

Projet : Inteco

Programmation mobile

Mohamad Satea Almallouhi - Tony Nguyen
M1 Génie Logiciel
Faculté des Sciences
Université de Montpellier.

31 mai 2024



Table des matières

Introduction	2
1 Les fonctionnalités présentes	3
1.1 L'inscription	3
1.2 La connection	5
1.3 Changement de mot de passe	6
1.4 Barre de navigation	6
1.5 Sauvegarder	7
1.6 Postuler	9
1.7 Rechercher	10

Introduction

Dans le cadre de l'Unité d'Enseignement Programmation Mobile, nous allons vous parler de notre avancé par rapport à l'application d'intérim.

Par manque de temps, beaucoup de fonctionnalités nous manquent

Démonstration vidéo

En ligne sur Youtube, à l'adresse URL <https://youtu.be/AspmuYZcom0> une démonstration vidéo de notre travail.

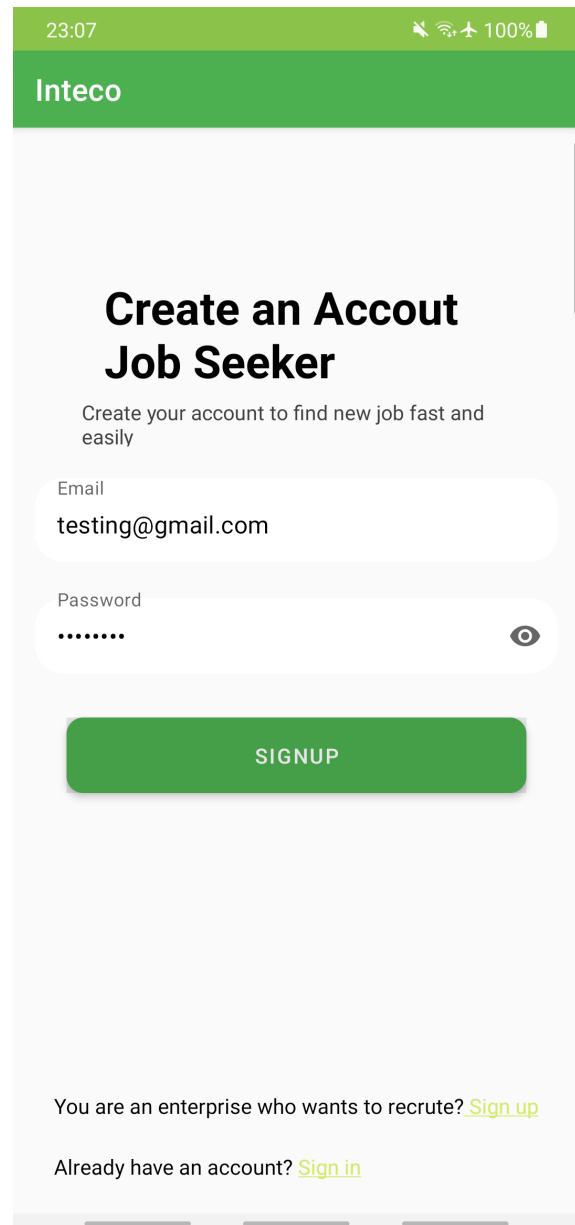
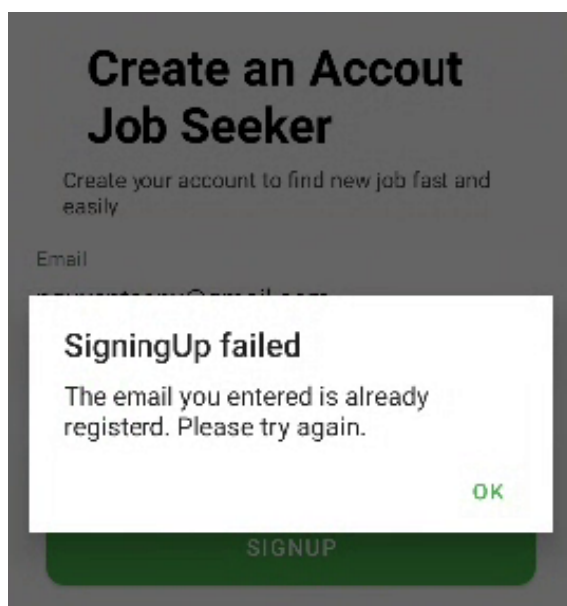
En ligne sur github, notre dépôt à l'adresse suivante : <https://github.com/tony-nguyen1/Inteco/>

1 Les fonctionnalités présentes

1.1 L'inscription

En tant qu'entreprise ou interimaire, vous pouvez créer un compte.

