UNIVERSITATEA ALEXANDRU IOAN CUZA IAŞI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENŢĂ

MyPhoneIsLost

propusă de

Alex Gabriel Tudosa

Sesiunea: *februarie, 2020*

Coordonator ştiinţific

Lect. Dr. Anca Ignat

UNIVERSITATEA ALEXANDRU IOAN CUZA IAŞI

FACULTATEA DE INFORMATICĂ

MyPhoneIsLost

Alex Gabriel Tudosa

Sesiunea: *februarie, 2020*

Coordonator științific

Lect. Dr. Anca Ignat

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul "MyPhoneIsLost", codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza Iași să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, data

Absolvent Prenume Nume

_____

(semnătura în original)

ACORD PRIVIND PROPRIETATEA DREPTULUI DE AUTOR

Facultatea de Informatică este de acord ca drepturile de autor asupra programelor-calculator, în format executabil și sursă, să aparțină autorului prezentei lucrări, Alex-Gabriel Tudosă.

Încheierea acestui acord este necesară din următoarele motive:

- *dorința de continua dezvoltarea aplicației*
- *Aporturi aduse:*
  - *Personal – 60%*
  - *Tehnologic – 40%:*
    - *Navigation Component – 5%*
    - *BroadcastReceiver – 10%*
    - *Location Service – 5%*
    - *Beacon Proximity Detection API – 15%*
    - *Background Job Scheduler – 5%*

Iași, *data*

Decan *Prenume Nume* Absolvent *Prenume Nume*

_____ _____

(semnătura în original) (semnătura în original)

Cuprins

Introducere

Motivația lucrării

Telefoanele mobile s-au transformat într-o necesitate pentru aproape orice om de pe planetă. Necesitatea de a se menține în contact cu familia sau prietenii, accesul la rețelele de socializare sau e-mail, dar și în administrarea afacerilor sunt doar câteva dintre motivele pentru care telefoanele mobile au devenit importante în viața de zi cu zi a omului.

Fiecare persoană care deține un telefon cu sistem de operare android întâmpină o problemă pe care deseori nu o conștientizează datorită vieților noastre cu un program aglomerat, cea de a uita telefonul, la birou, la școală, la facultate, la un prieten, într-un mijloc de transport sau într-un loc public. Când se întâmplă o situație de genul acesta deseori nu ne amintim unde am uitat telefonul sau chiar dacă ne amintim și ne întoarcem la locul în care a fost uitat, de cele mai multe ori nu îl mai găsim datorită unor persoane rău intenționate. În majoritatea situațiilor telefoanele pierdute sunt protejate cu diverse parole, însă nu au internetul sau Wi-fi-ul pornit și din acest motiv nu îl putem găsi cu ajutorul conturilor Google folosite de fiecare telefon cu sistemul de operare android. Acest lucru este datorat faptului că atât aplicațiile create de Google cât și mulți alți dezvoltatori de aplicații android și-au îndreptat atenția pe crearea de aplicații care pot să găsească telefonul doar cu condiția ca utilizatorii să aibă internetul și locația telefonului pornită. Prin faptul că una din cele două condiții nu este respectată de utilizatori, majoritatea persoanelor care se află în ipostaza de a avea telefonul pierdut nu vor reuși niciodată să își recupereze bunul pierdut.

Însă mai sunt situațiile în care telefonul cade după un birou, un pat sau orice obiect de mobilier fără a observa acest lucru și atunci pierdem foarte mult timp în căutarea acestuia și de multe ori ne dorim să avem o modalitate pentru a-l putea depista.

Obiectivul lucrării

Lucrarea „*MyPhoneIsLost”* încearcă dezvoltarea unor soluții care să rezolve două probleme pe care le poate întâmpina o persoană. Una fiind pentru situațiile în care telefonul este pierdut într-un spațiu public sau ne este furat de o persoană rău intenționată. Cea de a doua fiind pentru cazurile în care telefonul este de negăsit într-o zonă apropiată de utilizator cum ar fi în locuința proprie, acest lucru fiind posibil printr-un afișaj vizual de tip cald și rece.

Sumar

Cele trei capitole ale lucrării de licență sunt:

- Componente hardware
- Implementarea soluției
- Evoluția și rezultatele soluției

În primul capitol, *Componente hardware*, este prezentată descrierea caracteristicilor și modul de funcționare.

În continuare, capitolul 2, *Implementarea soluției*, sunt descrise concepte și tehnologii, analiza aplicației și arhitectura sistemului.

În ultimul capitol este descrisă evoluția aplicației *„MyPhoneIsLost”* însoțită de rezultatele acesteia în urma dezvoltării, dar și îmbunătățirile care ar mai putea fi aduse.

Contribuții

Această aplicație dorește să vină în ajutorul persoanelor care doresc să își găsească telefonul. Pentru situațiile în care acesta este pierdut într-un spatiu public utilizatorul va folosi aplicația pe un alt telefon prin intermediul căruia va trimire un SMS către dispozitivul pierdut cu un mesaj specific. Aplicația de pe telefonul pierdut îl va citi și va accesa locația acestuia pentru a se prelua coordonatele care vor fi trimise printr-un mesaj către expeditor.

Pentru cazurile în care telefonul este pierdut într-un spațiu din apropierea utilizatorului precum locuința proprie va fi folosit un senzor de rece si cald care va indica cat de aproape sau departe este telefonul de utilizator.

Aplicația este mult mai sigură pentru utilizator în comparație cu alte aplicații care sunt deja pe piață, deoarece aceasta nu cere să aibă acces la datele confidențiale, ea necesitând doar permisiuni pentru dreptul de a accesa locația, Bluetooth-ul, accesul de trimitere și citire mesaje.

