adăugând fotografii pentru observarea evoluției acneei.

Pe lângă toate activitățile la care un pacient are acces în cadrul aplicațiilor **Acné Monitor** sau **Rutina de îngrijire a pielii**, doar sistemul "Spot My Acne" facilitează accesul la servicii medicale de specialitate, punând la dispoziție informații utile despre locația cabinetelor dermatologice. Astfel, aplicația dezvoltată de mine pune la dispoziția utilizatorilor posibilitatea de a găsi cu ușurință specialiști în domeniul dermatologiei în zona lor sau o zonă aleasă de acestia. Această caracteristică vine în întâmpinarea nevoilor utilizatorilor care caută asistență profesională în tratarea problemelor lor de piele, oferind o conexiune rapidă și eficientă cu experti medicali. Consider că această funcționalitate este importantă, deoarece localizarea rapidă a cabinetelor dermatologice poate contribui semnificativ la accelerarea procesului de diagnosticare și tratament, îmbunătățind calitatea asistenței medicale primite de utilizatori.

Tabelul 2.1: Analiză comparativă între cele mai apreciate două aplicații mobile pentru sistemul de operare Android (Rutina de îngrijire a pielii & Acné Monitor) și "Spot My Acne"; funcționalitățile analizate regăsite într-o aplicație sunt marcate cu ✓, iar cele absente cu ✗

Caracteristici și funcționalități	Rutina de îngrijire a pielii	Acné Monitor	Spot My Acne
Sistem de operare	Android	Android	Android
Magazin de aplicații	Google Play	Google Play	✗
Nota din Google Play	4,8/5	✗	✗
Număr de instalări	100k+	500+	✗
Număr de ratinguri	666	✗	✗
Crearea unei rutine zilnice cu ajutorul calendarului	✓	✓	✓
Modificarea rutinei zilnice	✓	✗	✓
Fotografierea feței pentru urmărirea evoluției acneei	✗	✓	✓
Limba română	✗	✗	✓
Localizarea cabinetelor dermatologice pe hartă	✗	✗	✓
Obținerea sfaturilor pentru tratament	✗	✓	✗
Analiza și gestionarea produselor de îngrijire a pielii	✓	✗	✗
Temporizatoare și mementouri	✓	✗	✗

# **Capitolul 3**

## **Tehnologii folosite**

### **3.1 Baza de date**

#### **3.1.1 Firebase Cloud Storage**

### **3.2 Java**

### **3.3 Versionarea codului - Git**

<https://github.com/claudiab13/SpotMyAcne>

[https://github.com/claudiab13/SpotMyAcne<sub>Updated</sub>](https://github.com/claudiab13/SpotMyAcneUpdated)

### **3.4 Android Studio**



# Capitolul 4

## Proiectarea sistemului

### 4.1 Diagrama UML a cazurilor de utilizare

Diagrama cazurilor de utilizare este cea mai simplă reprezentare a interacțiunii utilizatorului cu sistemul, dar și a specificațiilor unui produs. Aceasta arată relația dintre utilizator și diferitele cazuri de utilizare în care este implicat. O diagramă de cazuri de utilizare poate identifica diferitele tipuri de utilizatori ai unui sistem și diferitele cazuri de utilizare și va fi însotită adesea și de alte tipuri de diagrame. Cazurile de utilizare sunt reprezentate fie prin cercuri, fie prin elipsă.