Elle distingue des pages d'inscription dédiée a utilisateur type entreprise ou employé. L'application vous demande petit à petit les informations et vous signale les problèmes éventuels.



23:08

100%

Inteco

Enter your information (1/3)

First name*

tes

Last name *

ting

Sex

Other

Birthday

12

NEXT

23:08

100%

Inteco

Enter your information (2/3)

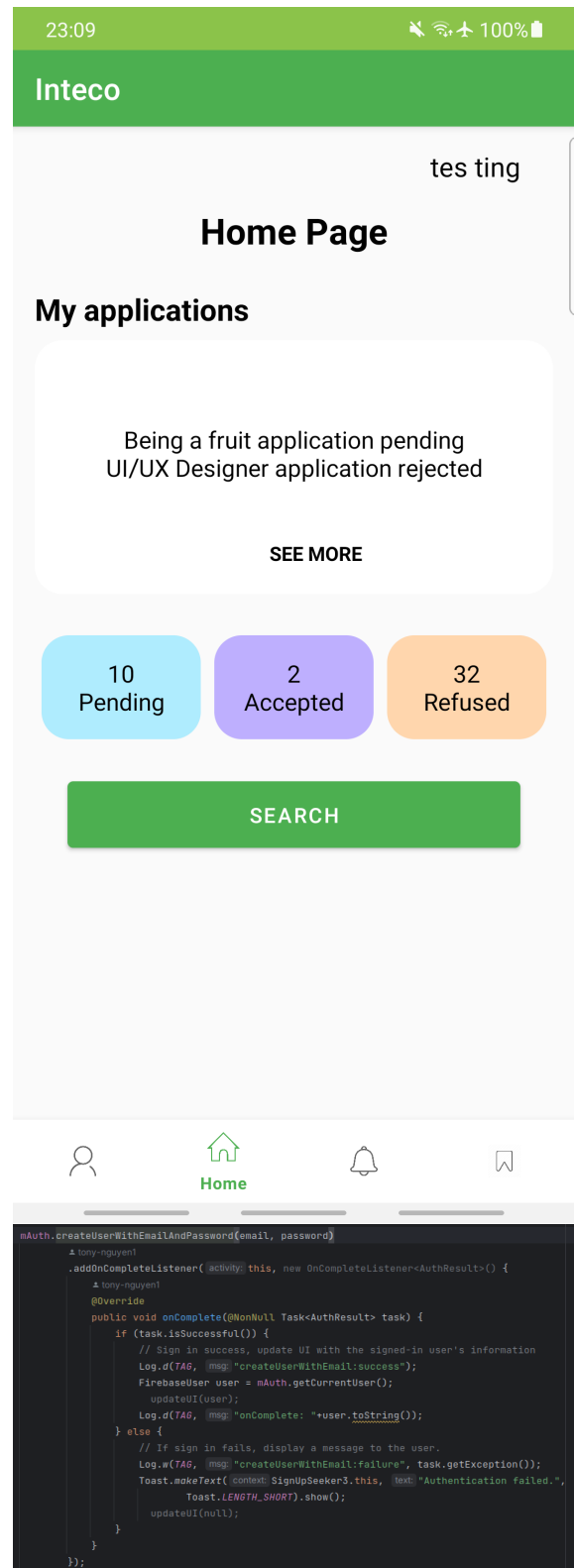
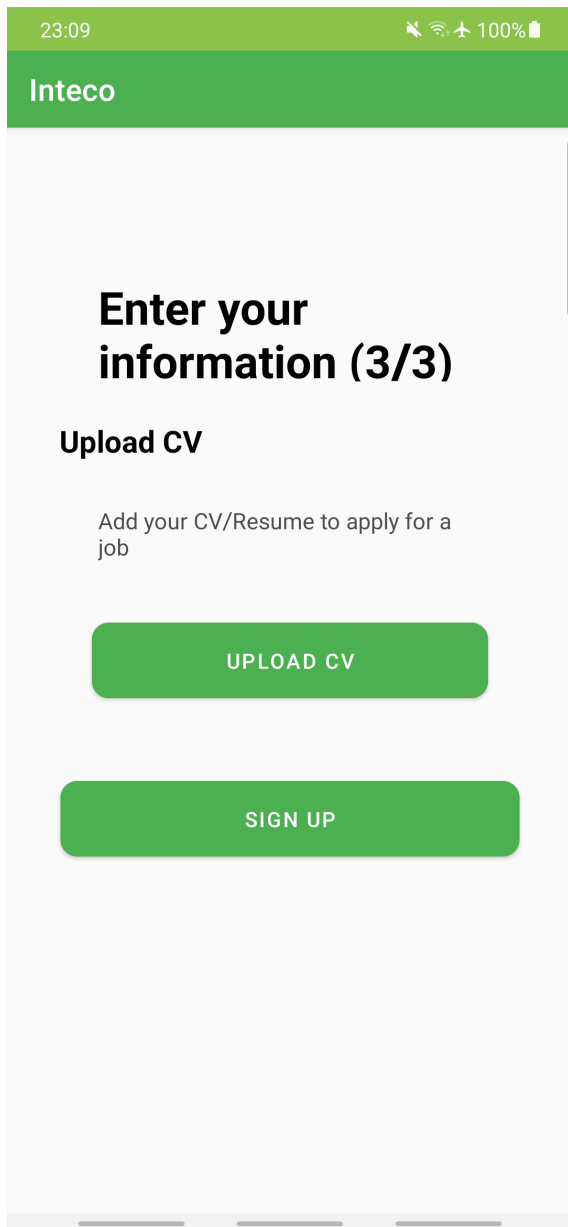
Phone number