1. Componente hardware

1.1. GPS

GPS reprezintă sistemul de poziționare globală și a fost dezvoltat de armata americană în 1973, dar a fost lansat în scopuri civile în anii 80. Acesta a fost inițial folosit cu 24 de sateliți, dar acum există 31 de sateliți GPS operaționali pe orbită. Telefonul cu sistem de operare android comunică cu acești sateliți printr-o antenă GPS, care face parte din hardware în marea majoritatea smartphone-urilor și tabletelor actuale. Antena este conectată prin intermediul unui driver la software.

1.2. Bluetooth

Tehnologia Bluetooth este o tehnologie wireless, este utilizată pentru a transfera date între diferite dispozitive electronice. Distanța este foarte mică pentru a transfera datele între cele două dispozitive electronice. Această tehnologie nu necesită cabluri și adaptoare pentru a comunica cu alte dispozitive

2. Implementarea soluției

2.1. Concepte și tehnologii

Aplicația “MyPhoneIsLost” a fost dezvoltată pentru dispozitive mobile ce dispun de sistemul de operare Android, folosind Android Studio IDE ca mediu de dezvoltare și Java ca limbaj de programare.

Android Studio - este mediul de dezvoltare oficial pentru sistemul de operare Google Android, fiind dezvoltat peste software-ul IntelliJ IDEA al JetBrains și proiectat în mod special pentru dezvoltarea de aplicații Android. Am ales folosirea acestui mediu de dezvoltare datorită funcționalităților oferite față de alte medii de dezvoltare pentru platforma Android, printre care putem enumera:

- refactorizări specifice de cod și remedieri rapide de bug-uri
- instrumente specifice pentru scanarea codului în vederea verificării performanței, a gradului de utilizare și a diferitelor probleme legate de compatibilitate
- diferite șabloane special concepute pentru dezvoltarea de noi componente și funcționalități

BroadcastReceiver - este o componentă Android care vă permite să vă înregistrați pentru evenimente de sistem sau aplicație. Toți receptorii înregistrați pentru un eveniment sunt notificați de Android runtime odată ce acest eveniment se întâmplă.

Service - este o componentă a aplicației care poate efectua operațiuni pe termen lung și nu oferă o interfață de utilizator. O altă componentă a aplicației poate porni un serviciu și continuă să ruleze în fundal, chiar dacă utilizatorul trece la o altă aplicație. În plus, o componentă se poate lega de un serviciu pentru a interacționa cu acesta și chiar pentru a efectua comunicarea între procese (IPC).

Beacon - este un mic emițător radio Bluetooth, alimentat cu baterii. Beacon-urile ca și funcționalitate sunt similare cu un far. Aceste mici dispozitive hardware transmit continuu semnale Bluetooth Low Energy(BLE). Smartphone-urile cu Bluetooth-ul activat sunt capabile să scaneze și să afișeze aceste semnale sau să transmită aceste semnale (luând astfel chiar ele rolul de Beacon).

2.2. Analiza aplicației

Funcțiile sistemului și descriere ecranelor

Aplicația *MyPhoneIsLost* este structurată în următoarele ecrane pe care utilizatorul le poate accesa:

Ecranul de dispozitive – ecranul inițial al aplicației, în care utilizatorul va întâlni o listă de dispozitive care pot fi selectate și un buton roșu cu o lupă folosit pentru căutarea telefonului. Prima oară utilizatorul va trebui să acorde permisiunile trimitere, primire și citire SMS, dar și pentru accesare locație. Pentru a cere locația , utilizatorul va selecta unul din dispozitivele din listă și apoi va apăsa butonul de căutare, moment în care va primi o notificare că va trimite către dispozitivul selectat cerere pentru a primi coordonatele acestuia. Dacă utilizatorul va accepta apăsând butonul „OK” atunci mesajul de urgență va fi trimis și va primi răspuns de la telefonul pierdut în timp de câteva secunde, doar în cazul în care telefonul este deschis și locația este activată.

