

# Rapport de Projet Fédérateur

## Bricoly

Bricoly : Conception et Développement d'une plateforme qui relie les prestataires avec les demandeurs de services.

### **Réalisé par :**

DOUSLIMI Yasser, MOUHIM Hanane

SIRAJ EDDINE Youssef, TALBI Hicham

### **Encadré par :**

Pr. OULAD HAJ THAMI Rachid

### **Jury :**

Pr. OULAD HAJ THAMI Rachid

Pr. ABIK Mounia



# REMERCIEMENT

A tous ceux qui, de près ou de loin ont contribué à l'accomplissement de ce projet.

A ceux qui ont été par leur disponibilité et leur collaboration d'un grand soutien.

Nous tenons également à remercier tous nos collègues et nos enseignants pour leurs soutiens et aides.

En fin, nos remerciements vont aux jurys pour avoir bien voulu juger ce travail. En espérant que ce travail sera utile et à la hauteur des attentes des utilisateurs.

Cordialement.

## Dédicace

---

A nos chers parents

A nos chers frères

A nos chères sœurs

A toute notre famille

A nos professeurs

A tous nos amis

A tous ceux qu'on aime

A tous ceux qui nous aiment

On dédie ce travail, à titre de reconnaissance

---

## Résumé

Le présent rapport résume le travail que nous avons réalisé dans le cadre de notre projet de fin de la 3<sup>ème</sup> année.

Ce projet consiste à réaliser une application qui permet à prestataire de publier son offre selon son métier, cette offre contient des informations concernant sa disponibilité, son contact et son métier. Le demandeur de service à son tour peut chercher parmi les prestataires selon son besoin, à travers des catégories, et entre en contact avec un ou plusieurs prestataires.

Toutes les étapes ont été respectées, allons de l'authentification jusqu'à l'implémentation de l'interface, en passant par l'analyse et la conception, accompagnés des tests pour un fonctionnement assuré.

## Abstract

This report summarizes the work we have done in the framework of our 3rd year project.

This project consists in the realization of an application that allows a service provider to publish his offer according to his profession; this offer contains information about his availability, his contact and his profession. The service seeker in turn can search among the providers according to his need, through categories, and get in touch with one or several providers.

All the steps have been respected, from authentication to the implementation of the interface, including analysis and design, accompanied by tests to ensure that it works.

# Table des matières

Table des matières.....	7
Introduction Générale .....	9
Chapitre 1: Contexte général du projet.....	10
1.1 Problématique.....	10
1.2 Solution proposée.....	11
Chapitre 2 : Analyse et conception.....	12
2.1 Analyse.....	13
2.1.1 Besoins fonctionnels .....	13
2.2 Conception .....	15
2.2.3 Architecture de l'application .....	18
Chapitre 4 : Réalisation et tests.....	19
4 .1 Outils de travail.....	20
4 .2 Réalisation de l'application.....	23
□ Page principale .....	23
□ Page d'accueil .....	24
□ La page d'une offre .....	25
□ Page d'un profil .....	26
□ Page modification de profil.....	27
□ Historiques des missions du prestataire.....	28
□ Liste des offres posés par les prestataires.....	29

□ Authentification .....	30
□ Inscription .....	31
Conclusion générale et perspectives .....	32
Bibliographie .....	33



# Introduction Générale

Pour assurer une formation professionnelle performante aux étudiants, les cours proposés au sein des établissements sont complétés par un projet fin d'année permettant de valider les connaissances déjà acquises tout au long de la formation et de faire une comparaison entre la théorie étudiée et la pratique afin de concrétiser ces connaissances et être plus proche au domaine de l'emploi.

Dans le cadre de la formation à l'Ecole Nationale Supérieure d'informatique et d'analyse des systèmes, et afin d'appliquer les méthodologies et les notions enseignées tout au long de l'année, nous devons réaliser un projet fin d'année.

La mission qui nous a été confiée est tout d'abord, la réalisation d'une application qui relie les prestataires et les demandeurs de services, développée avec le Framework Spring Boot et Angular, en travaillant sur toutes les fonctionnalités fournies par celui-ci.

Notre travail s'articulera autour de trois chapitres dont le premier sera consacré à la présentation du projet, le deuxième abordera l'analyse et conception du problème. Enfin, le dernier présentera les différentes démarches adoptées pour réaliser l'application. Une webographie sera également mise à la disposition du lecteur.

# **Chapitre 1 : Contexte général du projet**

Ce chapitre a pour objectif de situer le projet dans son contexte général, à savoir la problématique qui a inspiré la création de notre application, la description du projet et les objectifs à atteindre.

## **1.1 Problématique**

Lorsqu'un individu rencontre un problème qui nécessite un prestataire, généralement il commence rapidement à chercher un prestataire, en revanche ceci pose un problème de perte de temps puisque il demande parfois de se déplacer pour trouver un prestataire.

Ceci présente un vrai problème chez ceux qui ne peuvent pas se déplacer ou chez les personnes âgées.

## 1.2 **Solution** proposée

Ce projet consiste à réaliser une application qui permet à prestataire de publier son offre selon son métier, cette offre contient des informations concernant sa disponibilité, son contact et son métier. Le demandeur de service à son tour peut chercher parmi les prestataires selon son besoin, à travers des catégories, et entre en contact avec un ou plusieurs prestataires.

De cette manière les individus garantissent un gain de temps et ne sont pas censés de se déplacer afin de trouver un prestataire.

Cette application présente des interfaces pour chacun des utilisateurs tel que le prestataire, le demandeur de service et l'administrateur système.

Nous avons entamé ce chapitre par la problématique, elle nous a dirigé vers le développement de notre application qui est l'objectif principal, dans le chapitre suivant on verra la modélisation des fonctionnalités des espaces vues précédemment.

## **Chapitre 2 : Analyse et conception**

La réussite de tout projet dépend de la qualité de son départ. De ce fait, l'étape de spécification des besoins constitue la base de départ de notre travail, elle doit décrire sans ambiguïté le site web à développer.

Pour assurer les objectifs attendus, il est essentiel que nous parvenions à une vue claire des différents besoins escomptés de notre projet. Au cours de ce chapitre, nous allons dégager les fonctionnalités attendues en définissant les différents cas d'utilisation qui expliquent ces cas.

## 2.1 Analyse

### 2.1.1 **Besoins** fonctionnels

Les besoins fonctionnels sont les besoins qui déterminent les fonctionnalités indispensables auxquels doit répondre l'application.

L'application en sa globalité doit répondre aux exigences fonctionnelles des utilisateurs suivants :

- Un prestataire qui peut publier son offre selon son métier, tout en précisant des informations sur sa disponibilité, son contact et sa zone géographique.
- Un demandeur de service (un client) qui consulte les offres posés par les prestataires et entre en contact avec un ou plusieurs, selon la nature de son problème.
- Un administrateur système qui gère les comptes prestataires et les clients.