City name

Address

Nationality

NEXT



1.2 La connection

Les utilisateurs peuvent se connecter du même endroit, qu'ils soient entreprise ou chomeurs.

```

mAuth.signInWithEmailAndPassword(email, password)

private tony-nguyen1
.addOnCompleteListener(activity, this, new OnCompleteListener<AuthResult>() {
    private tony-nguyen1
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful()) {
            // Sign in success, update UI with the signed-in user's information
            Log.d(TAG, "signInWithEmail:success");
            FirebaseUser user = mAuth.getCurrentUser();
            updateUser(user);
            FirebaseAuth.postValue(user);
        } else {
            // If sign in fails, display a message to the user.
            Log.w(TAG, "signInWithEmail:failure", task.getException());
            Toast.makeText(context, LoginFirstActivity.this, text "Authentication failed.",
                Toast.LENGTH_SHORT).show();
            updateUser(null);
            response.postValue(null);
        }
    }
});

```

1.3 Changement de mot de passe

Les utilisateurs peuvent changer leur mot de passe. En effet, ils ont la possibilité de recevoir un mail pour faire ça grace au service Authentication de Firebase.

```

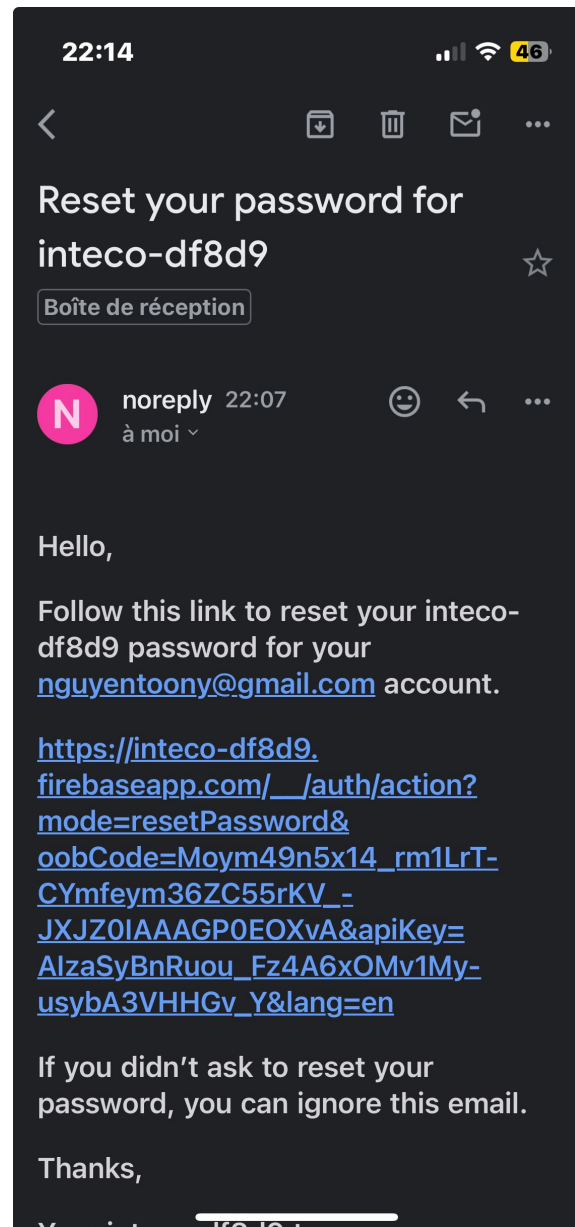
MutableLiveData<Boolean> listen = new MutableLiveData<>();
Helper.verifyEmailExists(email, collection: "allUsers", listen);

private tony-nguyen1+1*
listen.observe(owner: ResetPassword.this, new Observer<Boolean>() {
    private tony-nguyen1+1*
    @Override
    public void onChanged(Boolean aBoolean) {
        if (aBoolean) { // true
            private tony-nguyen1*
            Helper.resetPassword(email, new OnCompleteListener() {
                private tony-nguyen1*
                @Override
                public void onComplete(@NonNull Task task) {
                    if (task.isSuccessful()) {
                        Log.d(TAG, "msg: Email sent.");
                        Toast.makeText(context, ResetPassword.this, text "Email sent",