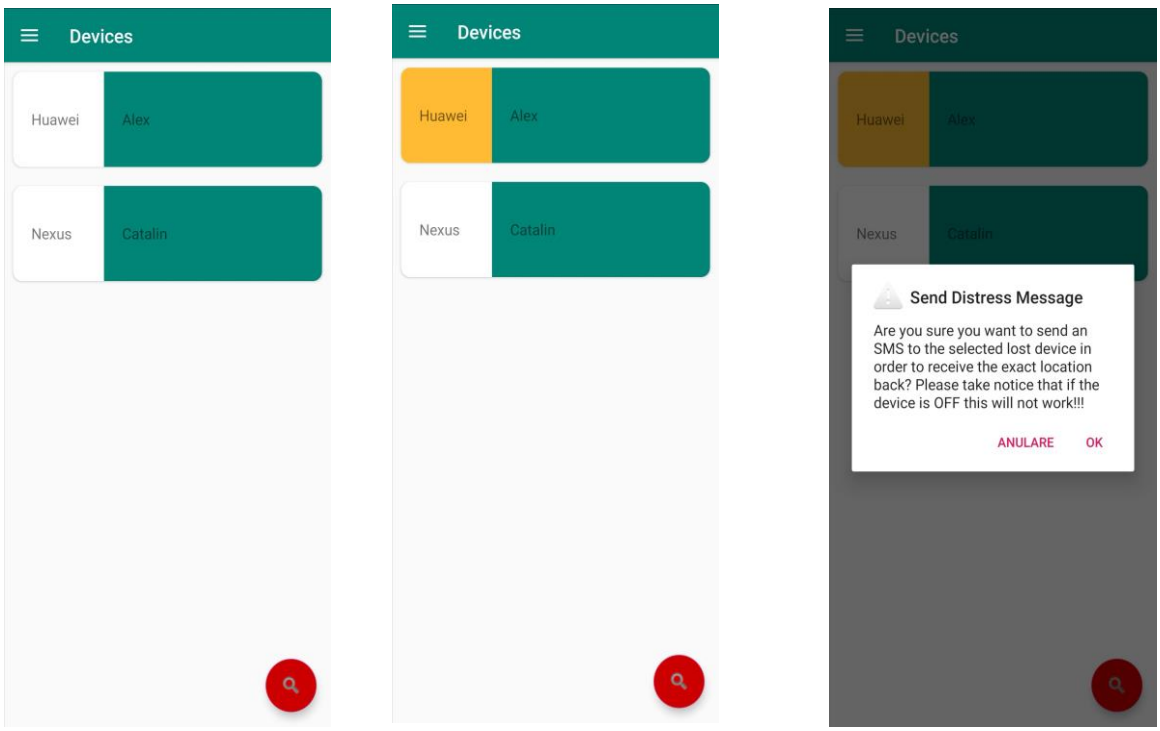

Fig. 1: Prezentare Ecran de dispozitive

Ecranul de setări – are cel mai important rol în aplicație, deoarece utilizatorul trebuie să acorde cinci permisiuni fără care aplicația nu poate funcționa. Acestea sunt: trimitere SMS, primire SMS, citire SMS, locația aproximativă și locația precisă. Inițial, toate butoanele vor fi roșii indicând că nici o permisiune nu este acordată. Fiecare buton care va fi apăsat se va colora în verde indicând că permisiunea este acordată. Al șaselea buton va acorda toate permisiunile necesare, ușurând astfel treaba utilizatorului de a apăsa cinci butoane pe rând.

