Inteco application mobile
Agence d'intérim
Rapport final

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

+31 mai 2024

Table des matières

Note Préliminaire

Nous avons initialement soumis notre travail à la date prévue de remise. Cependant, en raison de contraintes de temps importantes, notamment de nombreux examens et projets, nous n'avons pas pu finaliser toutes les fonctionnalités et les détails souhaités. Cette version du rapport, soumise un jour après la date de remise, représente une révision améliorée et plus complète de notre projet. Nous espérons que cette version reflète mieux notre vision et nos efforts déployés pour aboutir à une solution efficace et fonctionnelle.

Démonstration vidéo

En ligne sur Youtube, à l'adresse URL `https://youtu.be/t7TYONoK8DU` une démonstration vidéo de notre travail.

Lien Github

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

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. L'application Inteco conçue dans le cadre de nos études de master, vise à révolutionner le processus de recherche d'emploi et de recrutement. Inteco permet aux utilisateurs de s'inscrire et de se connecter en tant que chercheurs d'emploi ou entreprises, offrant ainsi une plateforme multifonctionnelle adaptée aux besoins spécifiques de chaque groupe.

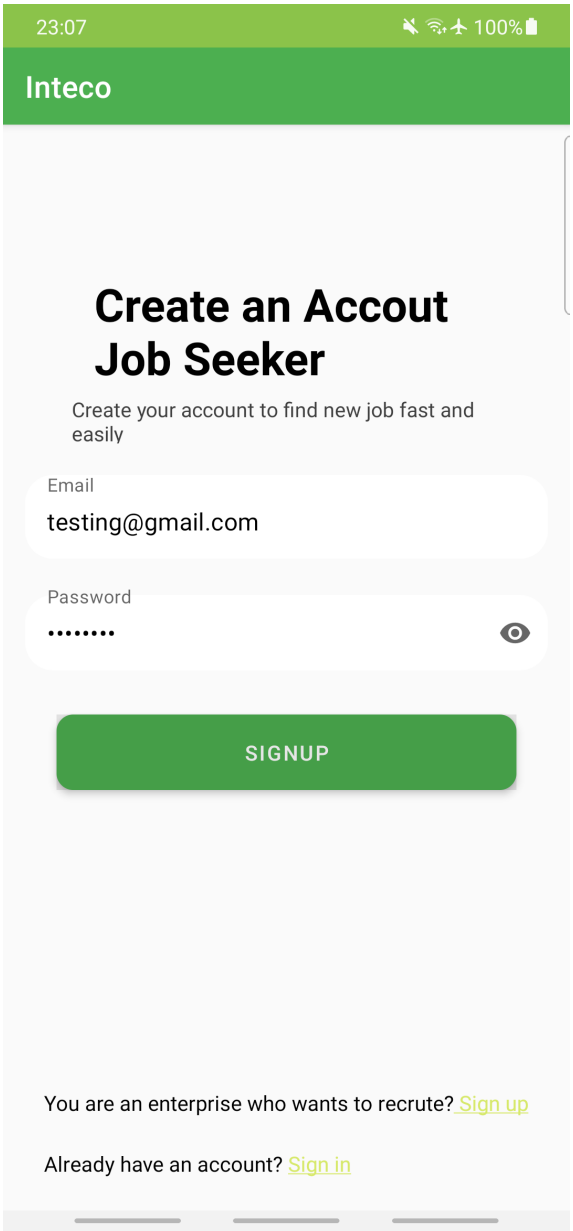
1 Les fonctionnalités présentes

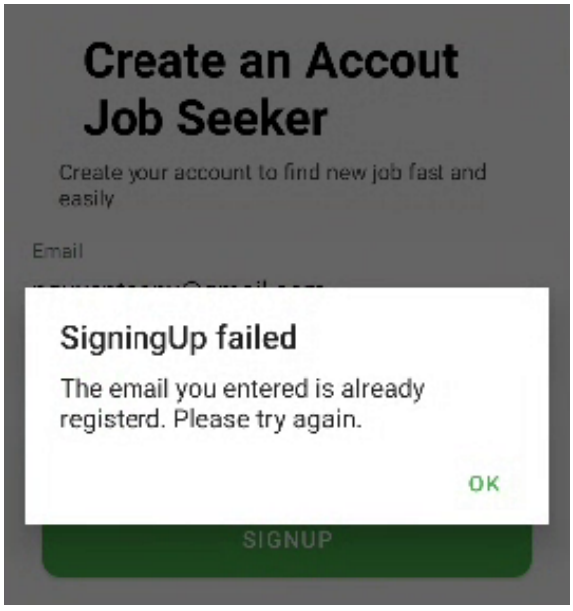
1.1 L'inscription

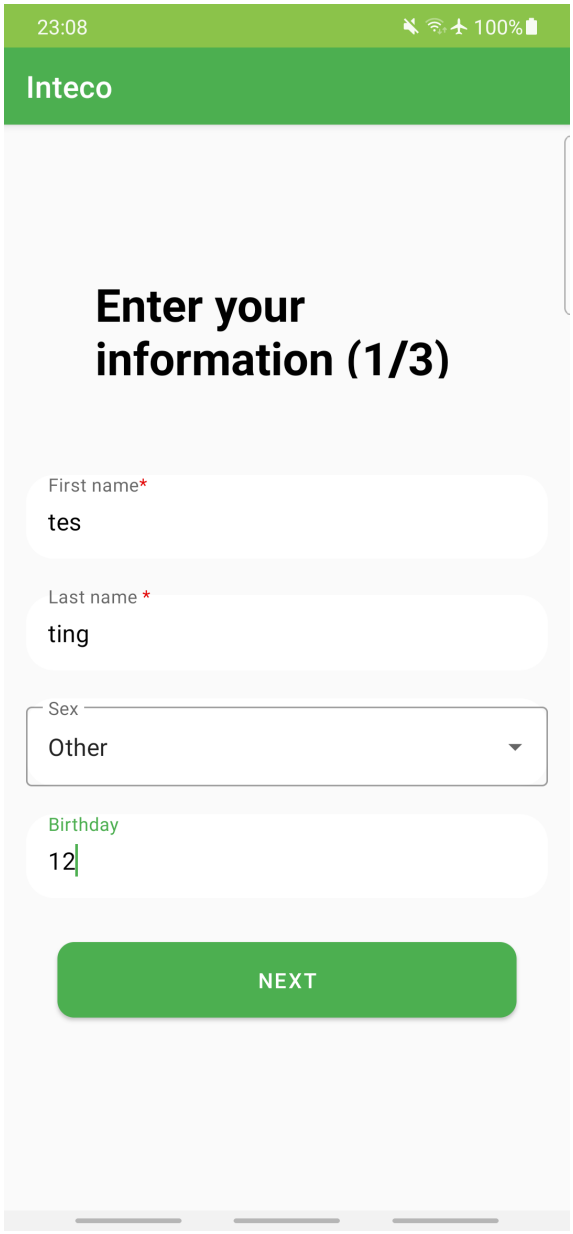
Le processus d'inscription sur l'application Inteco est conçu pour garantir la complétude et la précision des informations fournies par les utilisateurs. Chaque utilisateur, qu'il soit chercheur d'emploi ou entreprise, doit remplir tous les champs obligatoires du formulaire d'inscription, empêchant ainsi la création de comptes incomplets. L'authentification et la gestion des comptes via Firebase ajoutent une couche de sécurité et de fonctionnalité, permettant notamment la réinitialisation des mots de passe. Pour renforcer l'interface moderne et conviviale, nous avons utilisé plusieurs pages pour saisir les informations lors de l'inscription, facilitant ainsi la complétion des données et apportant une touche de modernité à notre application.

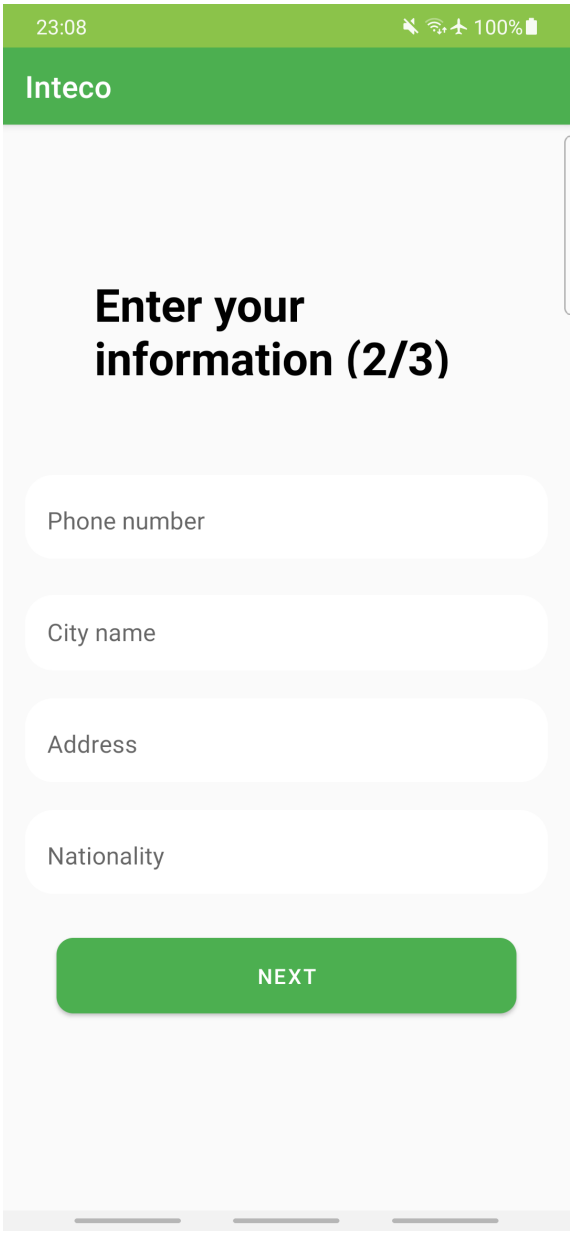
En tant qu'entreprise ou interimaire, vous pouvez créer un compte.