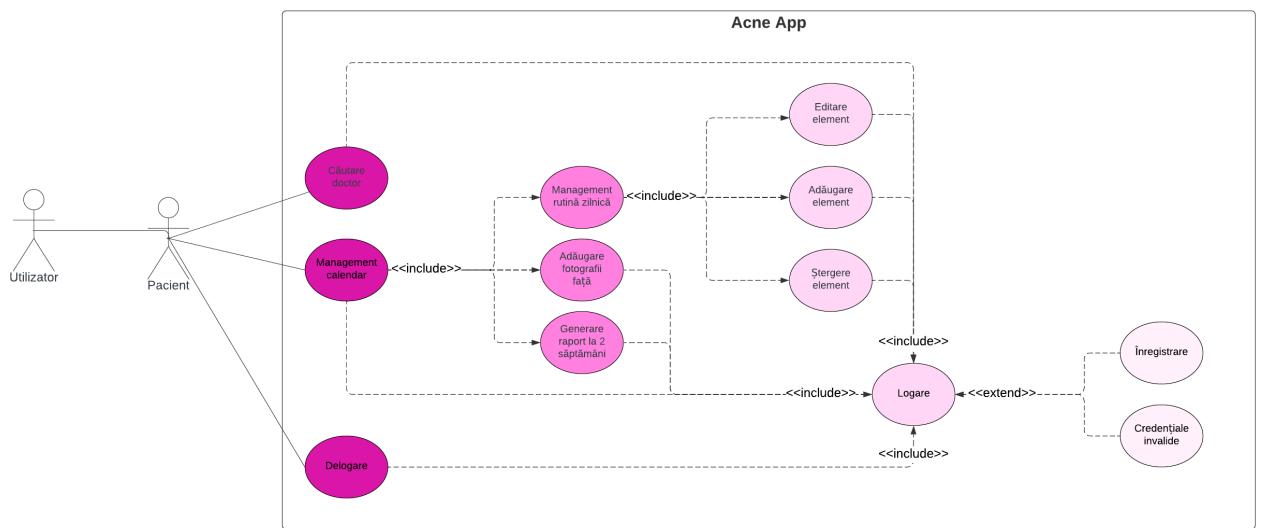


Figura 4.1: Diagrama UML a cazurilor de utilizare: actorii și principalele operații puse la dispoziția acestora.

**4.1.1 Sistemul****4.1.2 Actorii****4.1.3 Asocierile dintre un actor și un caz de utilizare****4.1.4 Asocierile dintre două cazuri de utilizare****4.2 Proiectarea bazei de date****4.2.1 Diagrama UML a bazei de date****4.2.2 Securitatea bazei de date**

# Capitolul 5

## Implementarea funcționalităților

### 5.1 Înregistrarea în aplicație

Activitatea de înregistrare este o componentă a aplicației Spot My Acne, care oferă utilizatorilor posibilitatea de a crea un cont personal în scopul utilizării funcționalităților aplicatiei. Fiind un singur tip de utilizatori, există o singură activitate de înregistrare în aplicație. Această activitate implică utilizarea unei interfețe de utilizator simplificate, care include câmpuri **EditText** pentru adresa de email și parolă.

Butonul **registerButton** declanșează procesul de înregistrare atunci când este apăsat. Odată ce utilizatorul completează câmpurile de email și parolă și apasă butonul, se încearcă crearea contului utilizatorului în Firebase Authentication.

De asemenea se fac verificări cu privire la datele completate: adresa de email trebuie să fie una validă, iar parola trebuie să contină cel puțin 6 caractere.

```
String email = registerEmail.getText().toString();
String password = registerPassword.getText().toString();

mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener(
    RegisterActivity.this, new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful()) {
            Toast.makeText(RegisterActivity.this, "Successful_Registration!",
                Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(RegisterActivity.this, "Enter_a_valid_email_address_&a
                _minimum_6_characters_password.", Toast.LENGTH_SHORT).show();
        }
    }
});
```

Secvența de cod 5.1: Memorarea în Firebase a datelor utilizatorului afisarea mesajului ce indică succesul sau eventuala problema ce apare în procesul de înregistrare

După finalizarea procesului de înregistrare, utilizatorul primește un mesaj scurt sub formă de notificare (**Toast**) indicând succesul sau eventuala problemă înregistrată în timpul procesului.

## 5.2 Monitorizarei acneeii cu ajutorul calendarului

Această activitate permite utilizatorilor să selecteze date din calendar, să realizeze fotografii, să salveze imaginile asociate și să vizualizeze fotografiile salvate pentru datele alese. Utilizarea Firebase Storage și Realtime Database asigură stocarea și recuperarea eficientă a datelor în cloud.

Utilizatorul poate alege o dată din calendar folosind **CalendarView**. Atunci când acesta selectează o dată, este afișat un mesaj într-un **Toast** care indică data aleasă.

```
calendarView.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
    @Override
    public void onSelectedDayChange(@NonNull CalendarView calendarView, int year, int
        month, int day) {
        month = month + 1;
        Toast.makeText(CalendarActivity.this, day + "/" + month + "/" + year, Toast.
            LENGTH_SHORT).show();
        stringDateSelected = day + "-" + month + "-" + year;
    }
});
```

Secvența de cod 5.2: Selectarea datei.

