

Table des Matières

I.	Introduction :	3
II.	Présentation du projet	4
1.	Objectifs	4
2.	Description du projet	4
III.	Méthodologie de travail	5
1.	Méthodologie Agile :	5
2.	Pourquoi avoir choisi SCRUM ?	6
IV.	Etude du contexte	7
1.	Identification des acteurs :	7
2.	Identification des besoins fonctionnels :	8
3.	Identification des besoins non fonctionnels :	8
V.	Analyse globale	9
1.	Diagramme de Cas d'utilisation :	9
2.	Modélisation de cas d'utilisation :	10
3.	Diagramme de classe d'analyse global	11
4.	Elaboration du Product Backlog	12
VI.	Etude et choix de l'environnement	14
1.		14
2.		14
VII.	Modélisation de l'architecture logique et physique globale de l'application	15
1.	Architecture logique :	15
2.	Architecture Physique :	16
VIII.	Elaboration des maquettes	17
1.	Maquettes de L'application Web :	17
2.	Maquettes de L'application mobile :	23
3.	Maquettes de L'application Java :	25
IX.	Conclusion :	32

Table des figures

Figure 1 Scrum	7
Figure 2:Diagramme de cas d'utilisation	10
Figure 3:Diagramme de Classes d'analyse	12
Figure 4 Product backlog	13
Figure 5 Diagramme de composant	15
Figure 6 Diagramme de déploiement	Erreur ! Signet non défini.
Figure 7 Maquette de la page d'accueil Web	17
Figure 8 Maquette à Propos Web	18
Figure 9 maquette de page contact web	19
Figure 10 Maquette de la page consulter offre	20
Figure 11 Maquette de la page inscription Client web	21
Figure 12 mquette de la page transporteur	22
Figure 13 maquette de la page authentification Mobile	23
Figure 14 Recherche Offre Mobile	24
Figure 15 Maquette de la page contact Mobile	25
Figure 16 Maquette de la page accueil de l'application Java	25
Figure 17 Maquette de page authentification Java	26
Figure 18 Maquette de page gerer les offres Java	27
Figure 19 Maquette de page modification des offres Java	28
Figure 20 Maquette de la page supprimer offre Java	29
Figure 21 Maquette de la page gerer visite Java	30
Figure 22 Maquette de page Inscription Java	31
Figure 23 Maquette de la page Profil Client	31

I. Introduction :

Afin d'appliquer les méthodologies et les notions enseignées, nous devons réaliser un Projet informatisé, intégré de développement PIDEV.

Celui-ci nous permet à nous, étudiants à ESPRIT, de nous initier à la recherche, d'appliquer les connaissances acquises durant notre scolarité et de favoriser le travail en groupe encadré par nos enseignants.

Afin de comprendre la démarche que nous avons utilisée pour mener ce projet à son terme, notre rapport du sprint 0 se structure de la façon suivante :

Tout d'abord, dans une première partie nous présentons le cadre général de notre projet, c'est-à-dire ce qui existe et ce que notre projet va apporter. Puis dans une seconde partie, nous présentons le travail d'étude et de recherche que nous avons effectué, en commençant par définir le cahier des charges, et par modéliser le projet avec une analyse globale UML. Ensuite dans une troisième partie, nous expliquons par des maquettes descriptives la conception finale de notre projet. Enfin, dans la dernière partie, nous discutons de la concordance de nos résultats par rapport à nos objectifs.

II. Présentation du projet

1. Objectifs

Après discussion entre nous, on a choisi comme sujet « Real Estate Agency », ce projet est une application Desktop, une application mobile et un site web d'une agence immobilière dont l'objectif est la performance des fonctionnalités et la facilité d'usage nous allons concentrer sur ce projet en y ajoutant de nouvelles fonctionnalités vis-à-vis les applications existantes dans le marché.

Notre projet devra être finalisé en respectant les délais.

2. Description du projet

Ce projet réalisé par LEADERS a pour nom «DARNA», Il représente une application d'une agence immobilière qui doit répondre conformément aux demandes du client ainsi qui doit être en grande partie aisément administrable grâce à une interface simple et intuitive par une personne ayant connaissance dans les langages informatiques.

a. Etude de l'existant

Actuellement, nous sommes à la veille de la véritable révolution culturelle, il existe plusieurs moyens permettant de suivre les actualités des immobiliers, ainsi on peut citer quelques sites web.

-LavitrineImmo.com

-Tecnocasa tunisie.tn

-Tayara.tn

-TunisieAnnance.tn

b. Problématique

L'étude de l'existant nous a permis de constater que ces ressources présentent plusieurs lacunes, parmi lesquelles nous pouvons citer :

- Manque d'informations nécessaires pour le client (Propriétaire ou locataire).
- Difficulté de défilement.
- le client peut ne pas voir l'annonce.
- Absence du service transport.

c. Solution proposée

Notre application consiste à favoriser le client (Propriétaire ou locataire) et l'informer par toutes les actualités.