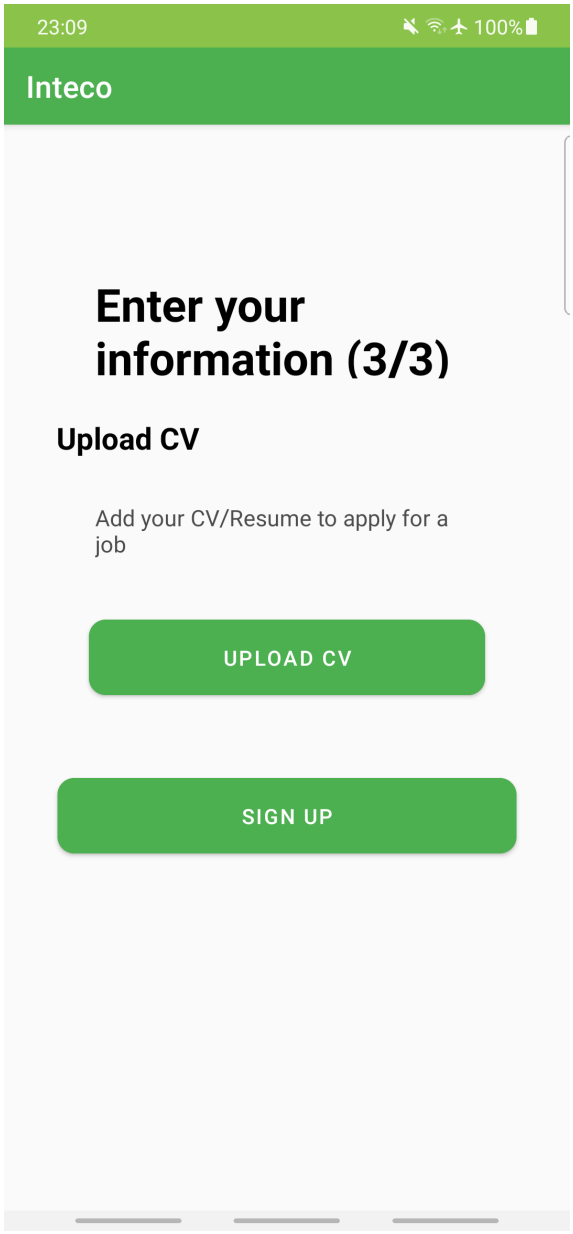
Elle distingue des pages d'inscription dédiée a utilisateur type entreprise ou employé. L'application vous demande petit à petit les informations et vous signale les problèmes éventuels.

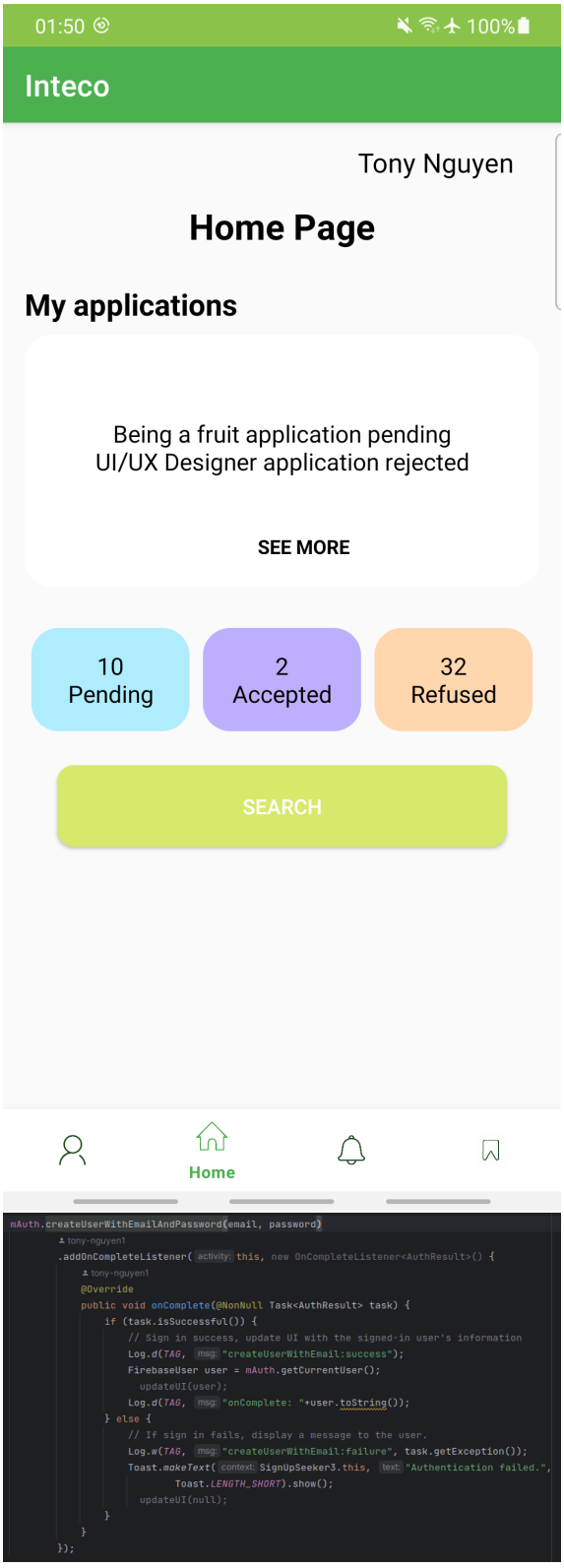





23:08
100%
Inteco
Enter your information (1/3)
First name*
tes
Last name *
ting
Sex
Other
Birthday
12
NEXT


23:08
100%
Inteco
Enter your information (2/3)
Phone number
City name
Address
Nationality
NEXT






```
mAuth.createUserWithEmailAndPassword(email, password)
        tony-nguyen1
        .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
            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, msg: "createUserWithEmail:success");
                    FirebaseUser user = mAuth.getCurrentUser();
                      updateUI(user);
                    Log.d(TAG, msg: "onComplete: "+user.toString());
                } else {
                    // If sign in fails, display a message to the user.
                    Log.w(TAG, msg: "createUserWithEmail:failure", task.getException());
                    Toast.makeText( context: SignUpSeeker3.this, text: "Authentication failed.",
                            Toast.LENGTH_SHORT).show();
                      updateUI(null);
                }
            }
        });
```

1.2 La connection

Pour faciliter l'authentification, les utilisateurs peuvent se connecter au même endroit, qu'ils soient entreprises ou chercheurs d'emploi. Cette

approche centralisée simplifie l'accès à la plateforme et améliore l'expérience utilisateur.