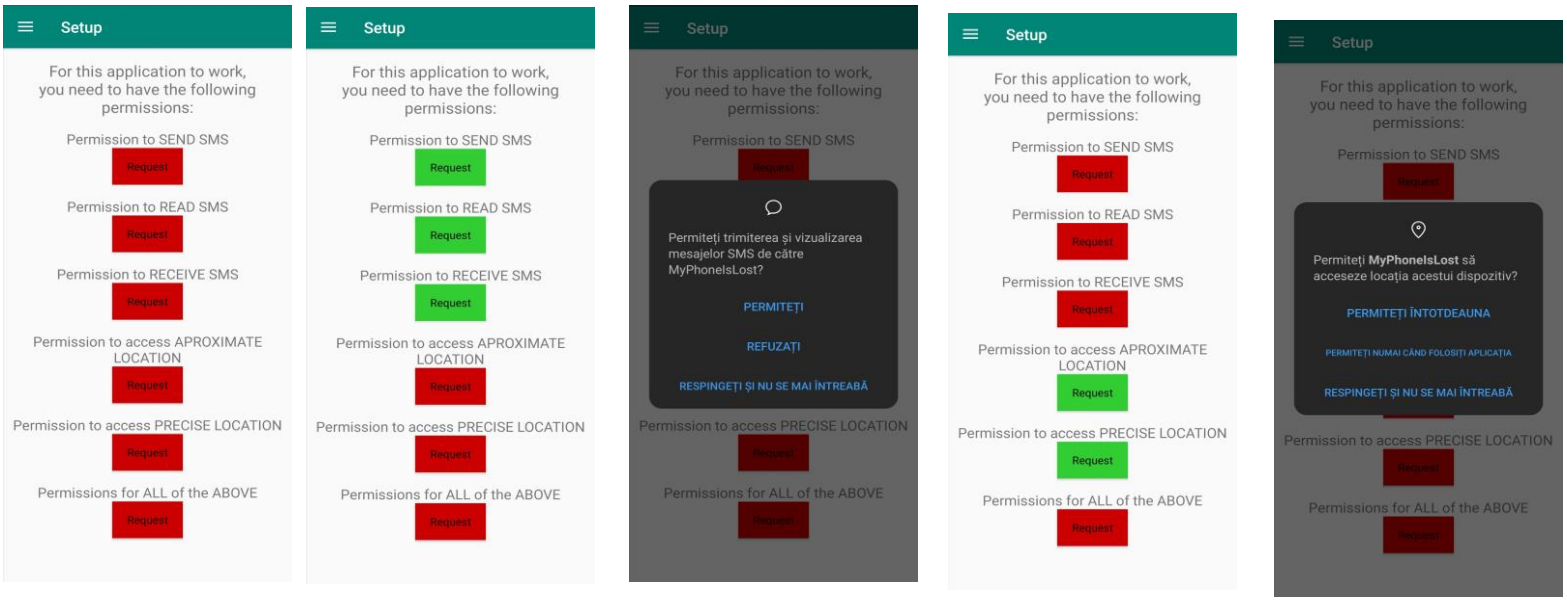

Fig. 2: Permitere acces la fiecare permisiune în parte

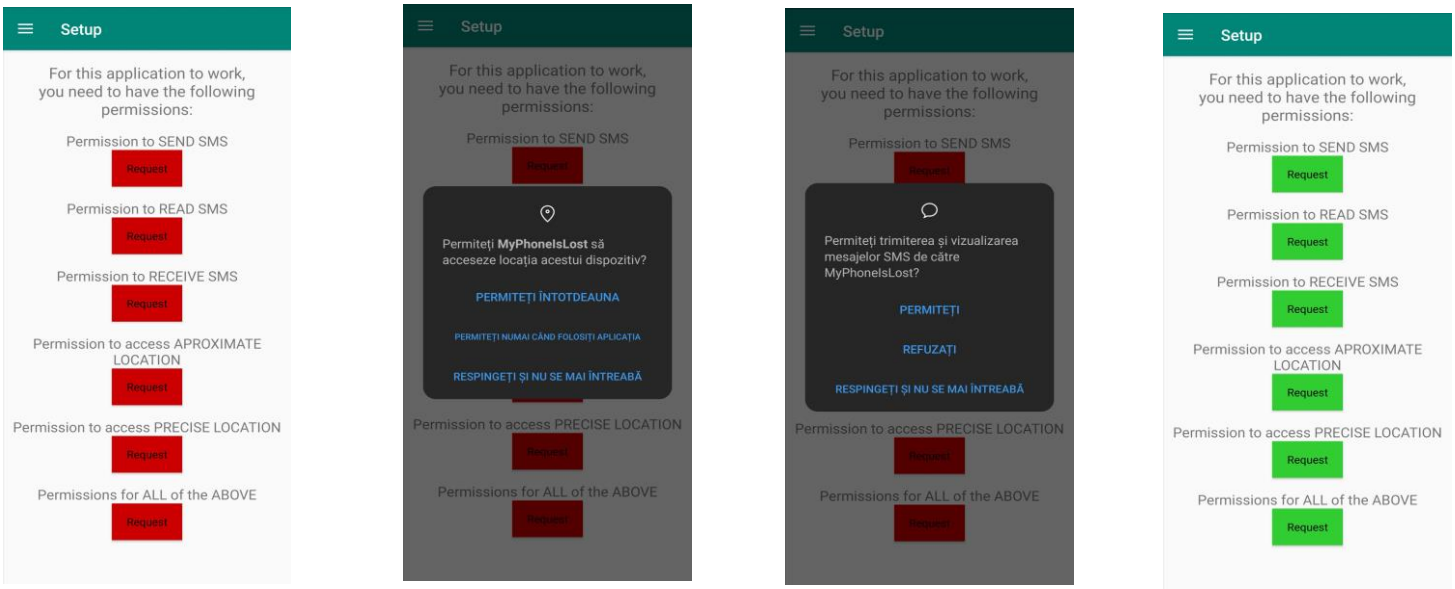

Fig. 3: Apăsarea butonului de acordare a tuturor permisiunilor

Ecran de căutare apropiată(proximitate) – are rolul de a permite utilizatorului să găsească un dispozitiv care este pierdut în casă sau într-o zonă apropiată de acesta. Pe ecran va apărea un text care va preciza distanța la care se află dispozitivul și o imagine cu trei culori diferite, roșu pentru cazul în care distanța la care se află de dispozitiv este cea mai mare, galben pentru o distanță medie, iar verde când utilizatorul este la o distanță mică față de dispozitivul pierdut.

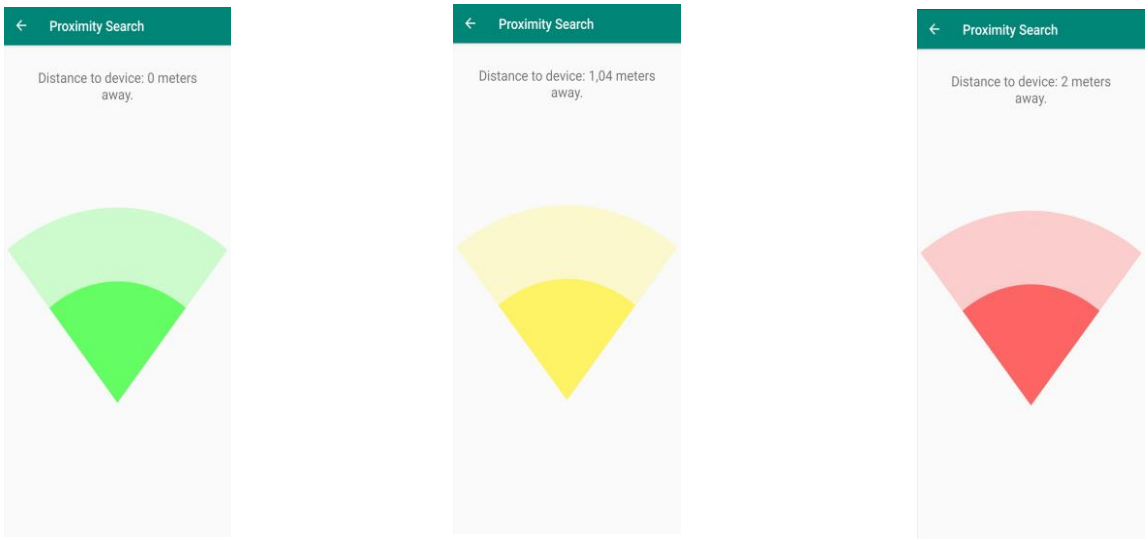

Fig. 4: Prezentare ecran de căutare apropiată(proximitate)

Notificare de activare GPS – are rolul de a obliga utilizatorul să aibă funcția GPS activă în permanență pentru cazul în care telefonul se pierde. În același timp are și rolul de a manipula un posibil hoț să creadă că dispozitivul este compromis în cazul în care nu se activează GPS-ul la dispozitiv. O notificare va apărea pe ecran care va anunța că telefonul nu este sigur, în momentul în care utilizatorul va apăsa pe notificare va fi directionat către setarea de activare locație.