Pour remédier à la « défaillance » nous avons choisi de développer une application informatisée plus efficace qui sert à :

- Une vue structurée et organisée des données pour une meilleure exploitation.
- L'ajout d'une alerte qui permet aux clients d'être à jours de toutes les annonces.
- La souplesse de saisie des informations.

III. Méthodologie de travail

1. Méthodologie Agile :

Les méthodes de développement agile, aussi appelé développement adaptatif, sont des méthodes axées sur les personnes avec pour objectif, la satisfaction du client en s'appuyant sur la réalisation d'un logiciel entièrement fonctionnel tout au long de sa fabrication.

Les méthodes de développement dites « **méthodologie agiles** » visent à réduire le cycle de vie du logiciel.

En développant une version minimale, puis en intégrant les fonctionnalités par un processus itératif basé sur une écoute du client et des tests tout au long du cycle de développement.

Il existe plusieurs Méthodologies agiles dont les principales sont : Méthode SCRUM, Méthode EXtreme Programming (XP), Rational Unified Process (RUP), Feature Driven Development (FDD).

Dans ce cadre, on a choisis de travailler avec la méthode SCRUM.

2. Pourquoi avoir choisi SCRUM ?

SCRUM est aujourd'hui la méthode agile la plus populaire. C'est un processus agile qui vise à produire la plus grande valeur métier dans la durée la plus courte sachant que le métier définit les priorités.

Cette méthode se caractérise par la notion des itérations appelées sprints (maximum 1 mois)

Nous avons choisi, dans notre projet, d'adopter la méthode SCRUM, car il permet de nous organiser en équipe, de créer une ambiance conviviale, et de bien répartir les tâches entre les différents membres de l'équipe, et ainsi de synchroniser le travail. Tout cela dans le but de réussir un projet et arriver à un résultat satisfaisant autant pour l'équipe que pour le client.

Aussi SCRUM a tant de valeurs dont on les cite :

a. Engagement :

Soyez prêt à vous engager sur un objectif.

SCRUM assure aux développeurs l'autorité dont ils ont besoin pour remplir leurs engagements.

b. Focus :

Faites votre travail.

Concentrer tous vos efforts et vos compétences à faire le travail que vous vous êtes engagé à faire. Ne vous inquiétez pas d'autre chose

c. Transparence :

Laisse tous les éléments d'un projet visibles à tous.

d. Respect :

Les individus sont façonnés par leurs antécédents et leur expérience. Il est important de respecter les différentes personnes qui composent une équipe.

e. Courage :

Ayez le courage de vous engager, d'agir, d'être ouvert et d'attendre du respect

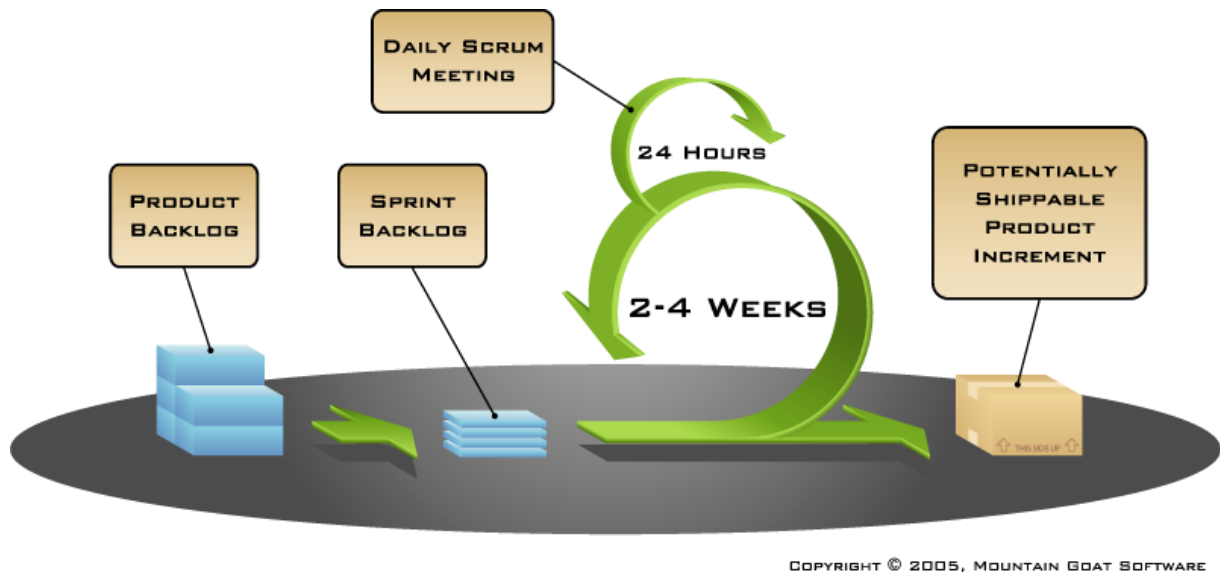


Figure 1 Scrum

La figure précédente décrit le processus général du SCRUM.

SCRUM possède quelques caractéristiques qui sont **ROLES** (Product OWNER, SCRUM MASTER, DEV-TEAM), **MEETINGS** (Sprint Meeting Planning, Daily Scrum, Sprint Review Meeting, Sprint retrospective meeting), **ARTIFACTS** (Product Backlog, Sprint Backlog).