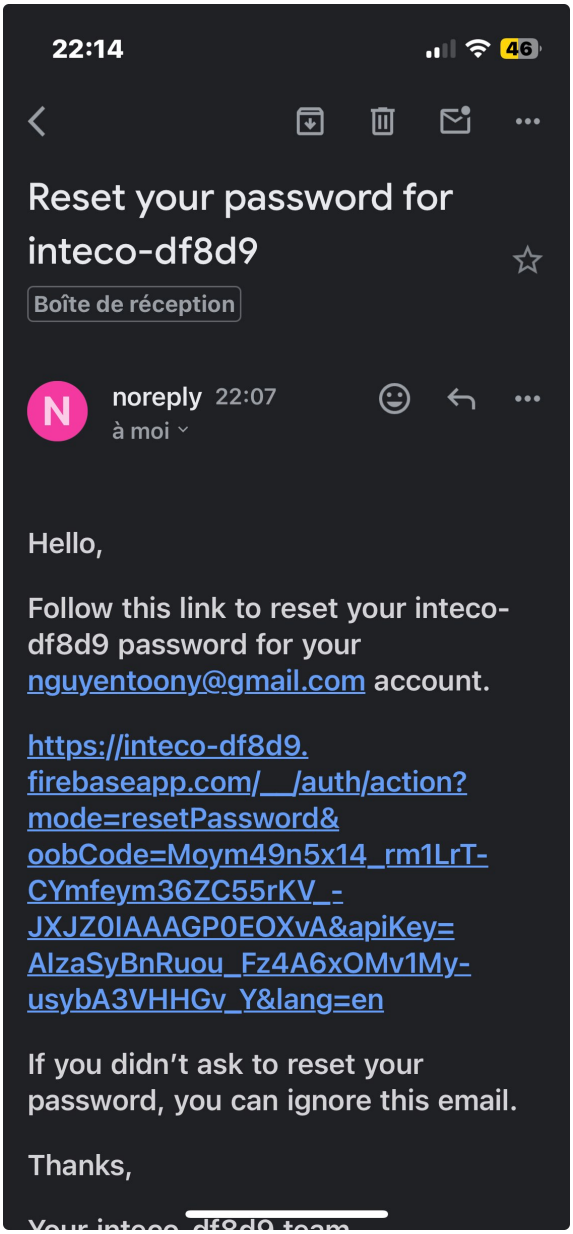
```
mAuth.signInWithEmailAndPassword(email, password)
        tony-nguyen1
        .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
            tony-nguyen1
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful())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

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

tony-nguyen1 +1 *
listen.observe( owner: ResetPassword.this, new Observer<Boolean>() {
    tony-nguyen1 +1 *
    @Override
    public void onChanged(Boolean aBoolean) {
        if (aBoolean) { // true
            tony-nguyen1 *
            Helper.resetPassword(email, new OnCompleteListener() {
                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",
                    }
                }
            });
        } 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);
}
```

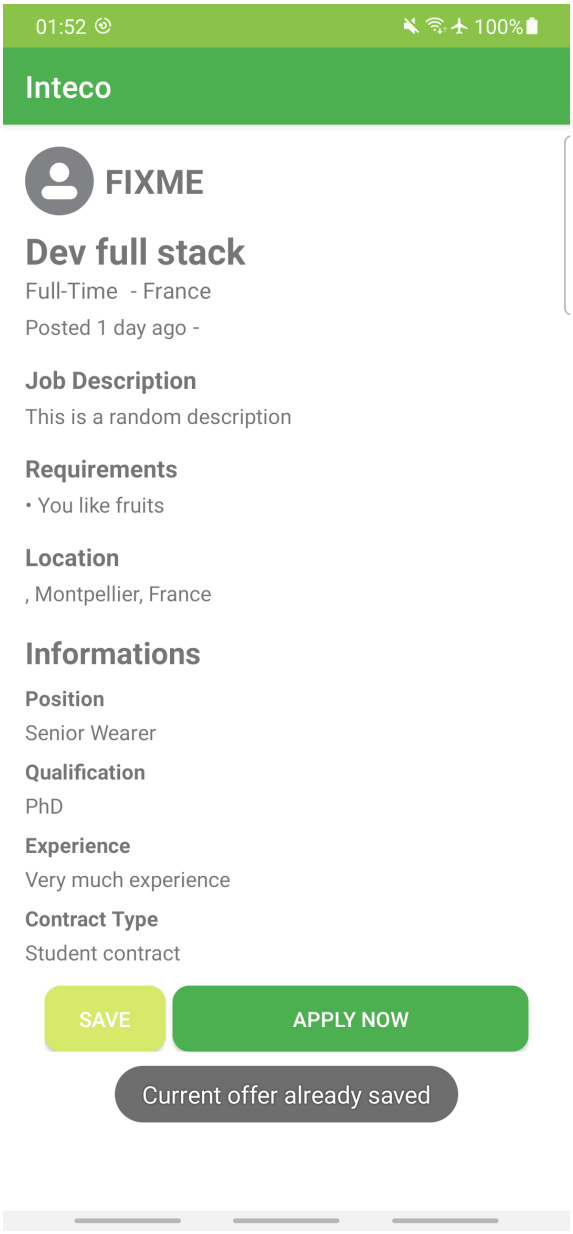


1.4 Barre de navigation

Nous avons implémenté une barre de navigation réactive et adaptée aux différents types d'utilisateurs (entreprises et chercheurs d'emploi). Cette barre de navigation est intégrée dans une activité principale (home), où chaque section est représentée par un fragment. Cette approche garantit une fluidité de changement de section au sein de l'application, améliorant ainsi l'expérience utilisateur. Les utilisateurs peuvent facilement accéder aux différentes fonctionnalités et informations pertinentes sans interruption, ce qui renforce l'ergonomie et la modernité de l'application.

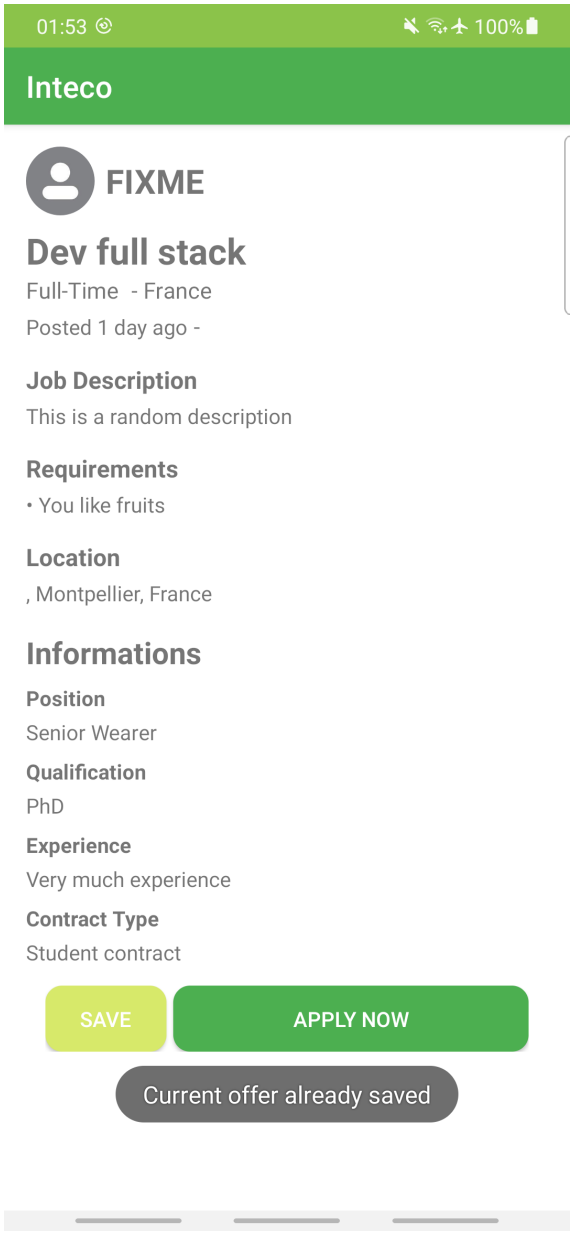
```
/********** Navigation Bar **********/

BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);
 SateaMall
bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnN

    1 usage   SateaMall
    @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;
    }
});
```