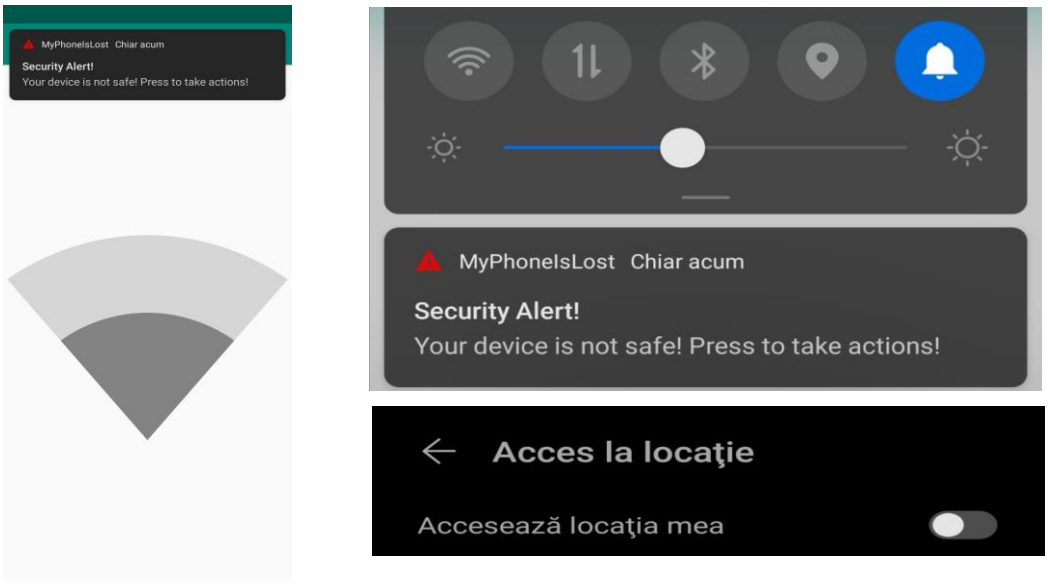

Fig. 5: Prezentarea apariției notificării pentru cazul în care locația dispozitvului este inactivă

Ecran de navigare - are rolul de a ghida utilizatorul către funcționalitățile aplicației. În antetul ecranului vom găsi o iconiță a aplicației, numele aplicației, numele utilizatorului și numărul dispozitivului selectat, iar în cazul în care niciun dispozitiv nu este selectat va fi afișat mesajul „Niciun dispozitiv nu este selectat”. Iar în corpul ecranului de navigare vom regăsi cele trei opțiuni, acestea fiind ecranul de selectare dispozitiv, ecranul de căutare apropiată(proximitate) și ecranul de setări.

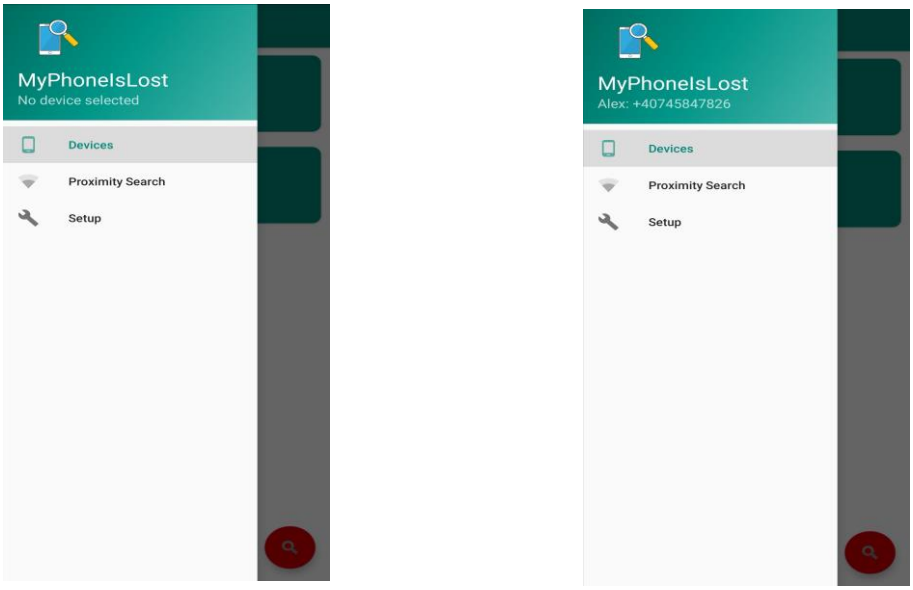

Fig. 6 : Prezentare Ecran de navigare

2.3. Arhitectura sistemului

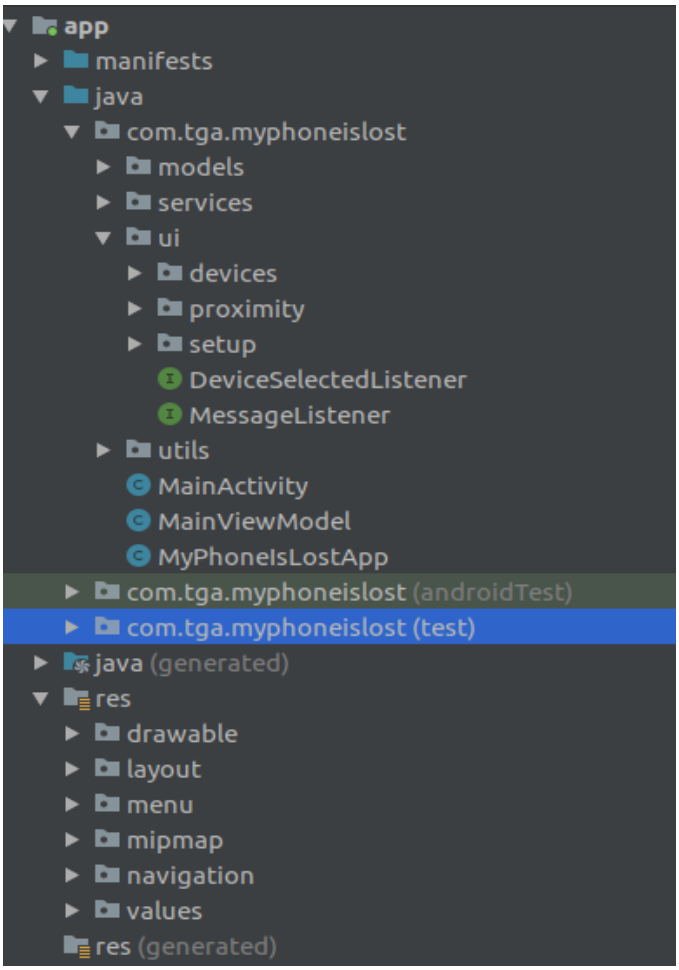

Fig. 7: Arhitectura aplicației „MyPhoneIsLost”

Aplicația este structurată în opt pachete ce conțin clase Java fiecare dintre ele îndeplinind un rol specific, fie legat de partea de logică a funcționalităților aplicației, manipularea datelor sau interacțiunea cu interfața grafică și unsprezece directoare specifice resurselor necesare aplicației.