## 2.1.2 **Besoins** non fonctionnels

Les besoins non-fonctionnels sont des besoins/contraintes liés à l'implémentation et à l'interopérabilité générale. Et pour compléter les besoins fonctionnels, le projet devra respecter un ensemble de propriétés contribuant à une meilleure qualité de la solution obtenue. Parmi ces critères on trouve :

**Sécurité** : L'application doit être sécurisée pour protéger les données sensibles des utilisateurs de l'application comme les informations de connexion, les informations des demandes de changements...etc.

**Temps de réponse** : la première chose à faire pour penser à améliorer la vitesse de chargement d'une application trop lent est de tourner vers un hébergement plus performant.

**Fiabilité** : L'application ne doit pas afficher aux utilisateurs des erreurs, par contre toutes les exceptions doivent être gérées et toute incidence doit être déclaré dans les logs du coté back-end pour l'identifier facilement, et pour la résoudre rapidement.

**La convivialité** : Le futur logiciel doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.

## 2.2 Conception

Après l'étape de l'étude préalable et la spécification des besoins, nous réservons cette partie à la conception et la modélisation du projet. En effet, nous allons décrire les différents modèles statiques et dynamiques, en utilisant le formalisme UML.

### 2.2.1 **Diagramme** des cas d'utilisations global

On verra dans cette partie la représentation graphique des fonctionnalités et des tâches de chacun des utilisateurs de l'application, la solution contient 3 acteurs :

- ❖ Le demandeur de service (le client), il effectue les opérations suivant :
  - Recruter un ou plusieurs prestataires.
  - Noter un prestataire.
  - Créer un profil.
  
- ❖ Le prestataire, il effectue les opérations suivant :
  - Créer une ou plusieurs offres.
  - Créer un profil.
  - Accepter ou refuser une offre d'un demandeur de service.
  
- ❖ L'administrateur, il effectue les opérations suivant :
  - Gérer les utilisateurs et les ressources.

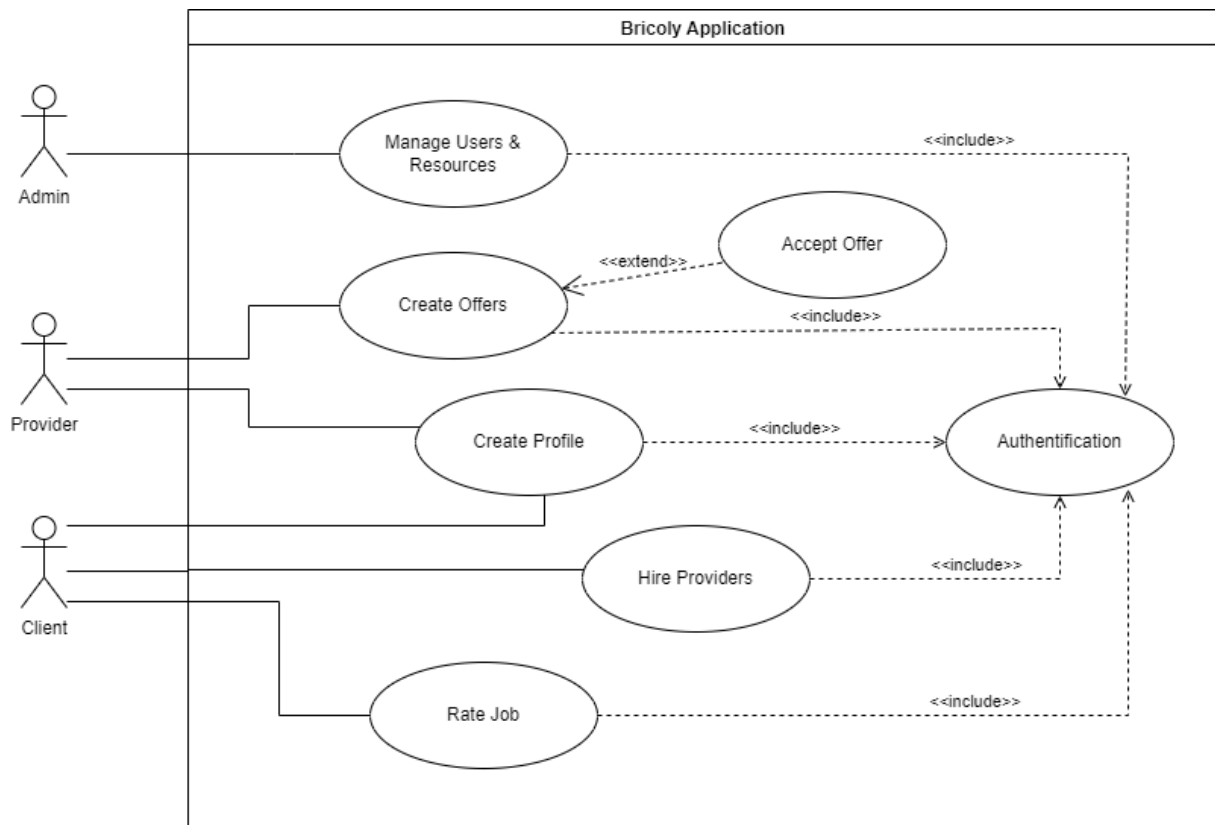


Figure 1 : Diagramme de cas d'utilisations

- Gérer les utilisateurs et les ressources : c'est la création, modification ou suppression d'un ou plusieurs utilisateurs ou une ressource.
- Créer les offres : c'est la création d'une ou plusieurs offres en mettant les informations nécessaires demandée lors de la création.
- Recruter les prestataires : un demandeur de service peut recruter les prestataires à travers les contacter.
- noter un prestataire : c'est le point de vu d'un demandeur de service après le service rendu par un prestataire.
- Accepter une offre : un prestataire peut accepter ou refuser une offre demandée par un demandeur de service.



- Gérer les partenaires : c'est la gestion des collaborateurs qui collabore avec ATNER dans la réalisation du projet cité dans l'appel d'offre.
- Authentification : toute opération dans l'application nécessite d'être authentifié.

## 2.2.2 Diagramme de classe

Le diagramme de classe exprime la structure statique du système en termes de classes et de relations entre ces classes.

L'intérêt du diagramme de classes est de modéliser les entités du système d'information.