IV. Etude du contexte

1. Identification des acteurs :

a. L'internaute :

C'est l'acteur qui bénéficie d'un accès restreint aux fonctionnalités du système. Il ne pourra que consulter les offres ou s'inscrire pour pouvoir réserver les offres.

b. Le Client :

Le client (locataire) est l'acteur qui bénéficie d'un accès aux fonctionnalités du système. Il sera inscrit sur le site ou sur l'application mobile afin de trouver un logement selon des critères spécifiques.

En effet, un client locataire peut naviguer sur le site, chercher des annonces de location ou de vente et contacter l'agent. Il a aussi la possibilité de proposer, d'évaluer et de commenter une annonce. Ainsi, qu'il aura son espace à lui ou il gère son profil et sa boîte de messagerie. Et s'il y a un problème ou une réclamation ou une remarque à faire à propos de l'application ou le site, il peut contacter l'administrateur directement.

c. Administrateur :

L'administrateur va se charger de la gestion des clients ainsi que la génération des statistiques.

d. Gérant :

Le gérant a plusieurs rôles liés tout d'abord à la gestion des clients (Proposer des offres, Proposer des croquis, Répondre aux questions des clients). Cette gestion inclut les tâches de manipulation des données qui sont présentées dans la figure 2.

2. Identification des besoins fonctionnels :

Les besoins fonctionnels listent les opérations réalisables de notre application. Ce sont des besoins spécifiant un comportement d'entrée / sortie du système. En fait, le système doit établir les charges préliminaires suivantes:

<u>Administration</u> Authentification Mise à jour des offres Répondre aux questions	<u>clients</u> Authentification Rechercher offre Evaluer les offres
<u>internauts</u> S'inscrire rechercher offre	<u>Gérant</u> répondre aux questions proposer les offres

3. Identification des besoins non fonctionnels :

Les besoins opérationnels représentent les besoins non fonctionnels, qui caractérisent le système comme la performance ainsi que la sécurité et l'ergonomie du système.

a. l'ergonomie des interfaces:

- les interfaces doivent être simples et claires.
- la manipulation des interfaces doit être facile à manipuler
- l'application web doit être compatible avec les différents navigateurs
- Les interfaces des applications Android et web doivent être bien organisées du point de vue graphique, le choix des couleurs, et des styles.

b. Robustesse:

- L'application doit permettre le stockage des informations des utilisateurs inscrits, ainsi qu'assurer une bonne gestion d'erreurs.
- la solution doit assurer la rapidité de la recherche des informations

c. Sécurité:

- L'application doit garantir à l'utilisateur connecté l'intégrité et la confidentialité de ses données. Notre système doit également certifier la disponibilité qui s'avère primordiale pour bon fonctionnement.

d. L'efficacité:

- L'application peut être visible dans des réseaux sociaux tels que Facebook, Instagram et twitter.
- L'application doit garantir la Fiabilité et la rapidité des scans ainsi que la flexibilité, l'évolutivité et la réutilisabilité de ses ressources.