                            Toast.LENGTH_SHORT).show();
                    }
                }
            });
        } else { // false
            showIncorrectEmailAlert(option: true);
        }
    }
});

public static void resetPassword(String email, OnCompleteListener onCompleteListener) {
    Log.d(TAG, "msg: resetPassword: "+email);
    FirebaseAuth auth = FirebaseAuth.getInstance();
    String emailAddress = email;

    auth.sendPasswordResetEmail(emailAddress)
        .addOnCompleteListener(onCompleteListener);
}

```



1.4 Barre de navigation

La page principale après la connexion est une activité. La navigation vous fait passer d'un fragment à l'autre.

```

/***** Navigation Bar *****/
BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);
SateasMail
bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        Fragment fragment = null;

        int itemId = item.getItemId();
        if (itemId == R.id.nav_settings) {
            fragment = SettingsSeeker.newInstance(email, firstname, lastname);
        } else if (itemId == R.id.nav_home) {
            fragment = HomeSeeker.newInstance(email, firstname, lastname);
        } else if (itemId == R.id.nav_notifications) {
            fragment = NotificationsSeeker.newInstance(email, firstname, lastname);
        } else if (itemId == R.id.nav_saved) {
            fragment = SavedSeeker.newInstance(email, firstname, lastname);
        }
        getSupportFragmentManager().beginTransaction()
            .replace(R.id.fragment_container, fragment)
            .commit();
        return true;
    }
});

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:background="#FAFAFA"
    tools:context=".Seeker.HomePageSeeker"
    android:id="@+id/homeJobSeeker">

    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:background="#FFFFFF"
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
        android:layout_height="160px"
        android:layout_alignParentBottom="true"
        app:menu="@menu/bottom_nav_menu_seeker" />
</RelativeLayout>