La figure ci-dessous représente le diagramme de classes des entités utilisées dans notre système :

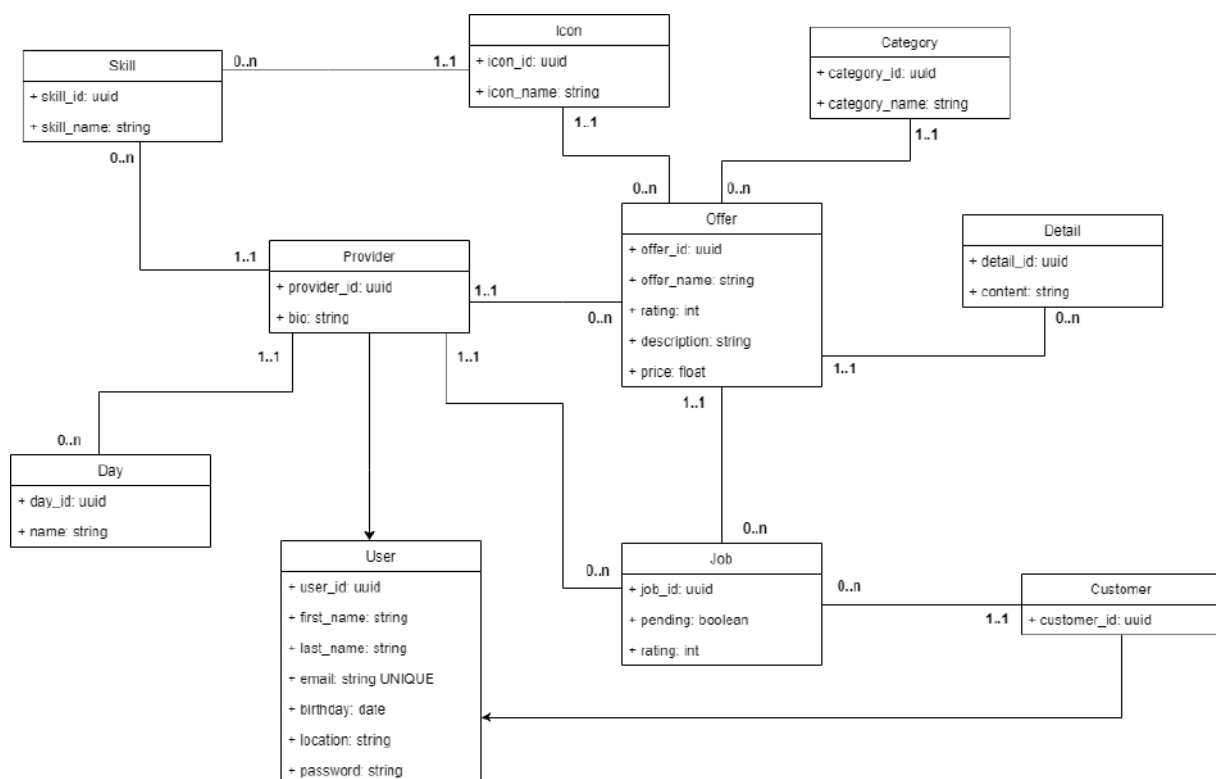


Figure 2 : Diagramme de classe

## 2.2.3 Architecture de l'application

Nous avons opté pour l'architecture MVC pour le développement de notre application.

Le modèle MVC (Modèle, Vue, Contrôleur) est une architecture qui organise les composants d'une application. Son principe consiste à séparer les données, les interfaces graphiques et les traitements en trois parties distinctes : modèle, vue et contrôleur.

- ✚ **Modèle** : correspond aux données. Il s'agit en général de plusieurs classes qui permettent d'accéder aux données.
- ✚ **Vue** : Couche PRÉSENTATION. C'est la présentation graphique de l'application (l'interface utilisateur).
- ✚ **Contrôleur** : Le contrôleur est un relieur entre le modèle et la vue. Son rôle consiste à analyser les événements provenant des vues et commander au(x) modèle(s) d'exécuter les actions appropriées.

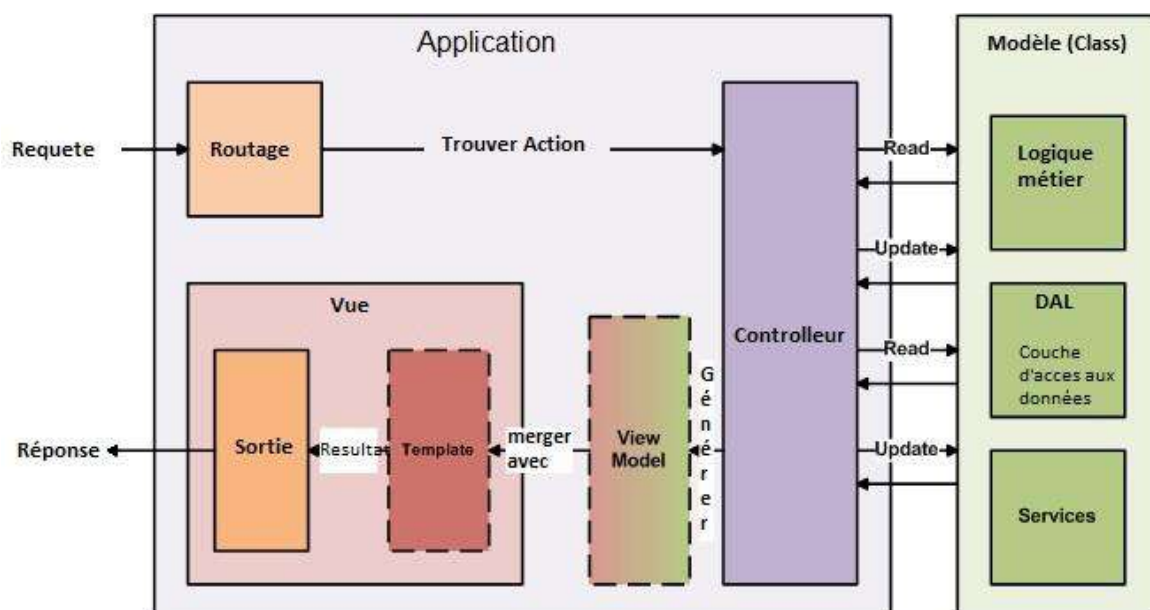


Figure 3 : L'architecture MVC

Ce chapitre met en évidence la conception de notre application ainsi que l'analyse. Dans le chapitre suivant, on verra les tests et la réalisation.

## **Chapitre 4 : Réalisation et tests**

Dans ce chapitre, nous allons mentionner les outils de travail que nous avons choisis pour la réalisation de notre projet et les résultats que nous avons obtenus.

## 4.1 Outils de travail



Angular est un framework côté client, open source, basé sur TypeScript, et co-dirigé par l'équipe du projet « Angular » à Google et par une communauté de particuliers et de sociétés. Angular est une réécriture complète d'AngularJS, cadriciel construit par la même équipe.