V. Analyse globale

1. Diagramme de Cas d'utilisation :

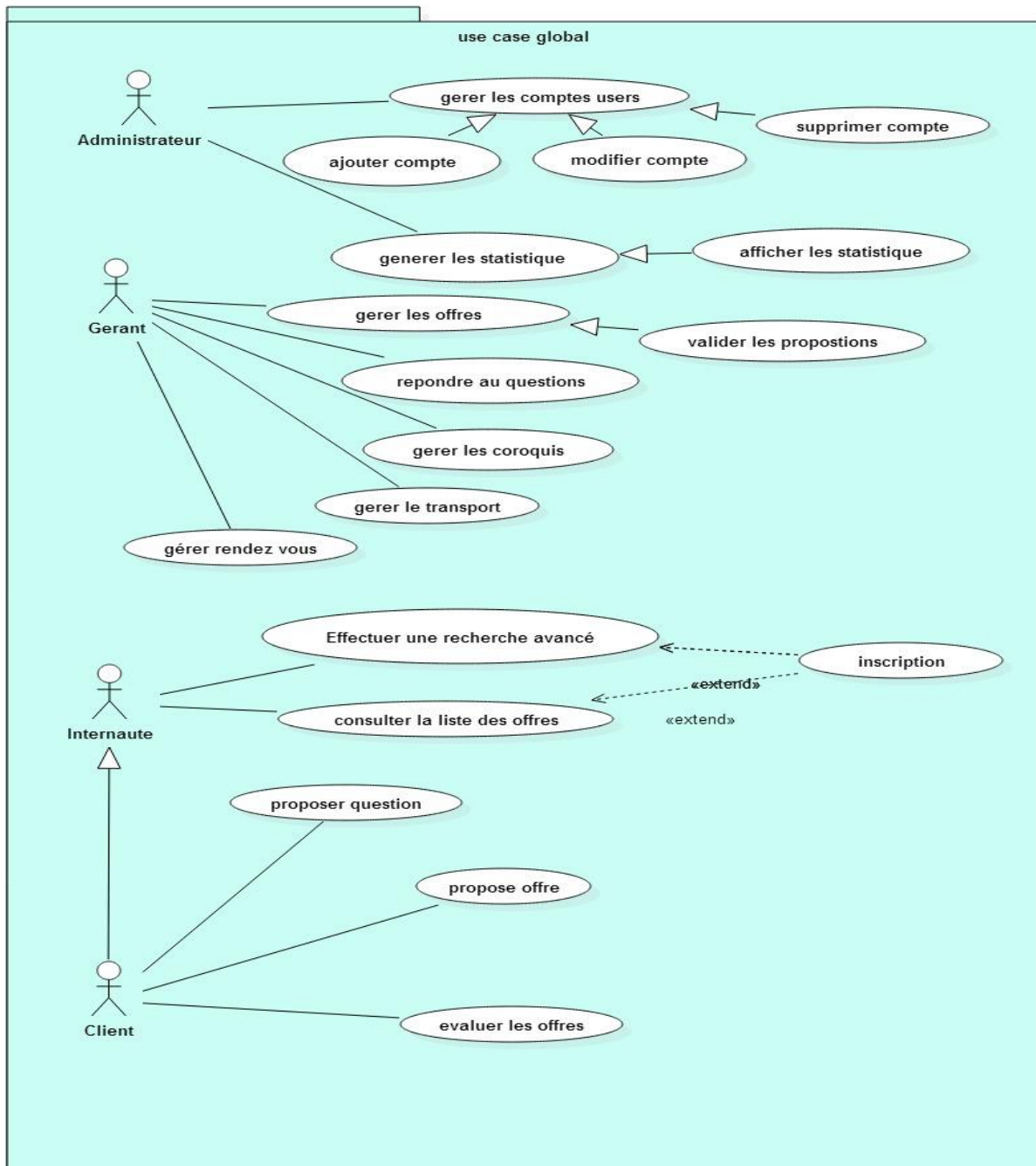


Figure 2:Diagramme de cas d'utilisation

Ce diagramme représente les cas d'utilisation sans montrer les détails de notre système.

2. Modélisation de cas d'utilisation :

S'inscrire : Chaque visiteur pour le site pourra s'inscrire pour devenir membre de l'application ou du site pour pouvoir suivre les nouveautés.

Effectuer une recherche : Même si l'internaute n'est pas inscrit, il a quand même la possibilité de consulter les annonces. Et il peut faire la recherche avancée selon des critères.

Evaluer les offres : chaque client peut attribuer une note à une offre.

Demander visite : Comme étant un client, il pourra demander une visite.

Proposer offre : Comme étant un client, il pourra mettre une annonce.

Proposer Question : Comme étant un client, il pourra poser une question concernant une annonce.

Générer les statistiques: l'administrateur peut générer les statistiques selon le taux de vente dans le site.

Gérer les comptes : l'admin gère les comptes des utilisateurs en ajoutant, supprimant ou en modifiant les comptes.

Contacteur l'administrateur : le client peut contacter l'administrateur pour avoir plus d'information.

Gérer une offre: le gérant peut gérer les annonces (ajout, modification ou suppression, vérifier les propositions des offres).

Gérer un rendez-vous : le gérant peut fixer un rendez-vous.

Gérer transport : le gérant peut désigner et appeler un transporteur au cas où le client va louer ou acheter un immobilier.

Répondre aux questions : le gérant pourra répondre aux questions des clients.