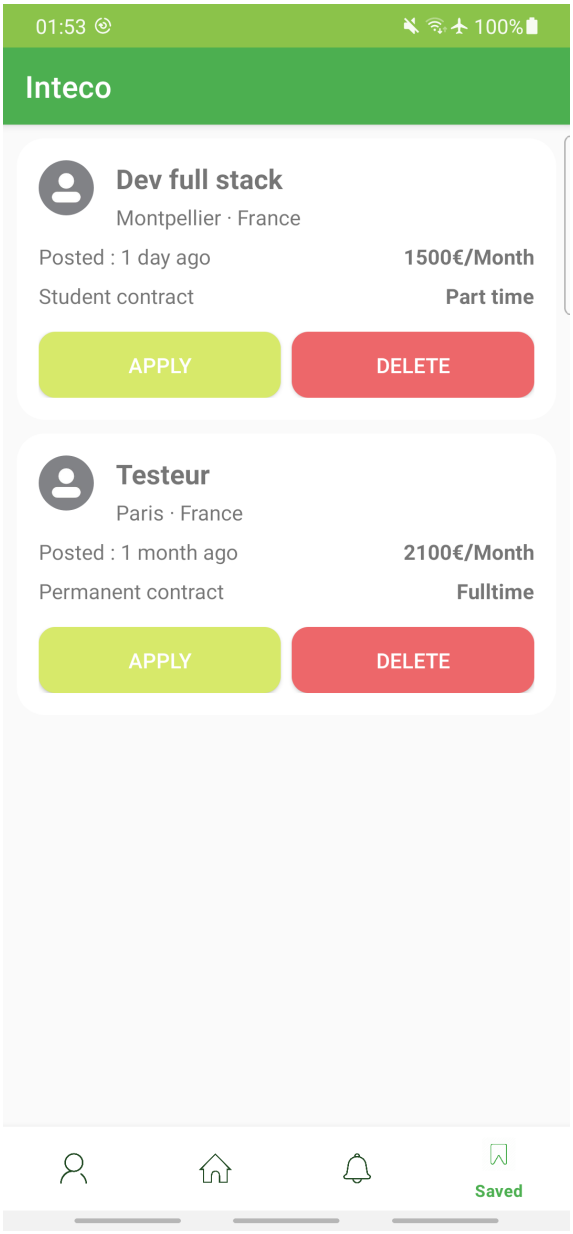
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:background="#FAFAFA"
    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>
```

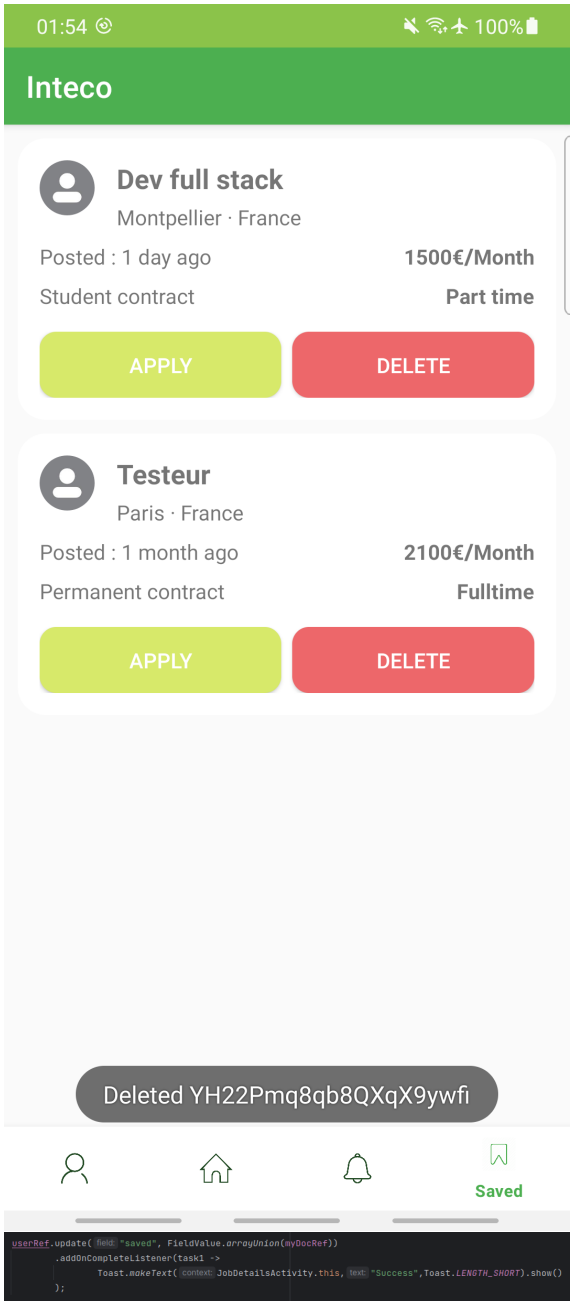


1.5 Sauvegarder

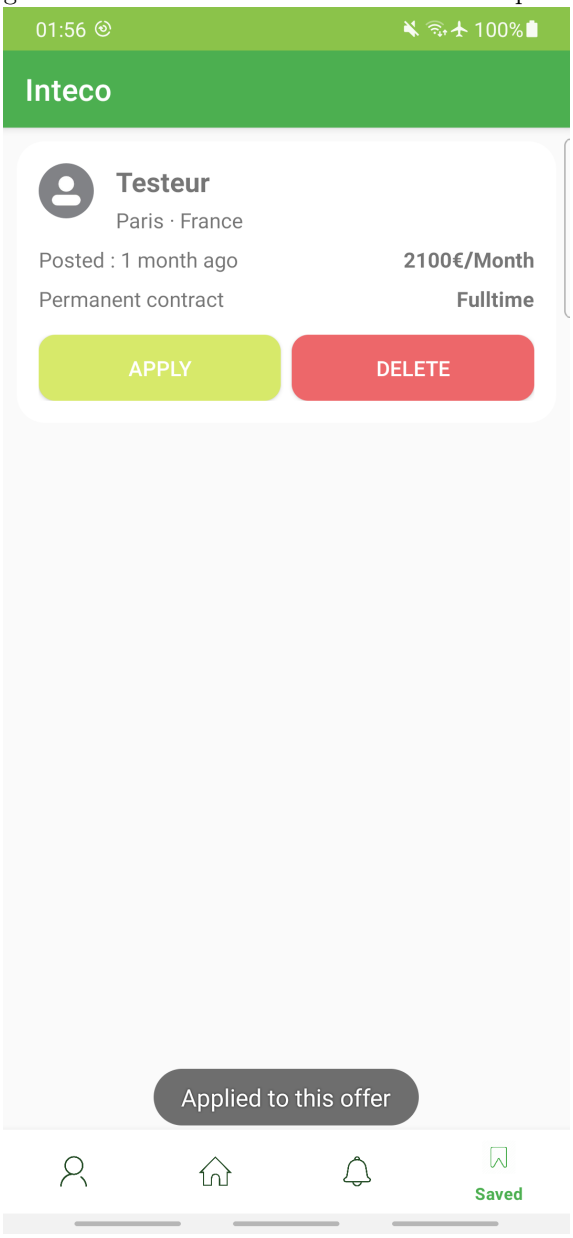
L'application Inteco permet aux chercheurs d'emploi de sauvegarder les offres qui les intéressent et de supprimer ces sauvegardes ultérieurement. Cette fonctionnalité aide les utilisateurs à garder une trace de leurs offres favorites, facilitant ainsi le suivi des opportunités qui les attirent le plus. Grâce à cette option, les chercheurs d'emploi peuvent gérer efficacement leurs recherches et rester organisés dans leur démarche de candidature.


01:53
100%
Inteco
FIXME
Dev full stack
Full-Time - France
Posted 1 day ago -
Job Description
This is a random description
Requirements
• You like fruits
Location
, Montpellier, France
Informations
Position
Senior Wearer
Qualification
PhD
Experience
Very much experience
Contract Type
Student contract
SAVE
APPLY NOW
Current offer already saved


01:53
100%
Inteco
Dev full stack
Montpellier · France
Posted : 1 day ago
1500€/Month
Student contract
Part time
APPLY
DELETE
Testeur
Paris · France
Posted : 1 month ago
2100€/Month
Permanent contract
Fulltime
APPLY
DELETE
Saved

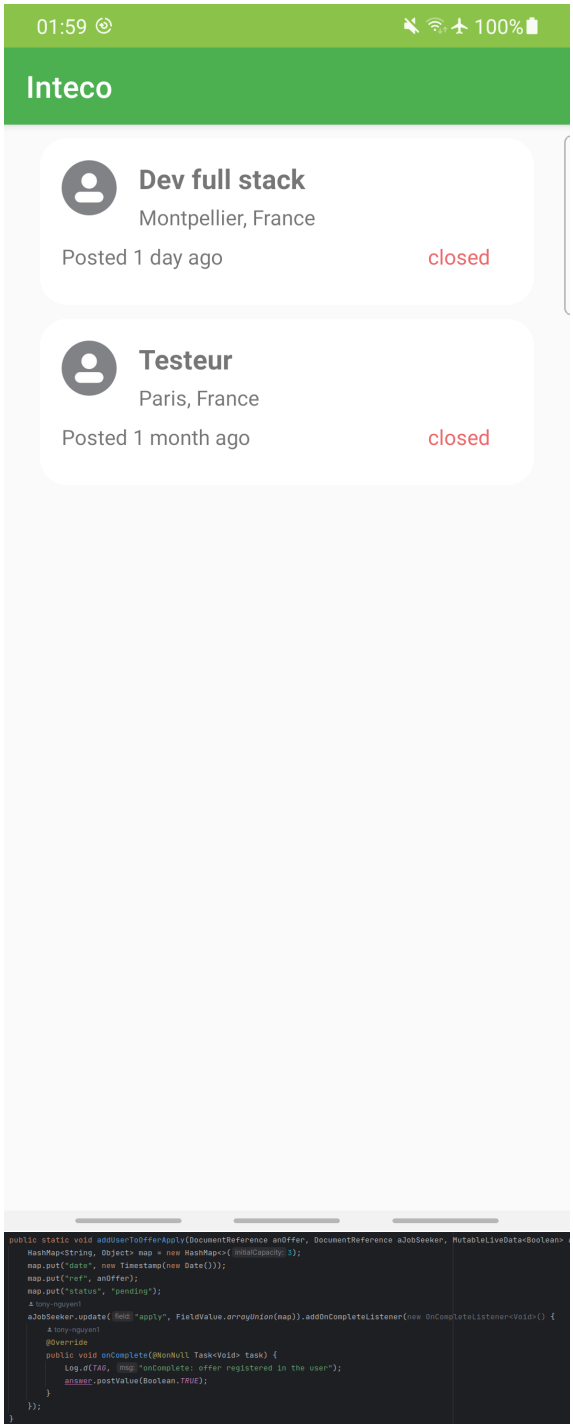


sus de candidature est simplifié et centralisé, offrant aux chercheurs d'emploi une meilleure gestion de leurs efforts de recherche d'emploi.



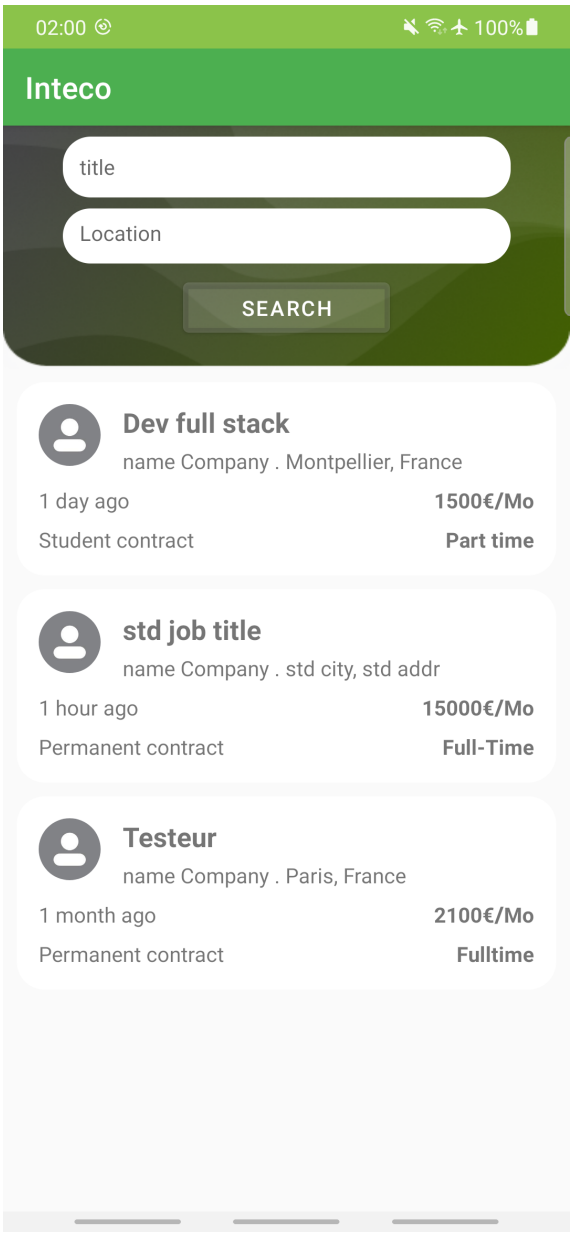
1.6 Postuler

Un chercheur d'emploi peut postuler à une offre directement via l'application Inteco. Cette fonctionnalité permet aux utilisateurs de soumettre facilement leurs candidatures en quelques clics. Les chercheurs d'emploi peuvent joindre leur CV et d'autres documents pertinents pour renforcer leur candidature. De plus, l'application conserve un historique des candidatures envoyées, ce qui aide les utilisateurs à suivre leurs démarches et à ne pas perdre de vue les opportunités pour lesquelles ils ont postulé. Grâce à cette fonctionnalité, le proces-

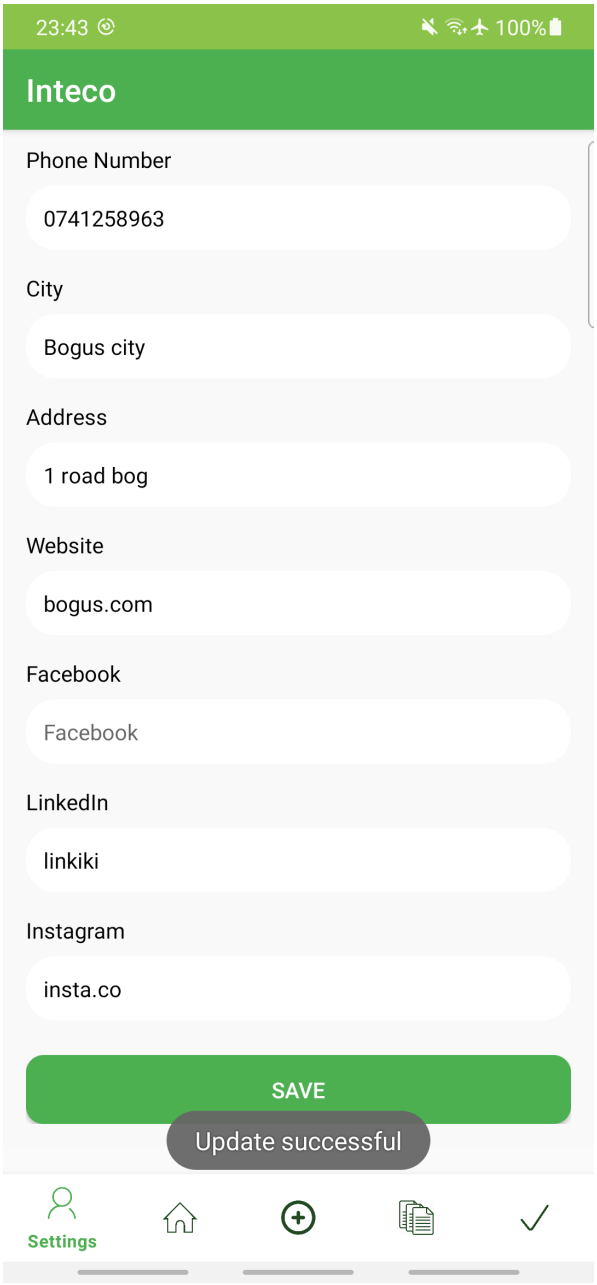


1.7 Rechercher

Un chercheur d'emploi, qu'il soit connecté ou en mode anonyme, peut rechercher parmi les offres disponibles en utilisant des mots-clés présents dans le titre du poste ou en spécifiant une localisation, qui peut inclure un pays ou même une ville. Cette fonctionnalité permet aux utilisateurs de cibler leurs recherches en fonction de leurs critères spécifiques, facilitant ainsi la découverte des opportunités les plus pertinentes. En offrant des options de recherche flexibles et précises, l'application Inteco aide les utilisateurs à trouver rapidement des offres correspondant à leurs compétences et préférences géographiques.