Cele opt pachete ale aplicației conțin cincisprezece clase Java și trei ecrane, care se află în patru pachete de bază:

- *„models”* - conține toate modelele necesare aplicației reprezentând entități cu proprietăți și metode specifice, fiecare având asociat un anumit rol.
- *„services”* - conține clasele care folosesc serviciile necesare acțiunilor realizate de aplicație
- *„ui”* - conține alte trei pachete reprezentative cu conținut referitor la ecranele aplicației, în Android ele fiind reprezentate de Adapter sau Fragments, având un propriu ciclu de viață facilitând comunicarea cu componentele și diferitele funcționalități ale aplicației.
- *„utils”* - conține clasele care oferă metode de utilitate comună

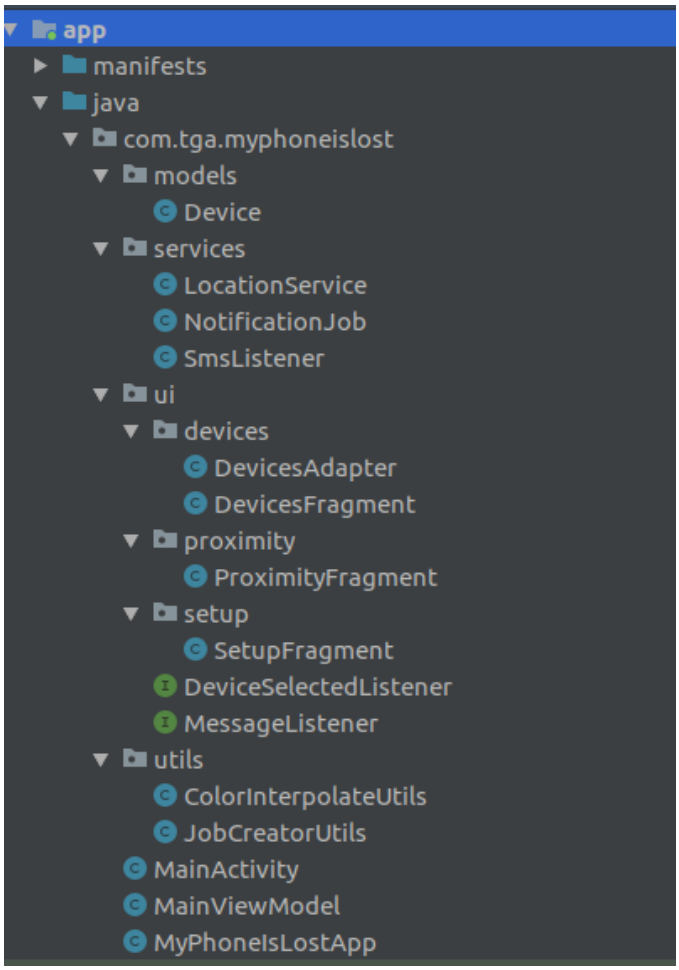

Fig. 8: Arhitectura pachetelor „models”, „services”, „ui” și „utils”

Ecranele aplicației au fost create folosind design pattern-urile MVVM(Model View ViewModel) și MVC (Model View Controller).

Design pattern-ul MVVM a fost creat pentru a simplifica programarea bazată pe evenimente a interfețelor de utilizator și este împărțit în trei componente importante:

- ***Model*** **–** reprezintă clasa care ține datele aplicatiei. Nu poate comunica direct cu View-ul. ViewModel-ul comunică cu Model-ul pentru a prelua și a salva date.
- ***View*** **–** reprezintă clasa ce îndeplinește rolul de interfață grafică, fiind responsabilă de manevrarea acțiunilor utilizatorilor către ViewModel.
- ***ViewModel*** **–** reprezintă clasa care are rol de mediator între Model și View. Este responsabil cu transformarea datelor din Model și oferă un flux de date către View.

În același timp aplicația folosește și design pattern-ul standard Android, Model View Controller, care presupune faptul că o componentă Activity sau Fragment va avea atât rol de Controller - fiind responsabilă cu dirijarea logicii aplicației, cât și rol de View, ocupându-se și de manevrarea interacțiunilor utilizatorilor cu interfața grafică a ecranelor. Ambele design pattern-uri aduc aplicației o organizare mai bună și îmbunătățesc arhitectura după care funcționează aceasta, impunând aplicației o separare mai bună a rolului pe care îl îndeplinește fiecare clasă.

După cum se poate observa în imaginile de mai sus, pachetul "ui" a fost organizat în 3 sub-pachete, ce conțin toate ecranele aplicației și funcționalitățile acesteia, după cum urmează:

- "***devices"*** - vom regăsi clase care vor inițializa lista de dispozitive pe care dorim să o căutăm, dar și butonul de trimitere a mesajului de căutare a telefonului pierdut care la apăsare va afisa mai întâi un dialog de alertă care să informeze utilizatorul că va trimite semnal către dispozitivul selectat.