Java est une technique informatique développée initialement par Sun Microsystems puis acquise par Oracle à la suite du rachat de l'entreprise. Défini à l'origine comme un langage de programmation, Java a évolué pour devenir un ensemble cohérent d'éléments techniques et non techniques



Spring Boot est un nouveau framework créé par l'équipe de chez Pivotal, conçu pour simplifier le démarrage et le développement de nouvelles applications Spring. ... La plateforme Spring IO a vivement été critiquée au fil des années à cause de la configuration XML volumineuse et d'une gestion des dépendances complexe.



Docker est une plateforme permettant de lancer certaines applications dans des conteneurs logiciels. Selon la firme de recherche sur l'industrie 451 Research, « Docker est un outil qui peut emballer une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur ».



Bootstrap est un Framework CSS, mais pas seulement, puisqu'il embarque également des composants HTML et JavaScript. Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style pour les boutons, les formulaires, la navigation ... Il permet ainsi de concevoir un site web rapidement et avec peu de lignes de code ajoutées.



HTML & CSS : HTML [4] : est un langage dit de « marquage » (de « structuration » ou de « balisage ») dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents.

CSS [4] : feuille de style c'est un langage qui permet de gérer l'apparence de la page web (agencement, positionnement, décoration, couleur, taille du texte...).



## kubernetes

Kubernetes est un système open source qui vise à fournir une « plate-forme permettant d'automatiser le déploiement, la montée en charge et la mise en œuvre de conteneurs d'application sur des clusters de serveurs ».



Github est un service web permettant l'hébergement des développements de logiciel, utilisant Git. Git est un gestionnaire de version de fichier qui permet d'enregistrer toutes les modifications sur un fichier, permettant de revenir en arrière en cas d'erreur sur celui-ci.

L'avantage de ce service est de pouvoir travailler de manière collaborative distribuée.

En effet chacun peut travailler sur ses tâches en parallèle et ensuite l'envoyer sur la plateforme. Le second avantages, Github favorise le travail à distance car les fichiers sont hébergés sur le serveur et accessible gratuitement.



PostgreSQL est un système de gestion de base de données relationnelle et objet. C'est un outil libre disponible selon les termes d'une licence de type BSD. Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres, ou propriétaires.

## 4.2 Réalisation de l'application

- **Page principale**

C'est la première page qui apparaît lors de l'ouverture de l'application.

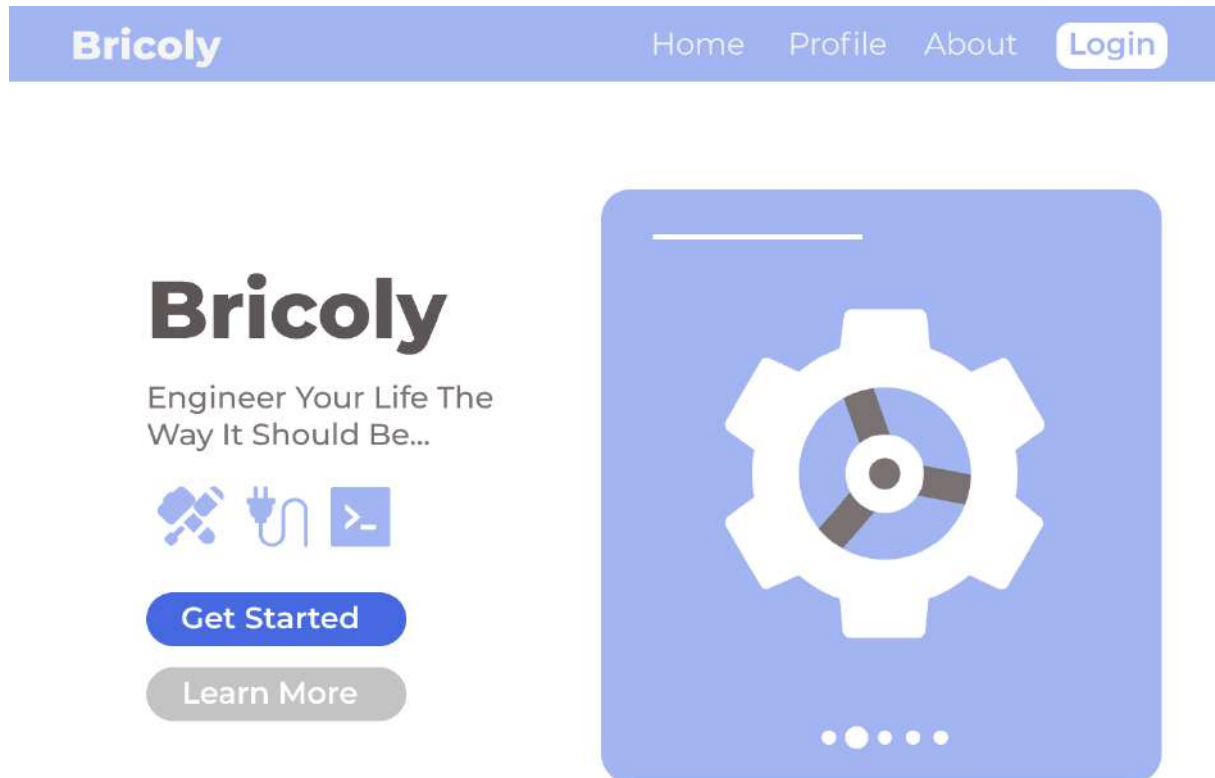


Figure 4 : Page principale

- **Page d'accueil**

La page d'accueil est la page qui apparait quand on clique sur **'Get Started'**. Cette page contient les offres par défaut de l'application selon les catégories.