```
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 l = Arrays.asList(tab);
                    for (String keyword : keywordsTitle) {
                        if (l.contains(keyword)) {
                            Log.d(TAG, "onComplete: keyword="+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: keyword="+keyword);
                            thisDocumentIsSelected = true;
                        }
                    }

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



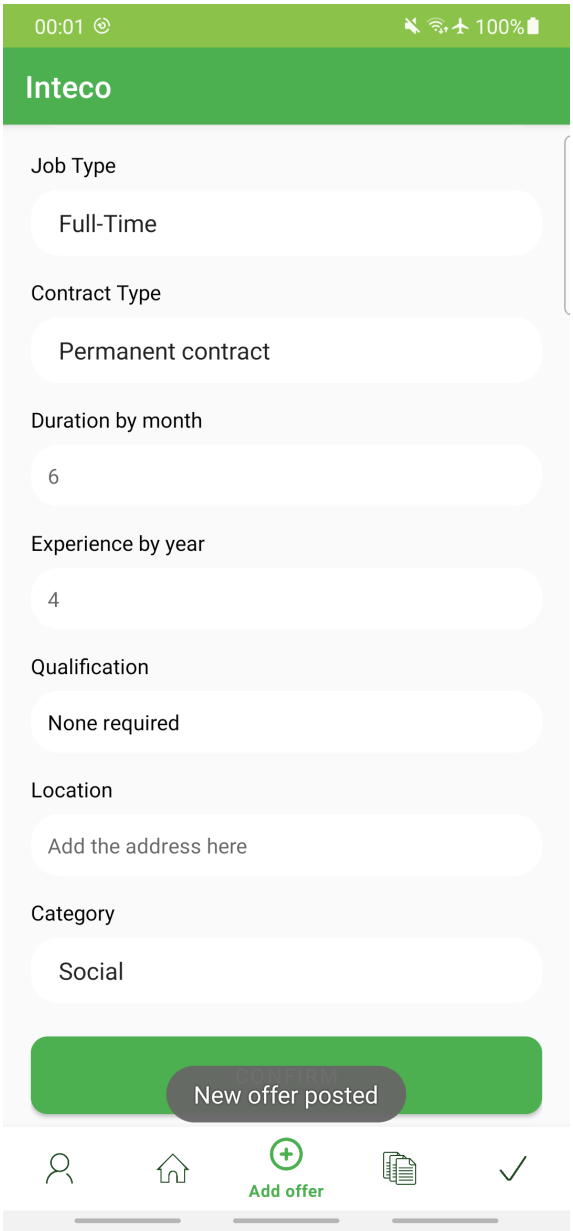
1.8 Profile

Les profils et les fonctionnalités de mise à jour de profil sont disponibles tant pour les chercheurs d'emploi que pour les entreprises. Le fragment Profile affiche les différentes valeurs de l'entreprise et permet la modification de ces informations. Toutefois, les images sont actuellement statiques en raison de contraintes de temps pour l'implémentation backend. Cette fonctionnalité assure que les utilisateurs peuvent maintenir leurs profils à jour, reflétant ainsi précisément leurs compétences, expériences et caractéristiques, tout en offrant une vue d'ensemble claire et concise des informations clés.