După ce a dat clic pe o zi din calendar, utilizatorul poate să apese pe **ImageButton**-ul cu id-ul **takephoto** pentru a accesa camera și a realiza o fotografie, care este afișată într-un **ImageView** (photo). Această funcționalitate este implementată în metoda **openCamera()** și **onActivityResult**.

```
private void openCamera() {
    //Toast.makeText(this, "Camera Open Request", Toast.LENGTH_SHORT).show();
    Intent camera = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    startActivityForResult(camera, CAMERA_REQUEST_CODE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @NonNull Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == CAMERA_REQUEST_CODE) {
        Bitmap image = (Bitmap) Objects.requireNonNull(data.getExtras()).get("data");
        photo.setImageBitmap(image);
    }
}
```

Secvența de cod 5.3: Fotografierea și afișarea în ImageView.

Utilizatorul poate salva fotografie apăsând butonul **Save Photo** (savephoto). Imaginea este salvată în Firebase Storage. O referință **StorageReference** este creată pentru a desemna locația exactă în care imaginea va fi stocată în Firebase Storage. Această referință include un nume unic generat pe baza datei și unui timestamp, astfel încât să fie asigurată unicitatea fiecărei imagini.

```
StorageReference imageReference = storageReference.child(stringDateSelected + "_" +
    System.currentTimeMillis() + "." + getFileExtension(uri));
imageReference.putFile(uri).addOnSuccessListener(new OnSuccessListener<UploadTask.
    TaskSnapshot>() {
    ...
});
```

Secvența de cod 5.4: Salvarea fotografiei în Firebase Storage (cod selectat din metoda "uploadToFirebase".

După încărcarea cu succes a imaginii în Firebase Storage, este necesar să se obțină URL-ul de descărcare al imaginii. Acest URL poate fi accesat prin intermediul referinței imaginii.

```
imageReference.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
    ...
});
```

Secvența de cod 5.5: Obținerea URL-ului de descărcare al imaginii (cod selectat din metoda "uploadToFirebase".

URL-ul de descărcare este apoi stocat în Firebase Realtime Database, sub nodul corespunzător datei selectate în calendar.

```
databaseReference.child(stringDateSelected).setValue(uri.toString());
```

Secvența de cod 5.6: Stocarea URL-ului de descărcare al imaginii în Firebase Realtime Database (cod selectat din metoda "uploadToFirebase".

Utilizatorul poate să vizualizeze fotografia asociată datei selectate. În metoda calendarClicked(), fotografia asociată datei selectate este recuperată din Firebase Storage și afișată în ImageView.

```
private void calendarClicked() {
    databaseReference.child(stringDateSelected).addListenerForSingleValueEvent(new
        ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    String uri = snapshot.getValue(String.class);
                    if (uri != null) {
                        Picasso.get().load(uri).into(photo);
                    } else {
                        Toast.makeText(CalendarActivity.this, "No_image_found_for_this_
                            date", Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(CalendarActivity.this, "No_data_found_for_this_date",
                        Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(CalendarActivity.this, "Database_error:" + error.
                    getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
}
```

Secvența de cod 5.7: Vizualizarea fotografiei.



# **Capitolul 6**

## **Utilizarea aplicației**

### **6.1 Înregistrarea în aplicație**

În cazul în care nu este deja conectat la aplicație, prima activitate pe care utilizatorul o vede în momentul deschiderii aplicației este pagina de autentificare, ecranul telefonului arătând exact ca în figura 6.1. În cazul în care utilizatorul nu are încă un cont creat, va fi nevoie să apese butonul "Register", aplicația nepermitându-i acestuia să efectueze nicio altă activitate fără a fi autentificat. Pentru a crea un cont, utilizatorul trebuie să introducă numele, prenumele, username-ul și parola . Formularul pe care utilizatorul va fi nevoie să îl completeze este cel din figura 6.2.

### **6.2 Ecranul Home**

După ce s-a logat, utilizatorul se poate folosi de cele 2 mari funcționalități ale aplicației. Vor fi afișate 2 butoane: My Calendar și Map, precum în figura 6.3.