3. Diagramme de classe d'analyse global

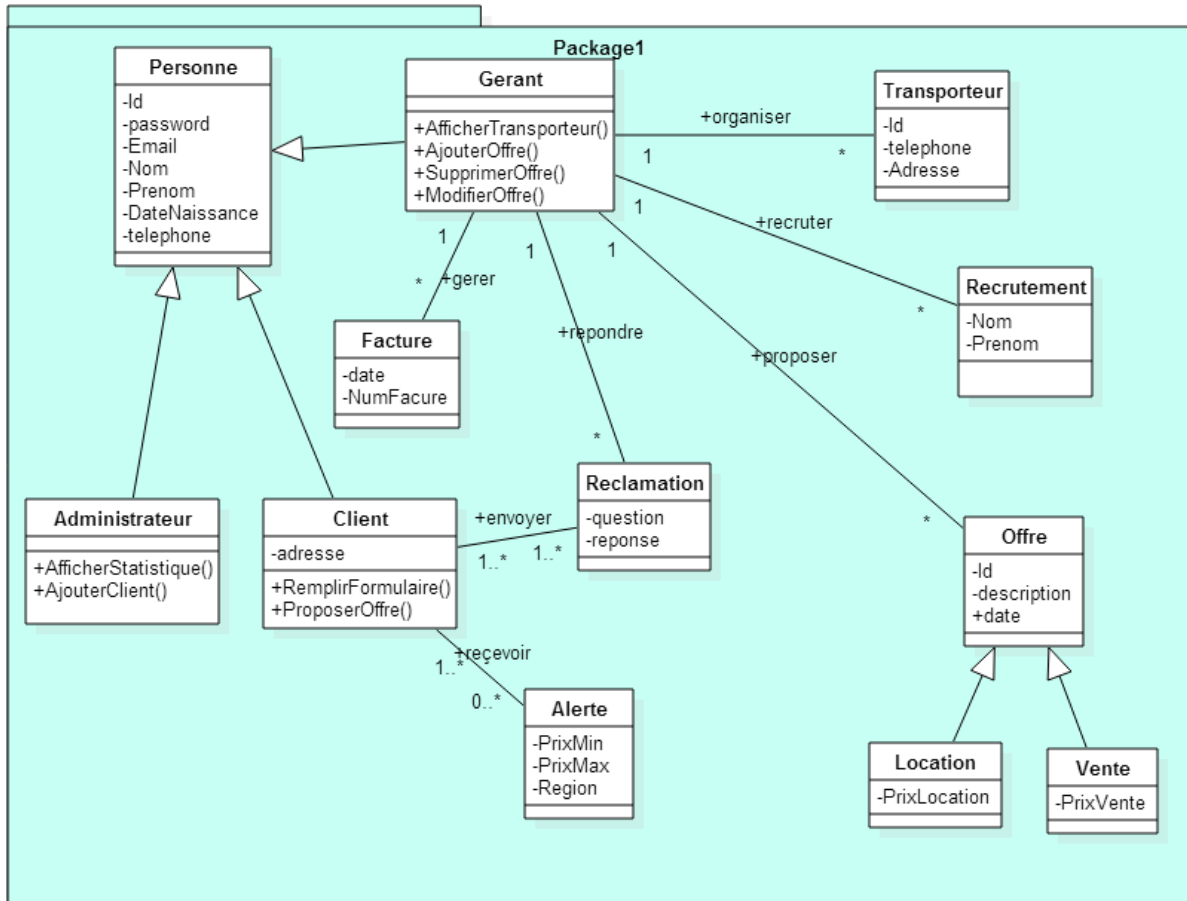


Figure 3:Diagramme de Classes d'analyse

4. Elaboration du Product Backlog

Le Product Backlog recueille des exigences et des "User Stories" *En tant que rôle*, Etabli et maintenu par le Product Owner, chaque élément doit apporter une valeur ajoutée au produit final, Les priorités sont définies par le Product Owner, ils sont revus à chaque Sprint, Les plus importantes sont les plus détaillées.

ID	features	ID	user story	Priority
		100	Entrant que administrateur je souhaite ajouter un compte	100
		101	entrant que administrateur je veux supprimer un compte	80
1	Gestion des comptes utilisateurs	102	En tant que admin je veux modifier un compte et s'authentifier	70
		103	en tant que admin je veux ajouter avis	69
		104	en tant que admin je veux calculer avis	77
2	Gestion des statistiques	105	En tant que admin je veux afficher statistiques	50
		106	en tant que gerant je souhaite verifier proposition des offres et les croquis	54
		107	en tant que client je souhaite evaluer les offres	55
3	Gestion des offres	108	en tant que client je souhaite consulter les information des offres	50
		110	En tant que gerant je veux repondre aux questions des clients	69
4,0	Gestion des questions	111	En tant que client je veux poser des questions	68
		112	en tant que gerant je souhaite generer facture	71
5	Gestion vente	113	en tant que client je souhaite acheter un immobilier	75
		114	en tant que gerant je souhaite generer facture	88
6	Gestion location	115	en tant que client je souhaite louer un immobilier	80
7	Gestion transport	116	en tant que gerant je souhaite organiser le travail des transporteur	30
8	Gestion des alertes	117	en tant que client je souhaite recevoir un formulaire	40

Figure 4 Product backlog

VI. Etude et choix de l'environnement

1. Choix Des Langues et des outils

Dans notre application nous allons utiliser les langages suivants:

Language html, CSS,php,javascript,ajax,jquery pour l'**application web**.

Language Java pour l'**application java**.

Language j2me pour l'**application mobile**.

Comme Outils on a utilisé le staruml pour élaborer le diagramme de classe d'analyse et le diagramme de cas d'utilisation ainsi que le balsamiq pour modeliser les maquettes.



2. Choix De L'outil de Gestion De Configuration

Comme outil de gestion de configuration on a utilisé le Gitlab pour les raisons suivantes:

- Gitlab permet de versionner code source, bien évidemment, mais aussi un moyen de visualisation.
- La possibilité de travailler collaborativement,et concurremment.
- gitlab nous permet de Fournir simplement et élégamment de la documentation.
- Gérer, trier, archiver les bogues.
- c'est un outil d'intégration continue. Pour faire simple, j'ai besoin d'être notifié du résultat.
- La validité des tests augmente avec le nombre d'environnements différents sur lesquels ils sont lancés (Linux, Android, MacOS, Windows etc.).



VII. Modélisation de l'architecture logique et physique globale de l'application

1. Architecture logique :

Sur le plan logique, notre architecture est mise en œuvre suivant le modèle MVC (Modèle Vue Contrôleur) qui s'applique donc au niveau du client. En effet, la spécification d'une architecture à composants métier 4-tiers implique la répartition des composants d'exploitation suivant les responsabilités.