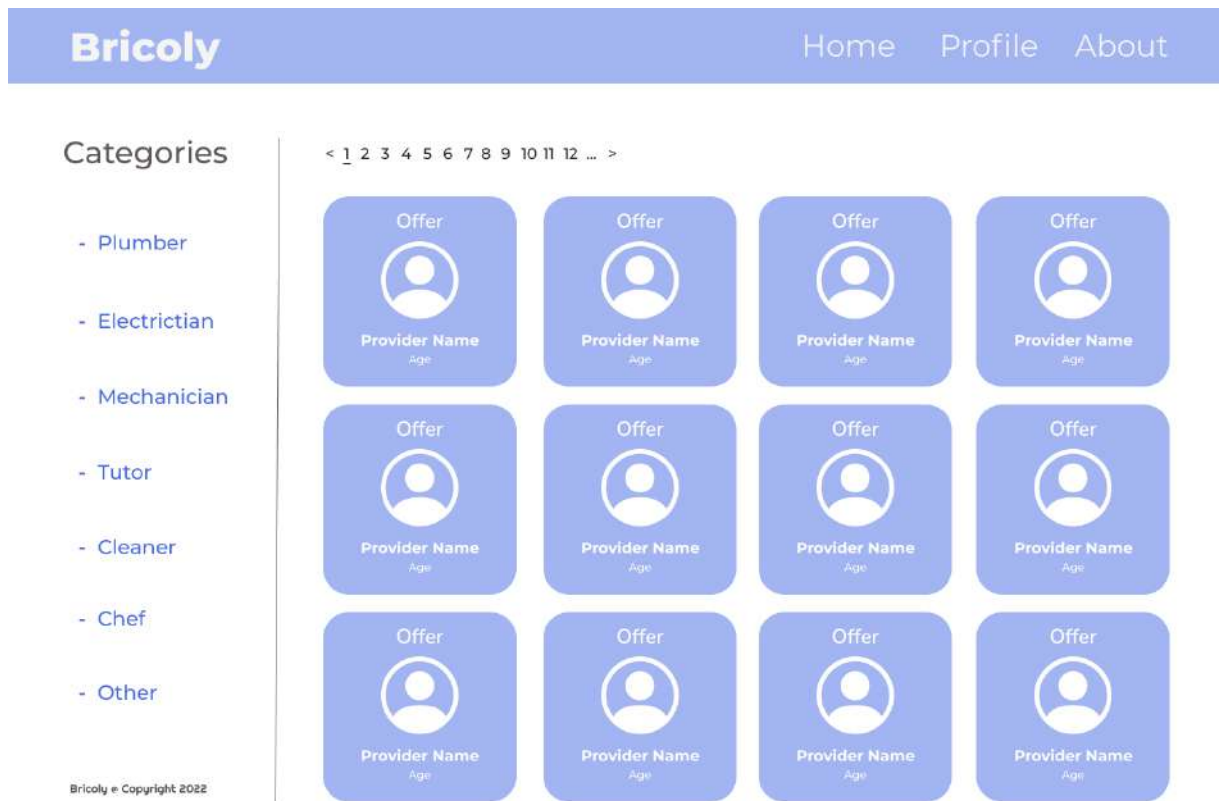


Figure 5 : Page d'accueil



- **La page d'une offre**

Lorsqu'on sélectionne une offre, cette page apparait en affichant les détails de chaque offre.

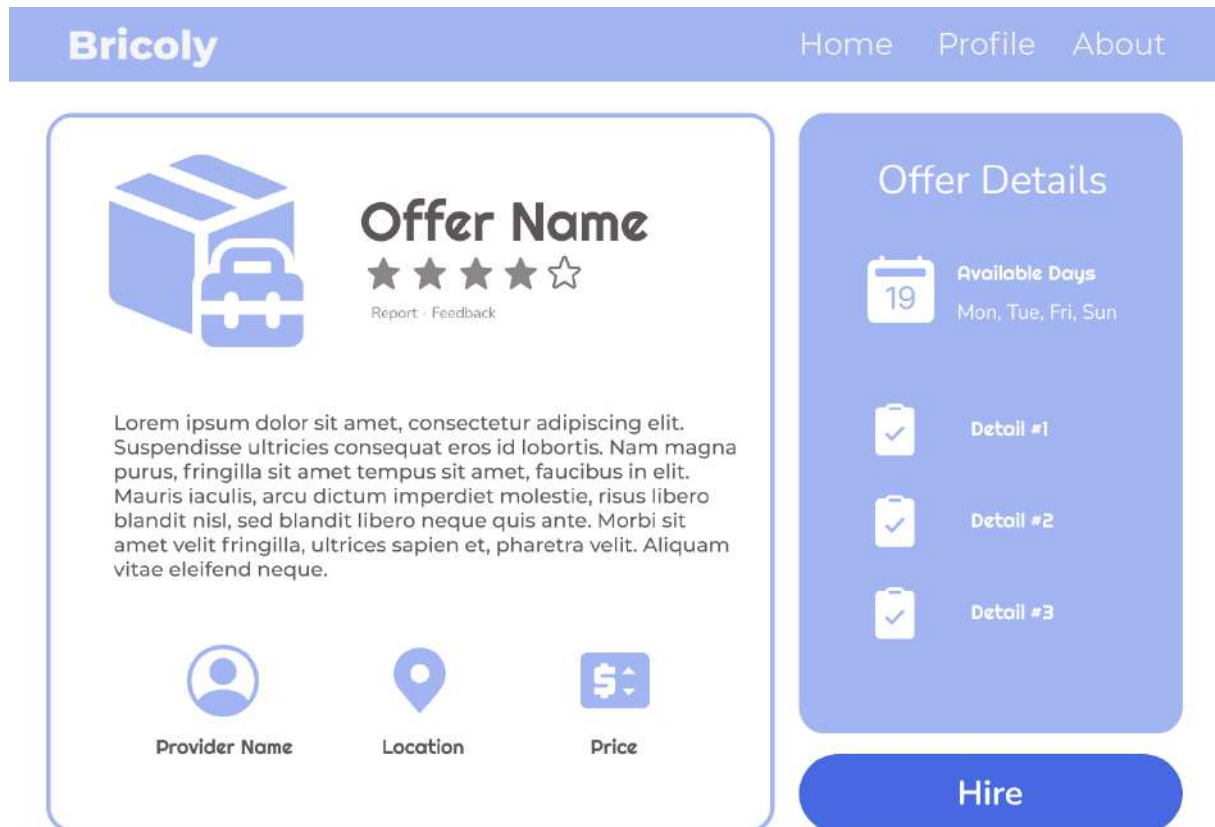


Figure 6 : Page d'une offre

- **Page d'un profil**

Lorsqu'on sélectionne un profil, cette page apparaît tout en affichant les détails du profil.