- "***proximity"*** **-** vom regăsi clasa care inițiază căutarea de aproape a dispozitivul care va folosi un API de Beacon. Această librărie poate fi configurată ușor pentru a detecta diferite tipuri de Beacon, pentru a putea prelua actualizări ale distanței și poate fi folosit pentru a transmite semnal de Beacon. Clasa va apela metoda *"onBeaconServiceConnect()"* când serviciul de beacon rulează și este pregătit să accepte comanda prin *"BeaconManager".* Tot aici va fi inițializată și colorarea imaginii în funcție de distanța față de dispozitivul căutat.

```
public class ProximityFragment extends Fragment implements BeaconConsumer {
```

```
@Override
public void onBeaconServiceConnect() {
    RangeNotifier rangeNotifier = new RangeNotifier() {
        @Override
        public void didRangeBeaconsInRegion(Collection<Beacon> beacons, Region region) {
            if (beacons.size() > 0) {
                Beacon firstBeacon = beacons.iterator().next();
                setSignalTint(ColorInterpolateUtils.getColor((float) firstBeacon.getDistance()));
                DecimalFormat decimalFormat = new DecimalFormat( pattern: "#.##");
                setSignalRange(decimalFormat.format(firstBeacon.getDistance()));
            }
        }

    };
    try {
        beaconManager.startRangingBeaconsInRegion(new Region( uniqueId: "myRangingUniqueId", id1: null, id2: null, id3: null));
        beaconManager.addRangeNotifier(rangeNotifier);
    } catch (RemoteException e) {
        e.printStackTrace();
    }
}
```

Fig. 9: Inițializarea căutării de aproape(proximitate)

- ***"setup"*** - vom regăsi clasa care se ocupă cu verificarea permisiunilor de primire SMS, citire SMS, trimitere SMS, locație aproximativă și locație exactă care vor avea ca efect schimbarea culorii butoanelor de cerere acces din roșu în verde semnalizând că accesul a fost dat. Astfel va putea continua către ecranul de "D*evices" sau "Proximity search"*.

```
sendSmsPermissionRequest.setOnClickListener((v) → {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
        {
            requestPermissions(new String[]{
                    Manifest.permission.SEND_SMS}, requestCode: 10);
        }
});
readSmsPermissionRequest.setOnClickListener((v) → {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
        {
            requestPermissions(new String[]{
                    Manifest.permission.READ_SMS}, requestCode: 10);
        }
});
receiveSmsPermissionRequest.setOnClickListener((v) → {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
        {
            requestPermissions(new String[]{
                    Manifest.permission.RECEIVE_SMS}, requestCode: 10);
        }
});
coarseLocationPermissionRequest.setOnClickListener((v) → {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
        {
            requestPermissions(new String[]{
                    Manifest.permission.ACCESS_COARSE_LOCATION}, requestCode: 10);
        }
});
fineLocationPermissionRequest.setOnClickListener((v) → {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
        {
            requestPermissions(new String[]{
                    Manifest.permission.ACCESS_FINE_LOCATION}, requestCode: 10);
        }
});
allPermissionsRequest.setOnClickListener((v) → {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
        {
            requestPermissions(new String[]{
                    Manifest.permission.SEND_SMS,
                    Manifest.permission.RECEIVE_SMS,
                    Manifest.permission.READ_SMS,
                    Manifest.permission.ACCESS_COARSE_LOCATION,
                    Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.ACCESS_BACKGROUND_LOCATION}, requestCode: 10);
        }
});
```

Fig. 9 : Metoda de inițializare a butoanelor pentru acordarea permisiunilor

2.4. Implementarea funcționalităților

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    initViewModel();
    setContentView(R.layout.activity_main);
    setNavigationControls();
    SmsListener.bindListener(this);
    NotificationJob.scheduleJob();
}
```

Fig. 11: Pornirea aplicației „MyPhoneIsLost”

La pornirea aplicației vom regăsi după cum se vede în poza de mai sus, inițializate următoarele: clasa care se ocupă de persistența datelor (ViewModel-ul), meniul de navigare, serviciul de ascultare a SMS-ului(SmsListener) și serviciul de notificare.

```
public class SmsListener extends BroadcastReceiver {

    private static MessageListener messageListener;
    public static String SMS_LISTENER_ORIGINATING_ADDRESS = "smsListenerOriginatingAddress";

    @Override
    public void onReceive(Context context, Intent intent) {
        Bundle data = intent.getExtras();
        if (data != null) {
            if (data.get("pdus") != null) {
                Object[] pdus = (Object[]) data.get("pdus");
                SmsMessage smsMessage = null;
                if (pdus != null) {
                    for (Object pdu : pdus) {
                        smsMessage = SmsMessage.createFromPdu((byte[]) pdu);
                    }
                    if (smsMessage != null) {
                        String message = "Sender : " + smsMessage.getDisplayOriginatingAddress()
                                + "Message: " + smsMessage.getMessageBody();

                        if (messageListener != null) {
                            if (message.contains("LostMyPhone")) {
                                messageListener.messageReceived(message);
                            }
                        }

                        if (message.contains("LostMyPhone")) {
                            Intent intentService = new Intent(context, LocationService.class);
                            intentService.putExtra(SMS_LISTENER_ORIGINATING_ADDRESS, smsMessage.getDisplayOriginatingAddress());
                            context.startService(intentService);
                        }
                    }
                }
            }
        }
    }
```

Fig. 12: Clasa ce folosește serviciul de BroadcastReceiver

Clasa *SmsListiner* folosește serviciul de BroadcastReceiver. Acesta este o componentă Android specifică cu rol de a se înregistra notificări la producerea unor anumite evenimente și servește aplicației prin ascultarea unui mesaj specific pentru a putea transmite coordonatele telefonului pierdut. Dacă mesajul primit nu va fi cel așteptat de aplicație atunci aplicația va rămâne în așteptarea mesajului special destinat acesteia. În cazul în care mesajul este cel așteptat de aplicație atunci va fi apelat serviciul de cerere locatie(LocationListener).