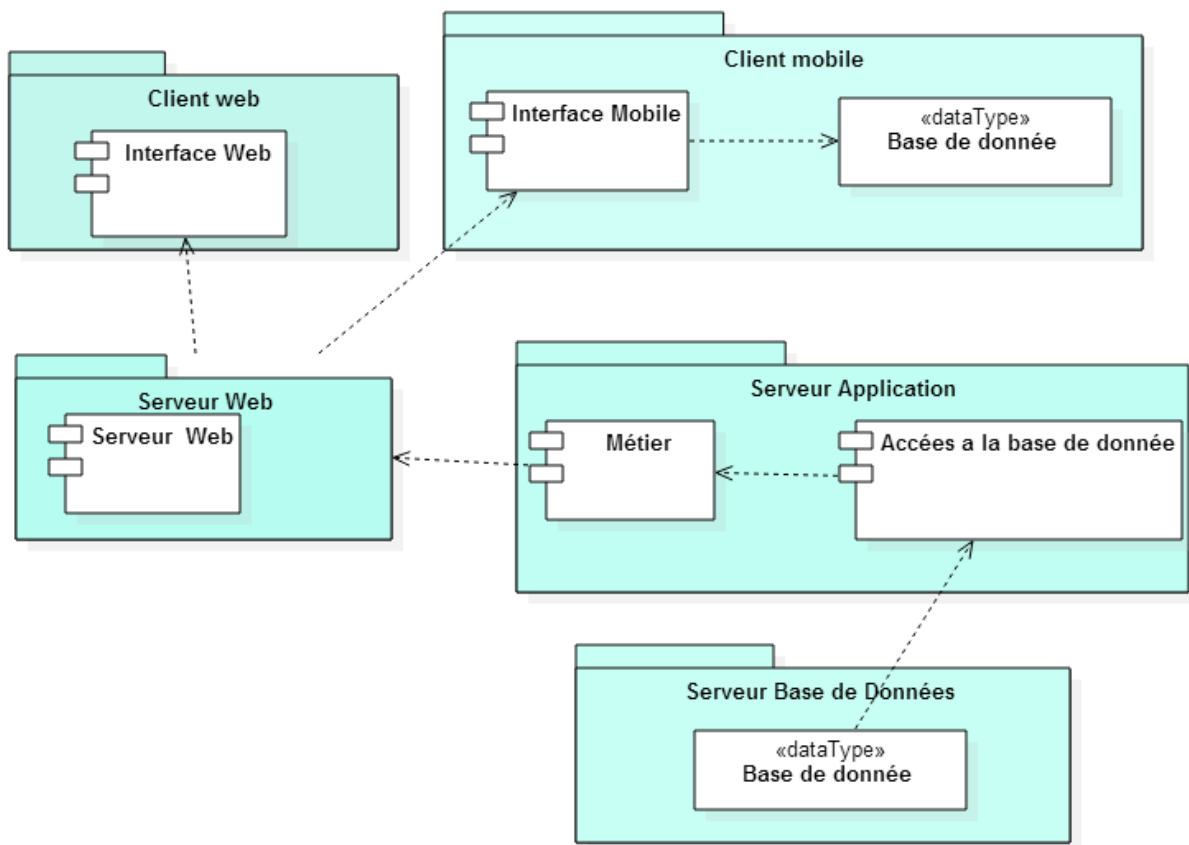


Figure 5 Diagramme de composant

Le diagramme de composant ci-dessus, montre les différents types de composants d'exploitation du système.