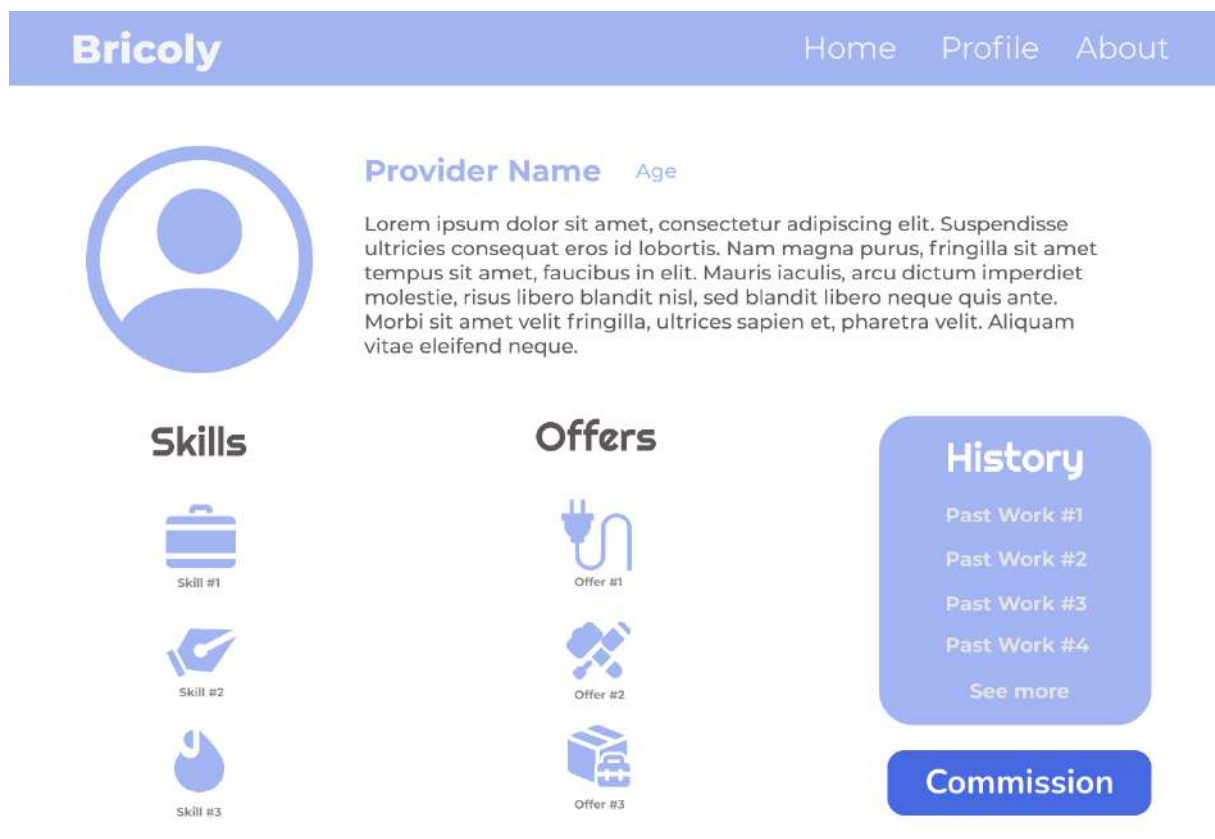


Figure 7 : Page d'un profil

- **Page modification de profil**

Cette page donne la main pour modifier et saisir les informations personnelles.

The image shows a web interface for a profile modification page. On the left is a blue sidebar with the logo 'Bricoly' at the top. Below the logo are three menu items: 'Profile', 'Offers', and 'Jobs'. At the bottom of the sidebar is a white 'Logout' button and a small copyright notice 'Bricoly © Copyright 2022'. The main content area has a light blue background. At the top left of this area is the word 'Profile' in a large, bold, blue font. To its right are two buttons: a blue 'Update' button and a red 'Change Password' button. Below these are four horizontal input fields, each with a label on the left and a white input box on the right: 'First Name', 'Last Name', 'Birthday', and 'Location'. At the bottom of the main area are two boxes. The left box is titled 'Skills' and contains three horizontal input fields. The right box is titled 'Bio' and contains a large rectangular text area.

Figure 8 : Page modification du profil

- **Historiques des missions du prestataire.**

Dans cette page, les prestataires consultent les services traités, ils sont notés par les demandeurs de service (les clients).

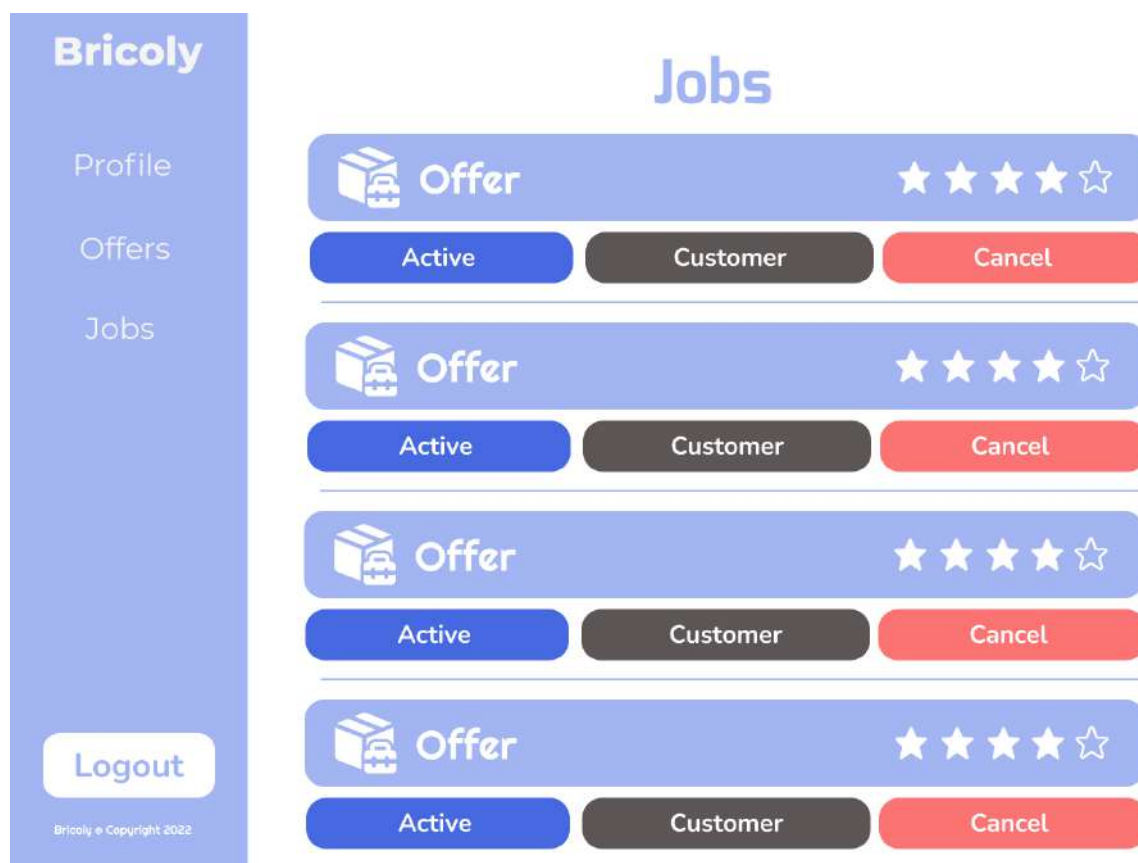


Figure 9 : Historique des services des demandeurs de service

- **Liste des offres posés par les prestataires**

Dans cette section les prestataires posent leurs offres en précisant le nom, le prix et la description de l'offre.