```

23:13

Inteco

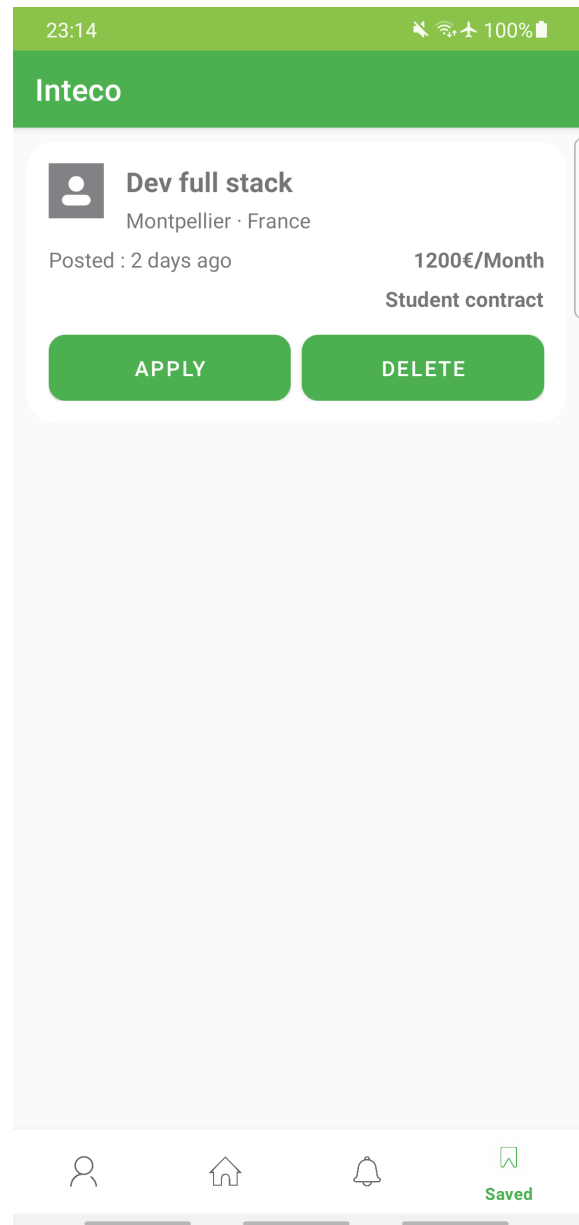
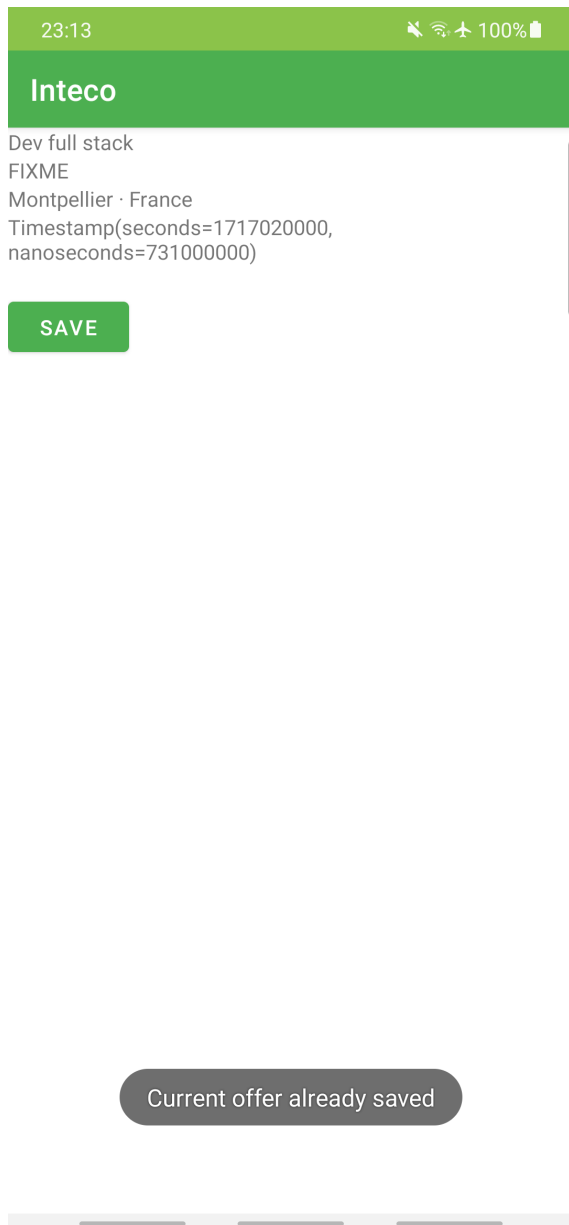
Dev full stack
FIXME
Montpellier · France
Timestamp(seconds=1717020000,
nanoseconds=731000000)