```
public class LocationService extends Service implements LocationListener {

    @Nullable
    @Override
    public IBinder onBind(Intent intent) { return null; }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        if (intent.getStringExtra(SmsListener.SMS_LISTENER_ORIGINATING_ADDRESS) != null) {
            String originatingAddress = intent.getStringExtra(SmsListener.SMS_LISTENER_ORIGINATING_ADDRESS);
            getLastLocationNewMethod(originatingAddress);
        }
        stopSelf();
        return START_NOT_STICKY;
    }
```

Fig. 13: Clasa ce folosește serviciul LocationListener

Clasa *LocationService* folosește serviciul LocationListener. Acesta este o componentă Android specifică cu rol de a primi notificări când locația dispozitivului se schimbă și servește aplicației prin ascultarea locației în momentul când mesajul de alertă este primit, trimițând coordonatele dispozitivul pentru a putea fi trimise înapoi persoanei care este în căutarea telefonului. În această clasă realizându-se și trimiterea înapoi către expeditor a mesajului cu coordonatele adăugate într-un link de Google Maps.

```java
public class NotificationJob extends Job {

    public static final String TAG = "GPS_not_active_notification_tag";

    @NonNull
    @Override
    protected Result onRunJob(@NonNull Params params) {

        LocationManager lm = (LocationManager) getContext().getSystemService(Context.LOCATION_SERVICE);
        boolean gps_enabled = false;
        try {
            gps_enabled = lm.isProviderEnabled(LocationManager.GPS_PROVIDER);
        } catch (Exception ex) {
            ex.printStackTrace();
        }

        if (!gps_enabled) {
            Intent intent = new Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            PendingIntent pendingIntent = PendingIntent.getActivity(getContext(), requestCode: 0, intent, flags: 0);

            NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(getContext())
                    .setContentTitle("Security Alert!")
                    .setContentText("Your device is not safe! Press to take actions!")
                    .setContentIntent(pendingIntent)
                    .setAutoCancel(true)
                    .setSmallIcon(R.drawable.ic_warning)
                    .setShowWhen(true)
                    .setColor(Color.RED)
                    .setLocalOnly(true);

            NotificationManagerCompat notificationManager = NotificationManagerCompat.from(getContext());


            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                String channelId = "Your_channel_id";
                NotificationChannel channel = new NotificationChannel(
                        channelId,
                        name: "Channel human readable title",
                        NotificationManager.IMPORTANCE_HIGH);
                notificationManager.createNotificationChannel(channel);
                notificationBuilder.setChannelId(channelId);
            }

            Notification notification = notificationBuilder.build();
            notificationManager.notify(new Random().nextInt(), notification);
        }
        return Result.SUCCESS;
    }
```

Fig. 14: Clasa ce folosește serviciul Job

Clasa *NotificationJob* folosește serviciul de Job. Acesta are rolul de a rula un thread în fundal și folosește aplicației pentru a verifica dacă funcția GPS a telefonului este activată, în cazul în care nu este activ pe ecranul telefonului va apărea o notificare, care dacă este accesată îl va direcționa pe utilizator către setarea telefonului de activare locație. În cazul în care este activat notificarea nu va apărea.

3. Concluzii

Scopul lucrării *„MyPhoneIsLost”* este să arate o posibilă soluție pentru conceperea unei aplicații capabile să ajute o persoană să își găsească telefonul furat sau pierdut. Acest lucru fiind posibil folosind doar o cartelă SIM și funcția GPS, fără a avea nevoie de acces la internet pentru a putea comunica cu dispozitivul. La dezvoltarea aplicației am întâmpinat probleme de natură software precum comunicarea între servicii, dar cea mai importantă fiind permisiunea de activare forțată de la distanță a GPS-ului, fapt datorat unor limitări a sistemului de operare Android.

Per ansamblu, aplicația își îndeplinește toate cerințele impuse. Aplicația este capabilă să trimită coordonatele exacte ale telefonului pierdut către expeditorul care a trimis mesajul de urgență, iar în cazul în care este pierdut în apropierea utilizatorului, aplicația afișează distanța la care se află telefonul amplasat într-o zonă din apropierea acestuia.

ÎMBUNĂTĂȚIRI

Pentru viitor, aplicația poate fi îmbunătățită pe plan software prin crearea unei baze de date cu utilizatori, prin care să poată adăuga mai multe dispozitive și astfel s-ar putea crea o funcționalitate în plus prin care să se poată căuta dispozitivul prin metoda căutarii cu ajutorul internetului. O altă îmbunătățire ar fi găsirea unei modalități de a activa de la distanță funcția GPS al telefonul pierdut atunci când primește mesajul de găsire a telefonului. O nouă îmbunătățire ar putea fi criptarea mesajului care este trimis de expeditor pentru a nu permite destinatarului să înțeleagă semnificația acestuia. O altă îmbunătățire ar putea fi ascunderea notificării de primire mesaj pe telefonul pierdut în cazul în care aplicația trimite mesaj de căutare.

Pentru căutarea de aproape a dispozitivul o îmbunătățire ar putea fi găsirea unui algoritm mai bun de căutare astfel încât precizia să fie cât mai exactă. O nouă îmbunătățire ar putea fi crearea unei animații interactive prin care barele de distanță să apară sau să dispară în funcție de distanța la care se află dispozitivul pierdut.

Bibliografie

1. **Documentație oficială Android:**
   https://developer.android.com/docs/
2. **Beacon Proximity Detection API:**
   https://altbeacon.github.io/android-beacon-library/
3. **Navigation Component:**
   https://developer.android.com/guide/navigation/navigation-getting-started
4. **Broadcast Receiver (for listening on SMS's):**
5. https://developer.android.com/reference/android/content/BroadcastReceiver
6. **Location Service(for sendint the current location):**
   https://developer.android.com/guide/components/services
7. **Background Job Scheduler (for sending background notification over location enabled/disabled):**
   https://github.com/evernote/android-job#workmanager
8. **Model-View-ViewModel (android Pattern):**
   https://developer.android.com/jetpack/docs/guide