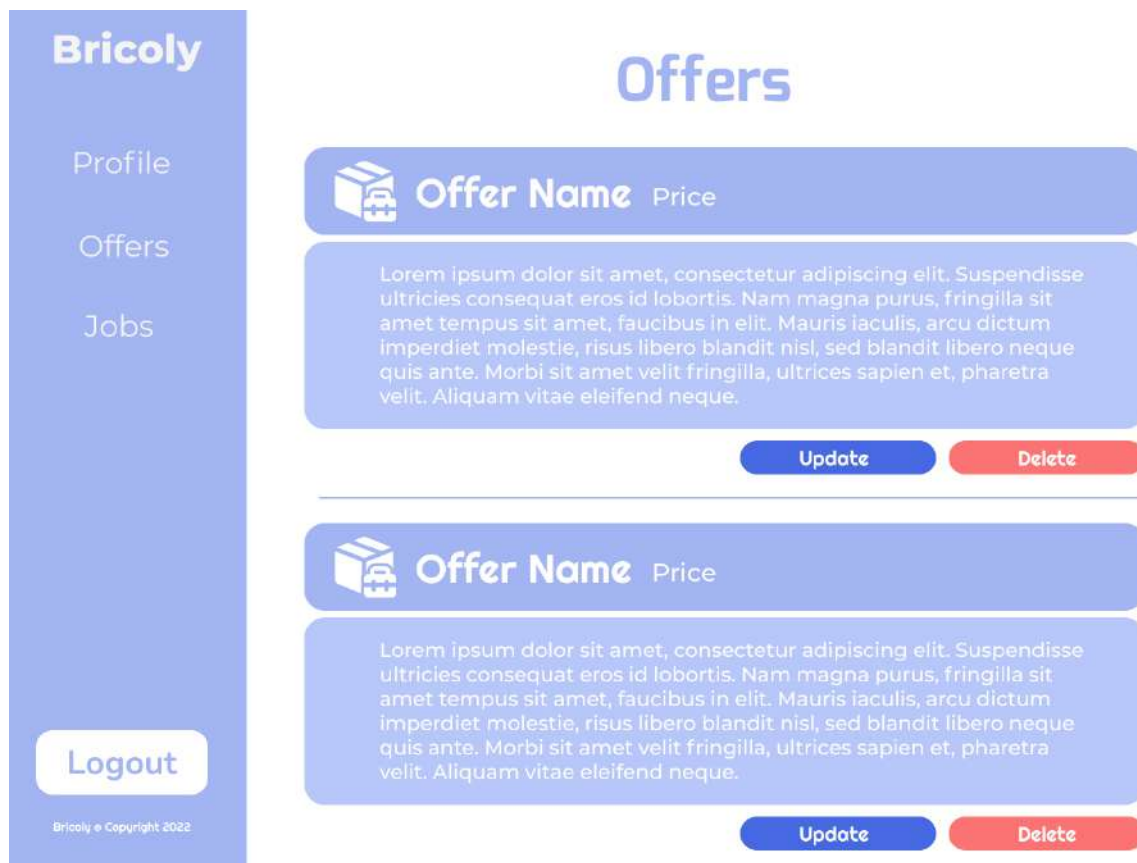
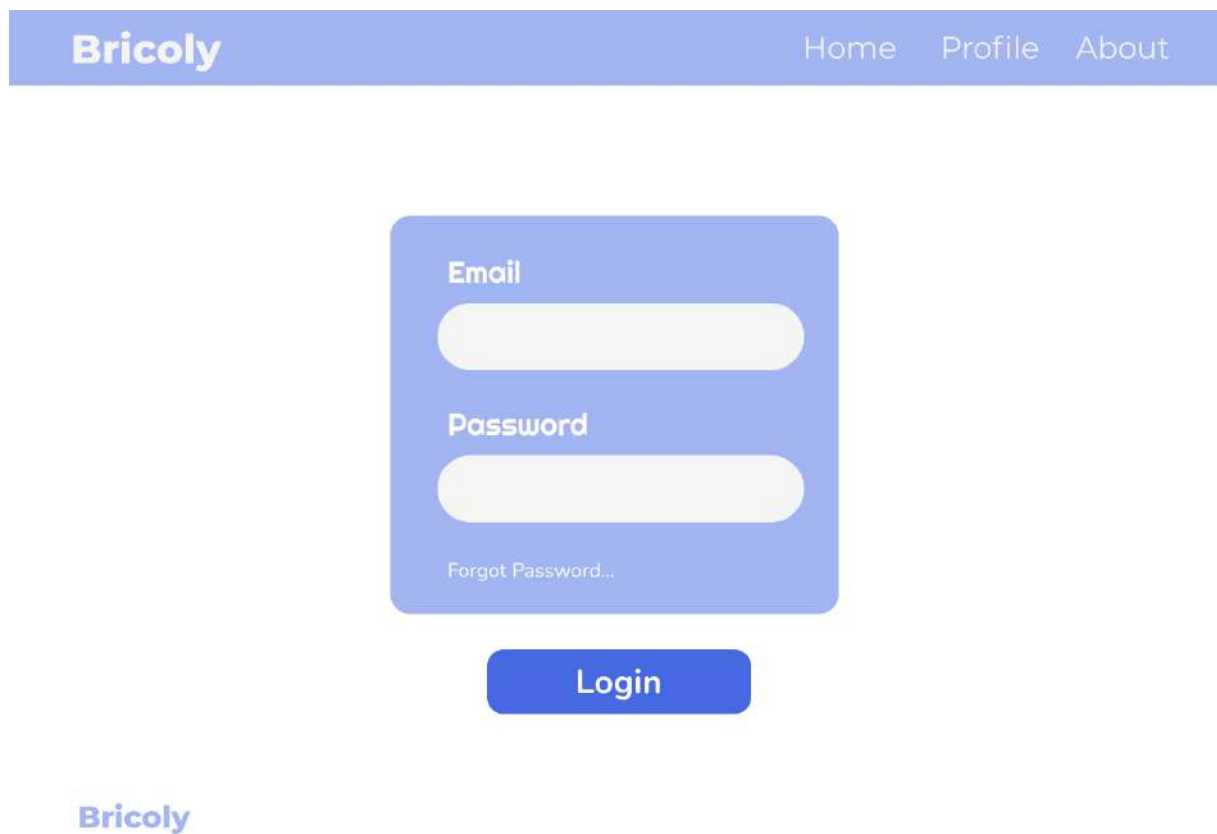


Figure 10 : Liste des offres des prestataires

- **Authentication**

Un utilisateur devrait avoir obligatoirement un compte pour pouvoir accéder à l'application.