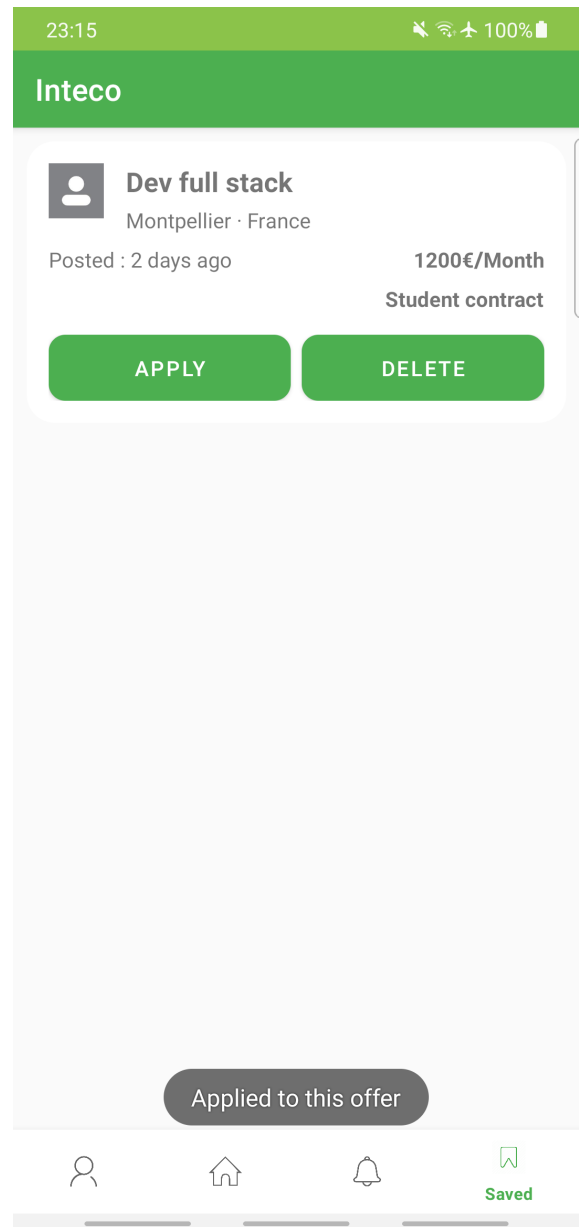
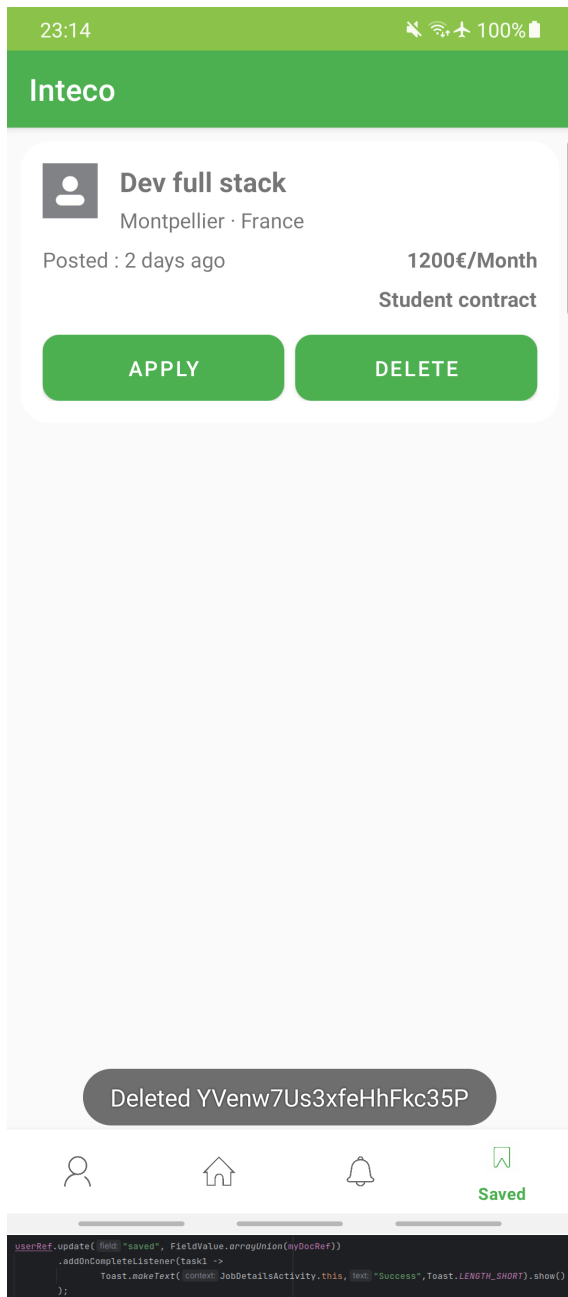
SAVE