Dans les captures d'écrans suivantes, on voit comment afficher les valeurs et les mettre à jour dans Firestore.

```
FirebaseUser currentUser = mAuth.getCurrentUser();
MutableLiveData<QueryDocumentSnapshot> listen = new MutableLiveData<>();
Helper.getCompanyDocumentReference(currentUser, listen);
tony-nguyen1 *
listen.observe(getViewLifecycleOwner(), new Observer<QueryDocumentSnapshot>() {
    tony-nguyen1 *
    @Override
    public void onChanged(QueryDocumentSnapshot aCompany) {
        mCompanyNameEditText.setText(aCompany.getString( field: "name"));
        mEmailEditText.setText(aCompany.getString( field: "email"));
        mPhoneNumberEditText.setText(aCompany.getString( field: "phone"));
        mCityEditText.setText(aCompany.getString( field: "city"));
        mAddressEditText.setText(aCompany.getString( field: "adress"));
        mWebsiteEditText.setText(aCompany.getString( field: "website"));
        mFacebookEditText.setText(aCompany.getString( field: "facebook"));
        mInstagramEditText.setText(aCompany.getString( field: "instagram"));
        mLinkedinEditText.setText(aCompany.getString( field: "linked_in"));
    }
});
```

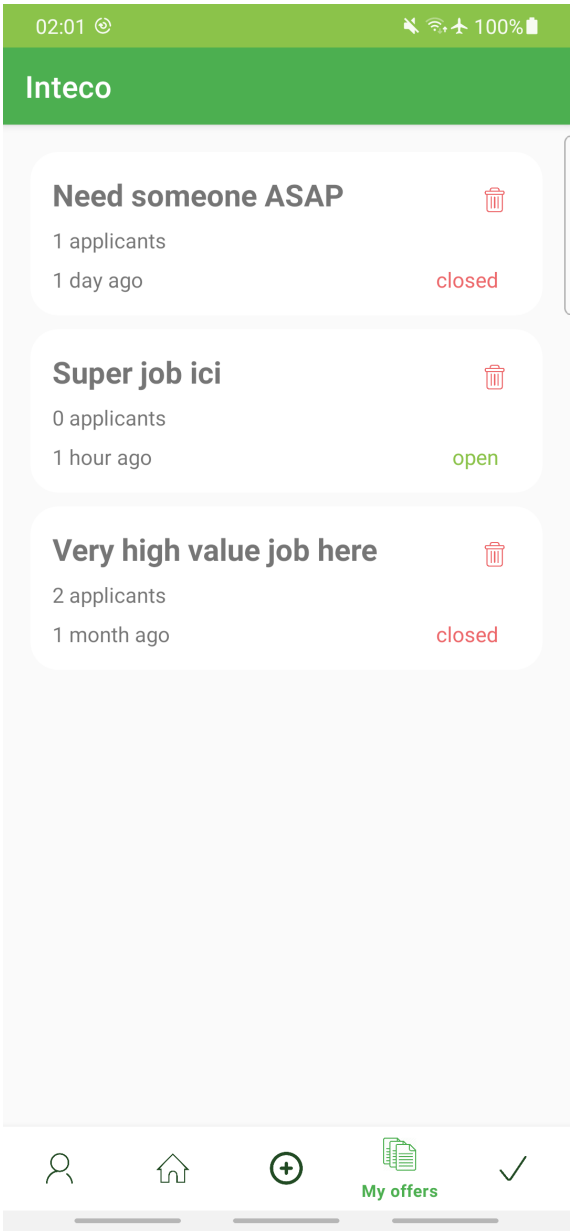
```
public void onChanged(QueryDocumentSnapshot aCompany) {
    Map<String,Object> updatedDataCompany = aCompany.getData();

    updatedDataCompany.replace("name",companyName);
    updatedDataCompany.replace("email",email);
    updatedDataCompany.replace("phone",phoneNumber);
    updatedDataCompany.replace("city",city);
    updatedDataCompany.replace("adress",address);
    updatedDataCompany.replace("website",website);
    updatedDataCompany.replace("facebook",facebook);
    updatedDataCompany.replace("instagram",instagram);
    updatedDataCompany.replace("linked_in",linkedin);

    Log.d(TAG,  msg: "onChanged: "+updatedDataCompany);

    tony-nguyen1
    aCompany.getReference().update(updatedDataCompany).addOnCompleteLi
        tony-nguyen1
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            Toast.makeText(getContext(),  text: "Update successful", Toa
        }
    });
```



1.9 Création d'offre

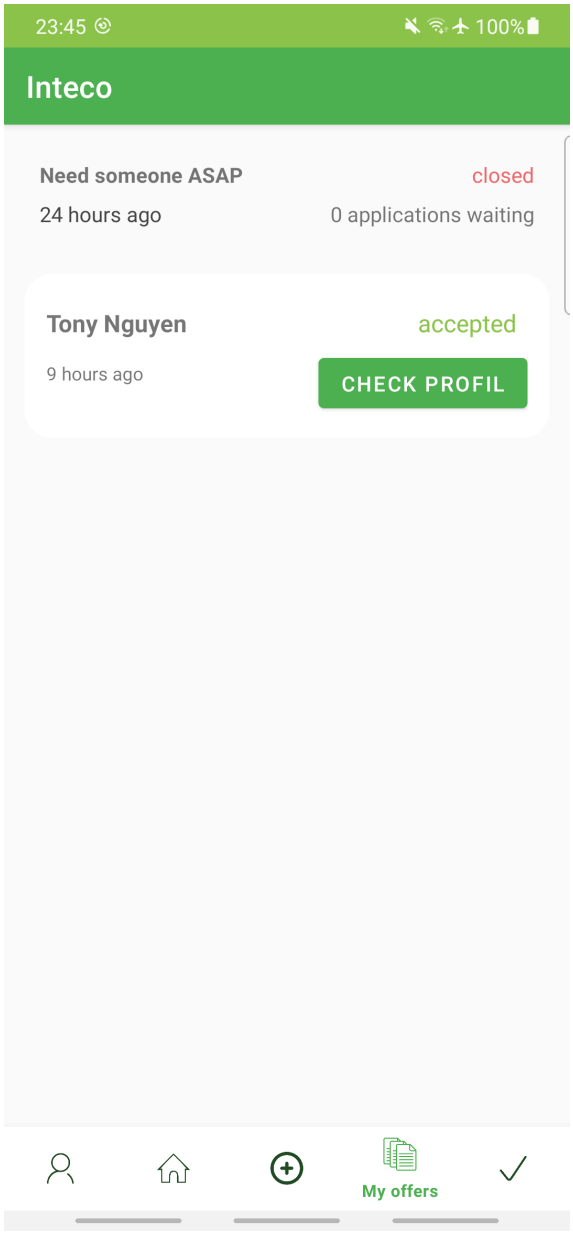
L'application Inteco offre la possibilité de créer de nouvelles offres d'emploi. Les entreprises peuvent facilement publier des annonces pour des postes vacants, en saisissant toutes les informations pertinentes directement via l'application. Cette fonctionnalité permet aux recruteurs de détailler les exigences du poste, les qualifications requises, et les autres informations importantes, assurant ainsi que les offres sont claires et attractives pour les candidats potentiels.