2. Architecture Physique :

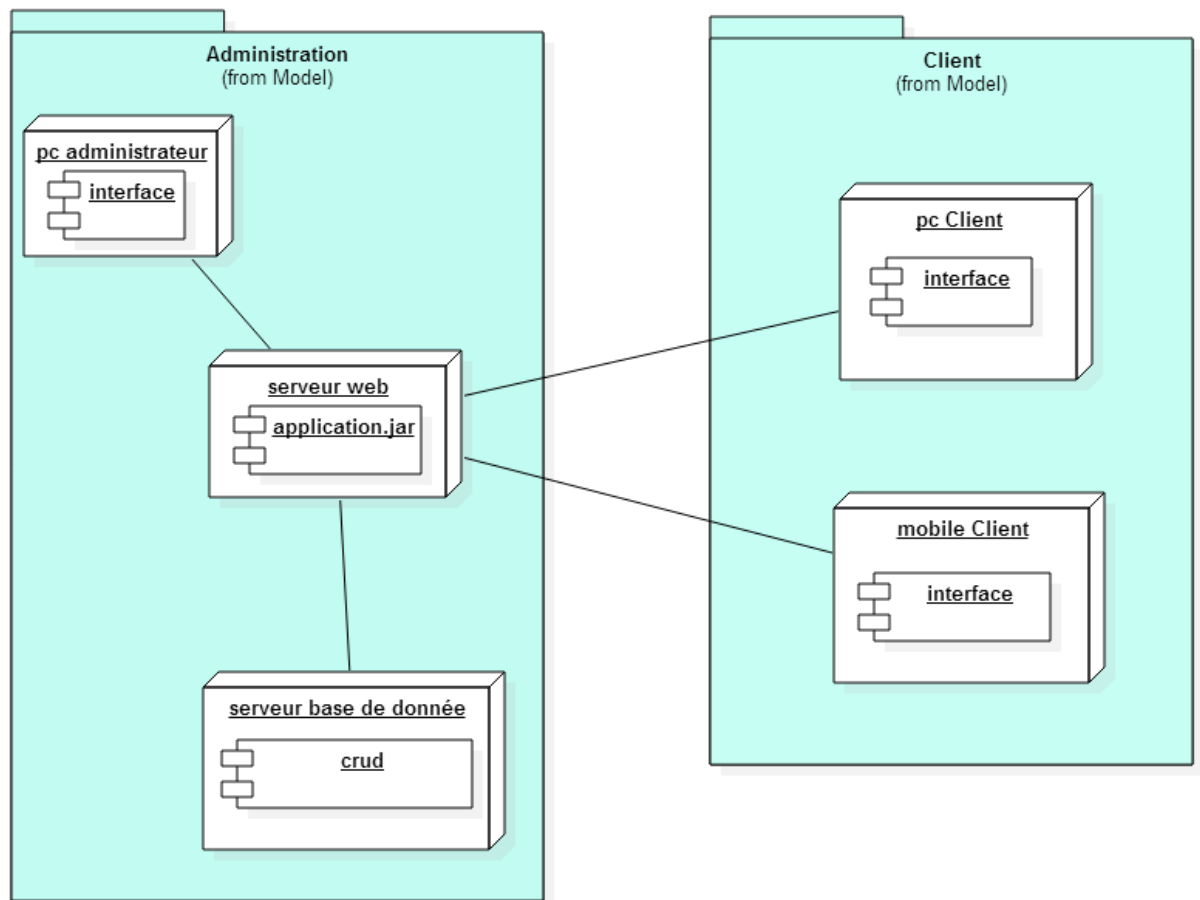


Figure 6 Diagramme de déploiement

VIII. Elaboration des maquettes

1. Maquettes de L'application Web :

a. Page d'Accueil :