Success

1.5 Sauvegarder

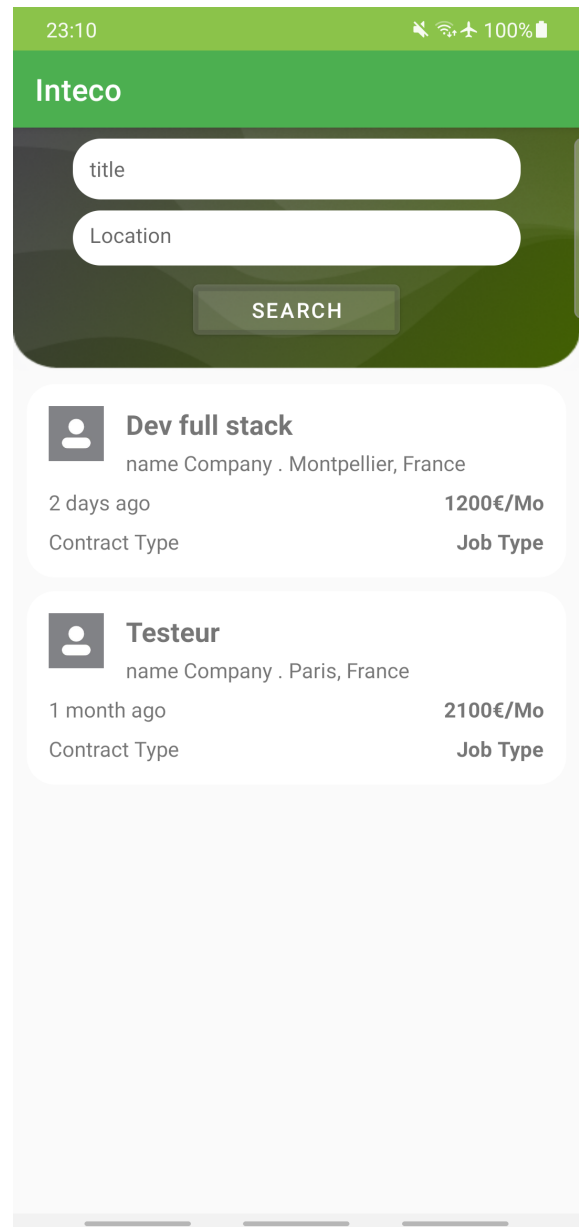
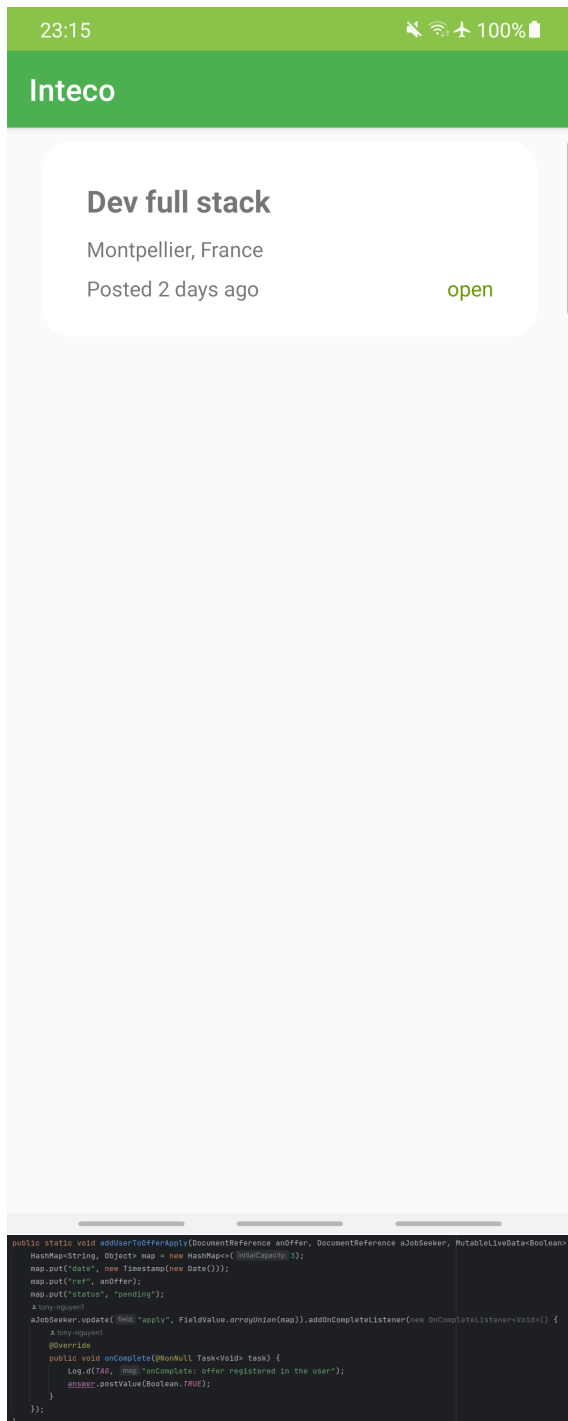
Si jamais, un chercheur de travail souhaite revenir à une offre d'emplois, il a la possibilité de l'enregistrer.