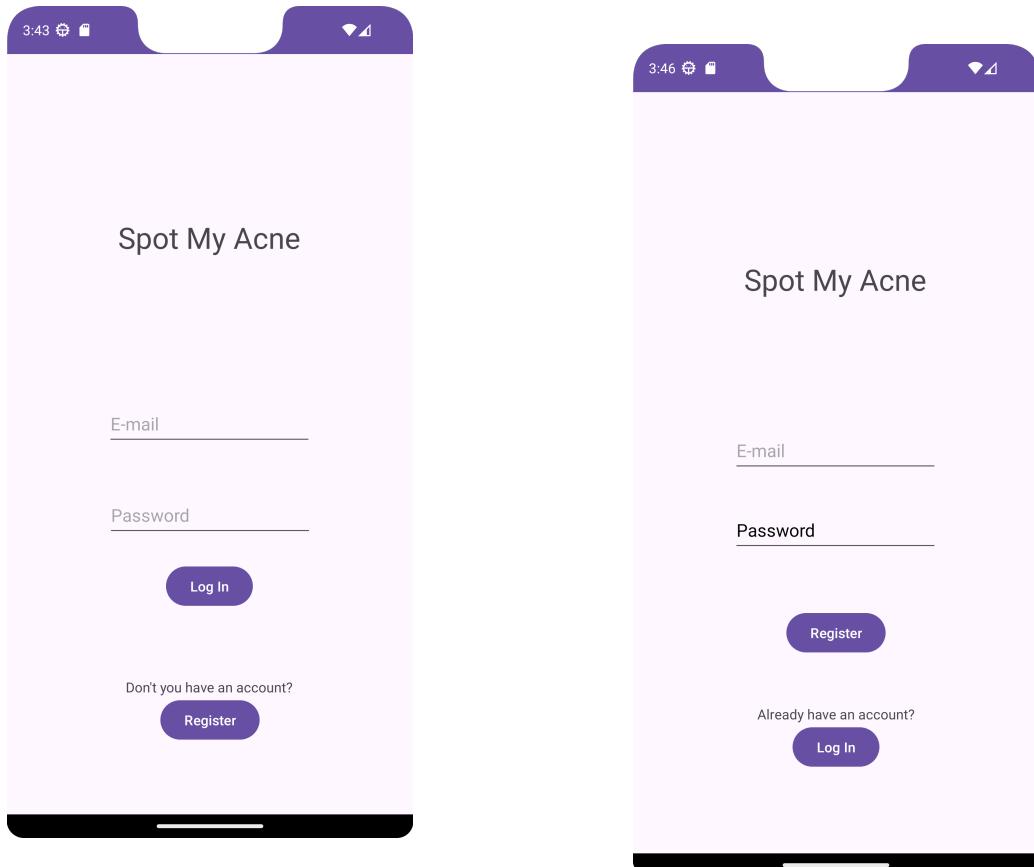


Figura 6.1: Autentificarea în aplicatie.

Figura 6.2: Înregistrarea în aplicatie.

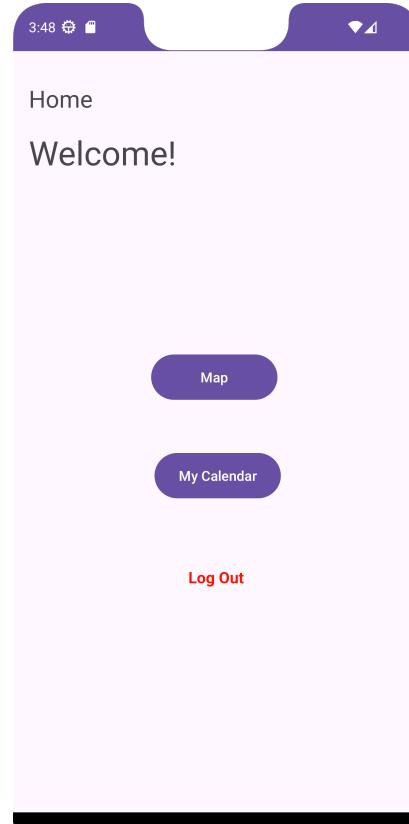


Figura 6.3: Ecranul telefonului după ce utilizatorul s-a logat.

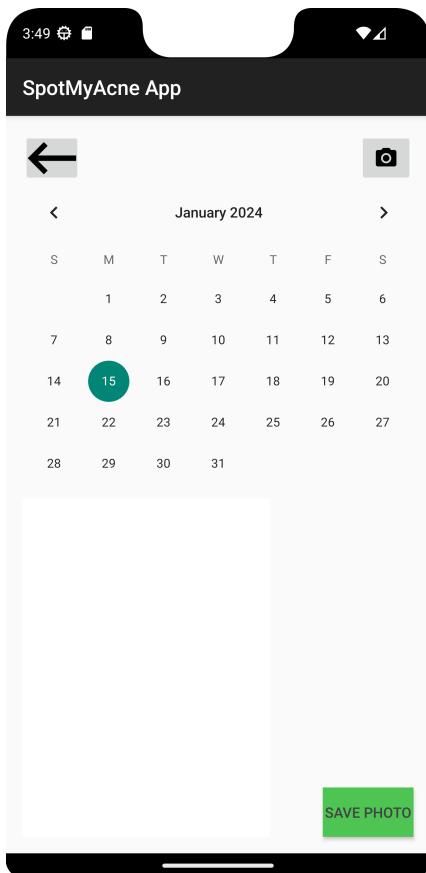


Figura 6.4: Calendar.