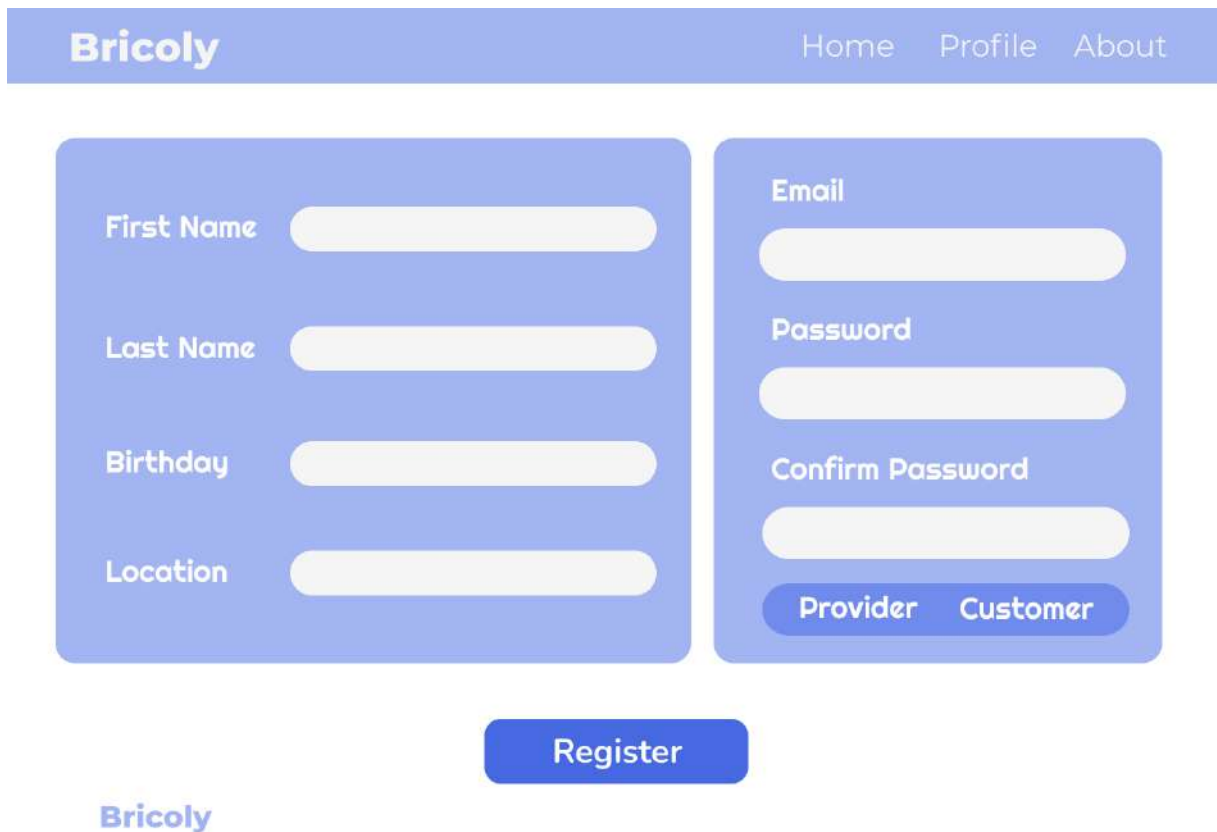


The image shows a web page design for an application named "Bricoly". At the top, there is a blue header bar with the "Bricoly" logo on the left and navigation links "Home", "Profile", and "About" on the right. The main content area has a light gray background. In the center, there is a white rounded rectangle containing the login form. The form has two input fields: "Email" and "Password", each with a light gray rounded input box. Below the "Password" field is a link that says "Forgot Password...". Below the white box is a blue rounded button with the text "Login". At the bottom left of the page, the "Bricoly" logo is repeated in a smaller, blue font.

Figure 11 : Page d'authentification

- **Inscription**

Un utilisateur devrait avoir obligatoirement créé un compte s'il ne possède pas un compte.



The registration form for Bricoly is presented on a light blue background. At the top, a dark blue header bar contains the Bricoly logo on the left and navigation links 'Home', 'Profile', and 'About' on the right. The form itself is divided into two main vertical sections. The left section contains four input fields labeled 'First Name', 'Last Name', 'Birthday', and 'Location'. The right section contains three input fields labeled 'Email', 'Password', and 'Confirm Password'. Below these fields are two radio buttons labeled 'Provider' and 'Customer'. A large blue 'Register' button is centered below the form fields. At the bottom left, the Bricoly logo is repeated.

**Bricoly** Home Profile About

First Name

Last Name

Birthday

Location

Email

Password

Confirm Password

Provider Customer

Register

**Bricoly**

Figure 12 : Inscription

## Conclusion générale et perspectives

Dans ce rapport, nous avons exposé les étapes de réalisation de notre projet qui a pour objectif la conception et le développement d'une application qui relie les prestataires et les demandeurs de services.

Notre travail s'est déroulé sur trois étapes. Nous avons commencé par une étude de l'existant, suivie de la proposition d'une solution adéquate. Dans la deuxième phase, nous avons spécifié les besoins fonctionnels et non fonctionnels que doit respecter l'application suivis par des diagrammes des cas d'utilisation mettant en jeu les acteurs qui interagissent avec le système. La troisième étape concernait la conception de notre application durant laquelle nous avons présenté le diagramme de classe de notre application. Dans la dernière phase, nous avons évoqué les différentes technologies utilisées ainsi que l'implémentation de notre système. Ce projet se situe en effet, dans le cadre du projet de fin de la troisième année du cycle ingénieur en informatique.

Toutefois, pour améliorer notre projet, on pense à réaliser une autre version du système sous forme d'une application mobile afin de faciliter l'accès aux utilisateurs de l'application.

Finalement, ce projet était une véritable expérience de travail en collaboration, qui nous a permis de bien gérer la répartition des tâches et de renforcer l'esprit de partage de connaissances ainsi que la synchronisation de notre travail.



# Bibliographie

## Sites web

[6] <https://www.w3schools.com/>

[7] <https://openclassrooms.com/>

[8] <https://www.icsi.co.uk/>

[9] <https://getbootstrap.com/docs/4.4/getting-started/introduction/>

[10] <https://www.reactnative.dev/>

[11] <https://www.techonthenet.com/>

[12] [https://fr.wikipedia.org/wiki/Conception\\_de\\_site\\_web](https://fr.wikipedia.org/wiki/Conception_de_site_web)

[13] <https://unsplash.com/>