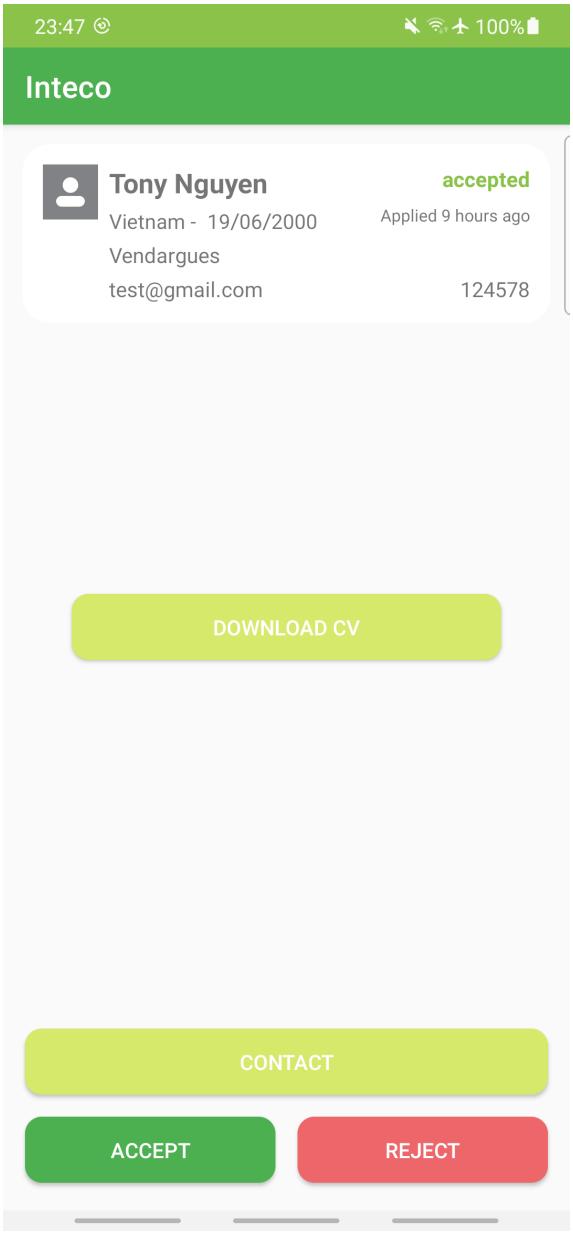
```
HashMap<String,Object> data = new HashMap<>();

data.put("apply", new ArrayList<>());//FIXME
data.put("adress", "std addr");//FIXME
data.put("city", "std city");//FIXME
data.put("contract_type", contractType);
data.put("country", "std addr");//FIXME
data.put("description", description);
data.put("duration", duration);
data.put("experience_wanted", experience);
data.put("job_title", "std job title");//FIXME
data.put("job_type", jobType);
data.put("post_title", title);
data.put("qualification_wanted", qualification);
data.put("realDate", new Timestamp(new Date()));//FIXME
data.put("refCompany", companySnapshot.getReference());
data.put("salary", Long.valueOf(salary.replace( target: " ", repla
data.put("start_time", new Timestamp(new Date()));
data.put("state", "open");

 tony-nguyen1
db.collection( collectionPath: "offers").add(data).addOnSuccessList
    tony-nguyen1
    @Override
    public void onSuccess(DocumentReference offerReference) {
        Helper.addPostToCompany(companySnapshot.getReference()
        Toast.makeText(getContext(), text: "New offer posted",
    }
```

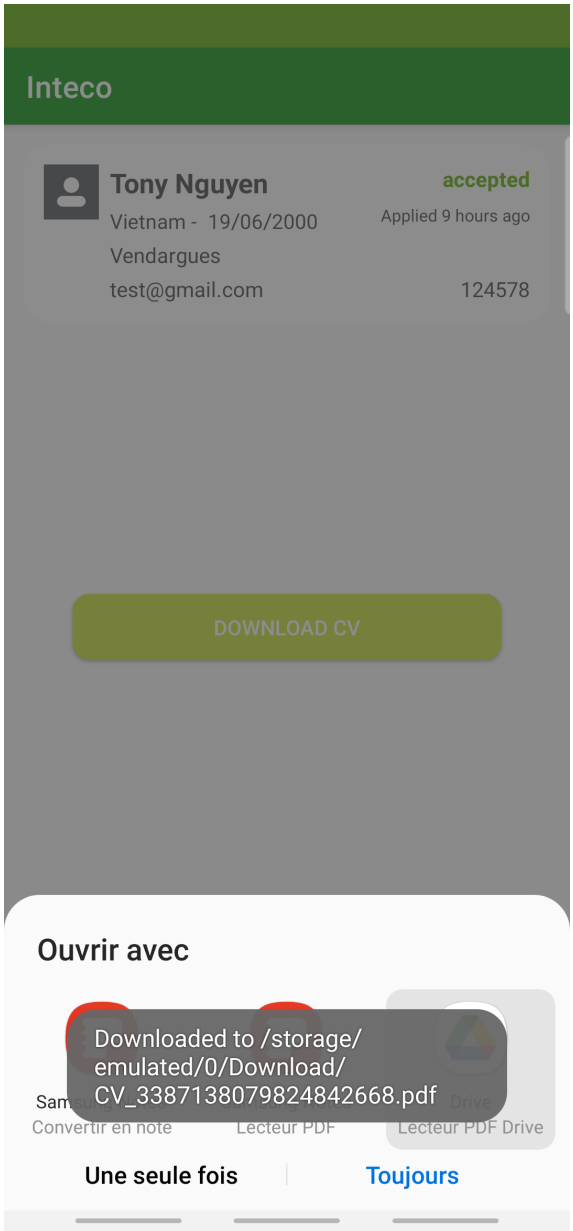


1.10 Visualisation des offres

Après avoir créer des offres, il est possible de les visualiser, de voir les candidats qui ont postulé dans une liste puis en détails. Dans le profil du candidat, nous pouvons les accepter ou les refusés et les contacter.


23:45
100%
Inteco
Need someone ASAP
closed
24 hours ago
0 applications waiting
Tony Nguyen
accepted
9 hours ago
CHECK PROFIL
My offers


23:47
100%
Inteco
Tony Nguyen
accepted
Vietnam - 19/06/2000
Applied 9 hours ago
Vendargues
test@gmail.com
124578
DOWNLOAD CV
CONTACT
ACCEPT
REJECT



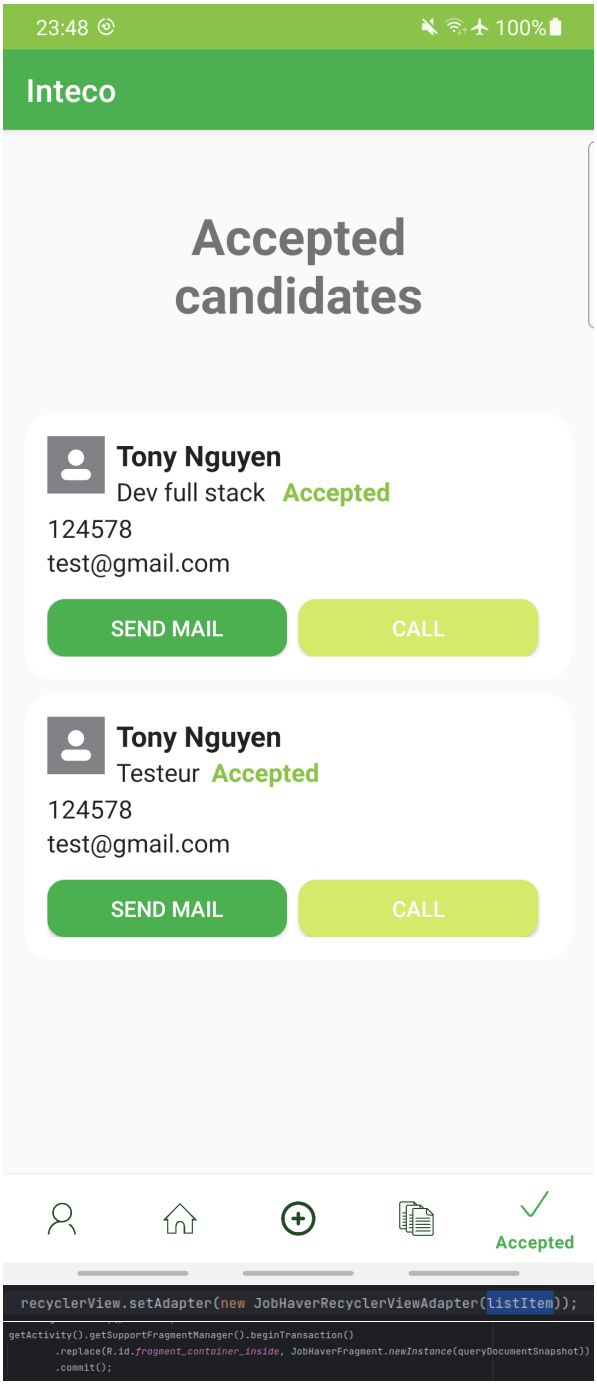
Voici un petit extrait de code, démontrant l'utilisation d'un RecyclerView.Adapter permettant de créer une liste de candidats.

```
public void onBindViewHolder(final ViewHolder holder, int position) {
    holder.mItem = mValues.get(position);
    holder.mIdView.setText(mValues.get(position).id);
    holder.mContentView.setText(mValues.get(position).content);

    holder.mJobTitle.setText("title");
    holder.mNbApplicants.setText("16 applicants");
    holder.mDate.setText("date");
    holder.mStatus.setText("status");

    holder.mJobTitle.setText(mValues.get(position).jobTitle);
    holder.mNbApplicants.setText(String.valueOf(mValues.get(position).numberApplicants)+" applicants");
    holder.mDate.setText(mValues.get(position).dateDetails);
    holder.mStatus.setText(mValues.get(position).state);

    // Set the text color based on the state value
    Context context = holder.mStatus.getContext();
    if ("open".equalsIgnoreCase(mValues.get(position).state)) {
        holder.mStatus.setTextColor(ContextCompat.getColor(context, R.color.status_open));
    } else if ("closed".equalsIgnoreCase(mValues.get(position).state)) {
        holder.mStatus.setTextColor(ContextCompat.getColor(context, R.color.status_false));
    }

    holder.itemView.setOnClickListener( v -> {
        Log.d(TAG, msg: "onBindViewHolder: click");
        itemClickListener.onItemClickListener(mValues.get(position), position);
    });
```

```
StorageReference islandRef = storageRef.child( pathString: "pdf/"+queryDocumentSnapshot.getString( field: "cv"));
  StorageReference islandRef = storageRef.child("images/doc274903706990755539 9.pdf"); // le nom de ce qu'on
Log.d(TAG, msg: "onChanged: "+islandRef);

islandRef.getFile(finalLocalFile).addOnSuccessListener(new OnSuccessListener<FileDownloadTask.TaskSnapshot>
```