Figura 6.5: Camera.



Figura 6.6: Fotografie efectuată.

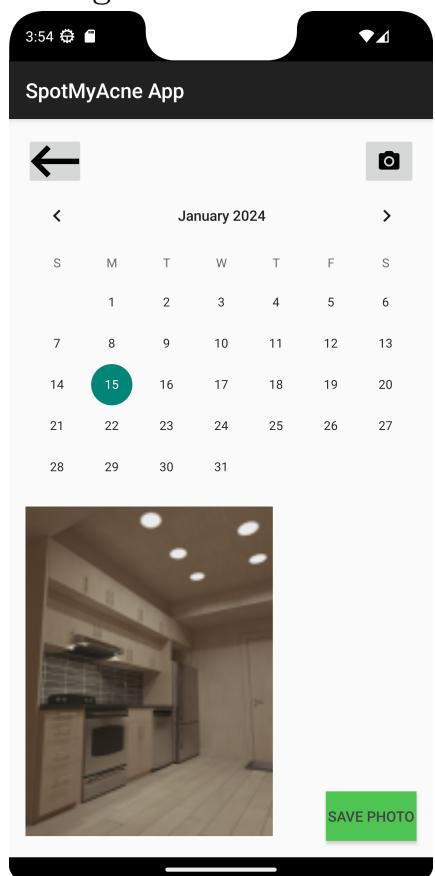


Figura 6.7: Fotografie atașată la ImageView.

### **6.2.1 Monitorizarea evoluției acneei cu ajutorul calendarului**

Utilizatorul poate să dea clic pe una din zilele calendarului (figura 6.4).

### **6.2.2 Deschiderea camerei telefonului**

După ce selectează o zi din calendar, utilizatorul poate apăsa pe ImageButton-ul ce va deschide camera telefonului, precum în figura 6.5.

### **6.2.3 Efectuarea fotografiei**

După ce utilizatorul face fotografia (figura 6.6), aceasta va fi atașată în ImageView-ul din activitatea calendar, precum în figura 6.7.

# **Lista secvențelor de cod**

5.1 Memorarea în Firebase a datelor utilizatorului afisarea mesajului ce indica succesul sau eventuala problema ce apare în procesul de înregistrare . . . . .	11
5.2 Selectarea datei. . . . .	12
5.3 Fotografierea și afișarea în ImageView. . . . .	12
5.4 Salvarea fotografiei în Firebase Storage (cod selectat din metoda "uploadToFirebase." . . . . .	12
5.5 Obținerea URL-ului de descărcare al imaginii (cod selectat din metoda "uploadToFirebase." . . . . .	13
5.6 Stocarea URL-ului de descărcare al imaginii în Firebase Realtime Database (cod selectat din metoda "uploadToFirebase." . . . . .	13
5.7 Vizualizarea fotografiei. . . . .	13



# **Lista figurilor**

## **Capitolul 4**

4.1	Diagrama UML a cazurilor de utilizare: actorii și principalele operații puse la dispoziția acestora. . . . .	9
-----	--	---

## **Capitolul 6**

6.1	Autentificarea în aplicație. . . . .	16
6.2	Înregistrarea în aplicație. . . . .	16
6.3	Ecranul telefonului după ce utilizatorul s-a logat. . . . .	16
6.4	Calendar. . . . .	17
6.5	Camera. . . . .	17
6.6	Fotografie efectuată. . . . .	17
6.7	Fotografie atașată la ImageView. . . . .	17



## **Lista tabelelor**

- 2.1 Analiză comparativă între cele mai apreciate două aplicații mobile pentru sistemul de operare Android (Rutina de îngrijire a pielii & Acné Monitor) și "Spot My Acne"; funcționalitățile analizate regăsite într-o aplicație sunt marcate cu ✓, iar cele absente cu ✗ . 6