1.6 Postuler

Un chercheur d'emplois peut postuler a une offre.



1.7 Rechercher

Il peut rechercher parmi les offres selon des mots clés dans le titre du travail et la ville.

```

public void searchForJob(List<String> keywordsTitle, List<String> keywordsLocation, MutableLiv
Log.d(TAG, msg: "searchForJob: keywordsTitle="+keywordsTitle+" keywordsLocation="+keywo
// tony-nguyen1
db.collection( collectionPath: "offers").get().addOnCompleteListener(new OnCompleteListener<Que
// tony-nguyen1
@Override
public void onComplete(@NonNull Task<QuerySnapshot> task) {
    ArrayList<QueryDocumentSnapshot> matchingDocumentsArrayList = new ArrayList<>();
    if (task.isSuccessful()) {
        for (QueryDocumentSnapshot document : task.getResult()) {
            boolean thisDocumentIsSelected = false;
            String jobTitle = document.get( field: "jobTitle", String.class);
            String location = document.getString( field: "place");

            String[] tab = jobTitle.toLowerCase().split( regex: "[ ]");
            List<String> l = Arrays.asList(tab);
            for (String keyword : keywordsTitle) {
                if (l.contains(keyword)) {
                    Log.d(TAG, "onComplete: keywords="+keyword);
                    thisDocumentIsSelected = true;
                }
            }

            tab = location.toLowerCase().split( regex: "[ ]");
            l = Arrays.asList(tab);
            Log.d(TAG, msg: "onComplete: searching in"+l);
            Log.d(TAG, msg: "onComplete: keywordsLocation="+keywordsLocation);
            for (String keyword : keywordsLocation) {
                if (l.contains(keyword)) {
                    Log.d(TAG, "onComplete: keywords="+keyword);
                    thisDocumentIsSelected = true;
                }
            }

            if (thisDocumentIsSelected) {
                matchingDocumentsArrayList.add(document);
                Log.d(TAG, msg: "onComplete: match");
            }
        }
        Log.d(TAG, msg: "onComplete: matched :"+matchingDocumentsArrayList);
        response.postValue(matchingDocumentsArrayList);
    }
}

```