1.11 Visualisation des embauches

Ensuite, dans la rubrique "Accepted", un récapitulatif de tout les candidat accepter.

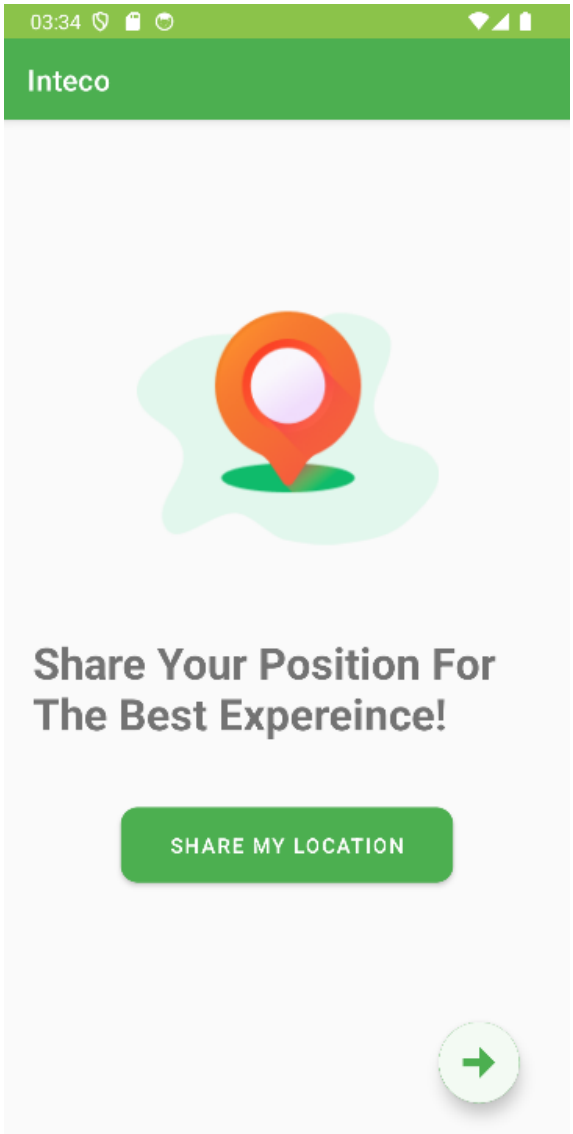


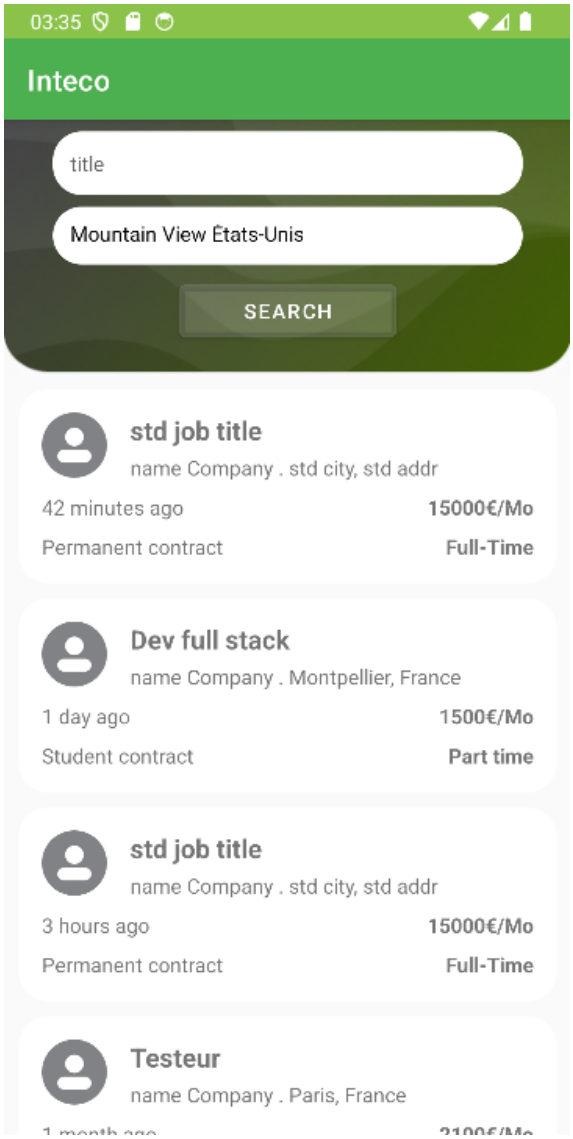
1.12 Facilité d'utilisation

L'application Inteco a été conçue pour être intuitive et accessible, offrant une expérience utilisateur agréable et efficace.

1.12.1 Localisation GPS

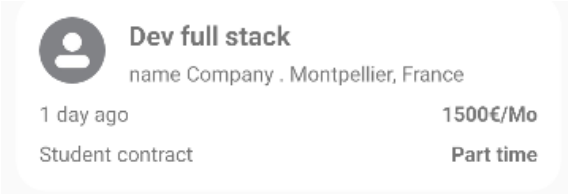
Dans l'écran d'accueil, nous avons ajouté une option permettant à l'utilisateur d'activer la localisation GPS. Si l'utilisateur refuse initialement, il sera à nouveau invité à activer le GPS lors de la recherche d'offres. Une fois accepté, l'application utilise la localisation pour remplir automatiquement la ville et le pays de l'utilisateur dans les critères de recherche, permettant ainsi d'obtenir des résultats plus personnalisés. L'utilisateur peut alors simplement cliquer sur "Recherche" pour voir les offres pertinentes dans sa région.

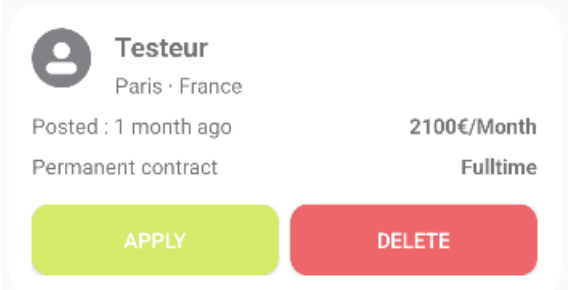




1.12.2 Affichage du temps

Le calcul du temps écoulé depuis la publication des offres (par exemple, "il y a 1 an", "il y a 7 jours") est une fonctionnalité clé qui améliore l'expérience utilisateur. Cette fonctionnalité permet aux utilisateurs de rapidement évaluer la fraîcheur des offres d'emploi et de prioriser leurs candidatures en fonction de la récence des publications.





Conclusion

Inteco se veut être un outil complet et efficace pour faciliter le processus de recrutement et de recherche d'emploi, tout en offrant une expérience utilisateur optimale et sécurisée. Pour conclure, la majeure partie des fonctionnalités sont présentes et fonctionnent, même si certains détails restent à corriger.

L'application propose un processus d'inscription rigoureux, garantissant la complétude et la précision des informations des utilisateurs. Les chercheurs d'emploi et les entreprises bénéficient d'une interface moderne et conviviale, avec une barre de navigation réactive qui assure une transition fluide entre les différentes sections.

Les fonctionnalités de sauvegarde et de suppression des offres permettent aux chercheurs d'emploi de garder une trace de leurs opportunités préférées, tandis que le système de candidature simplifie le processus de postulation. La recherche avancée par mots-clés et localisation offre une flexibilité accrue pour trouver des offres pertinentes.

De plus, l'authentification via Firebase renforce la sécurité des comptes et permet une gestion facile, incluant la réinitialisation des mots de passe. Les entreprises peuvent également créer et modifier des offres d'emploi, et afficher leurs valeurs de manière claire grâce au fragment Profile.