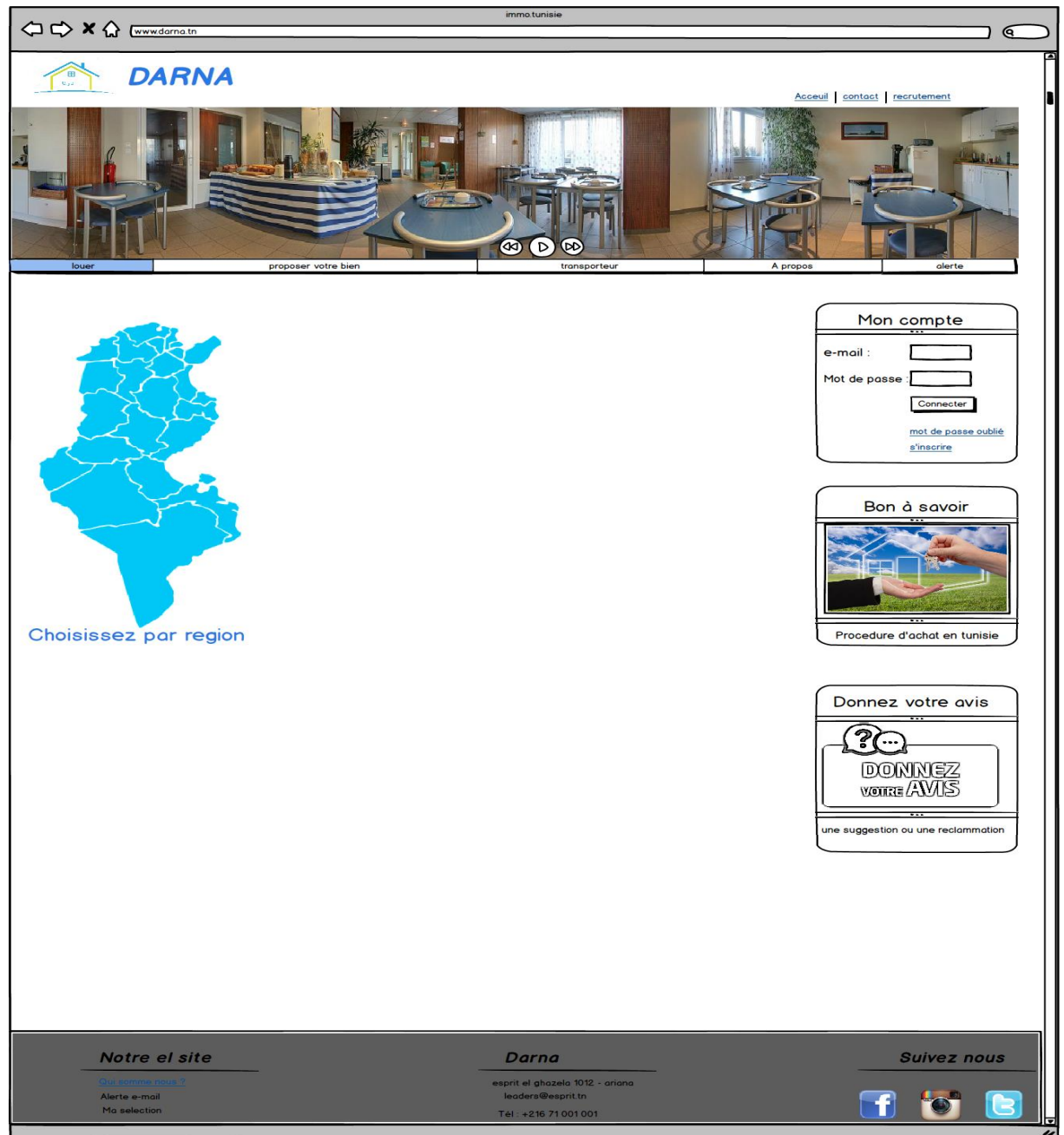


Figure 7 Maquette de la page d'accueil Web

b. Page à propos :

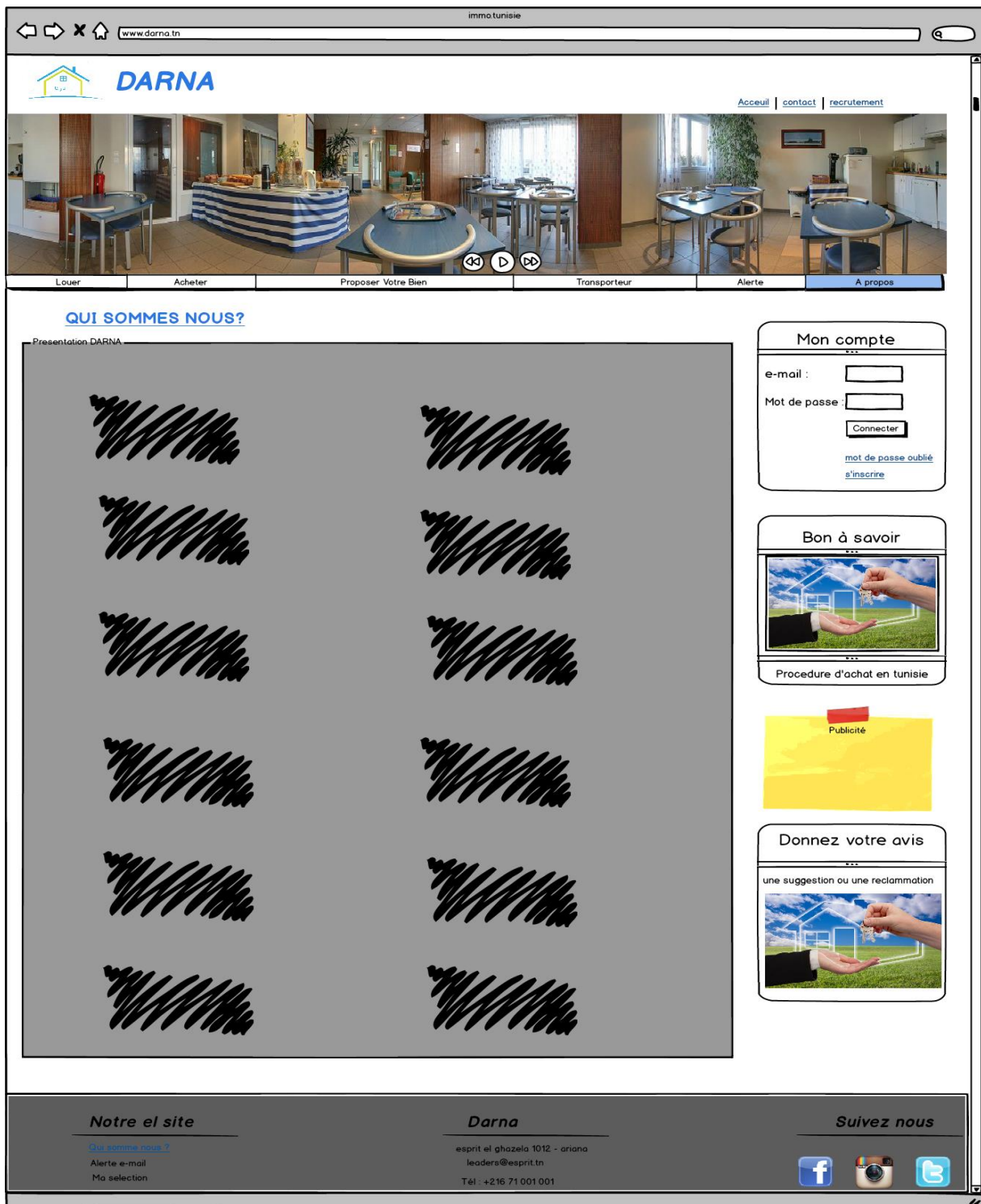


Figure 8 Maquette à Propos Web

c. Page contact

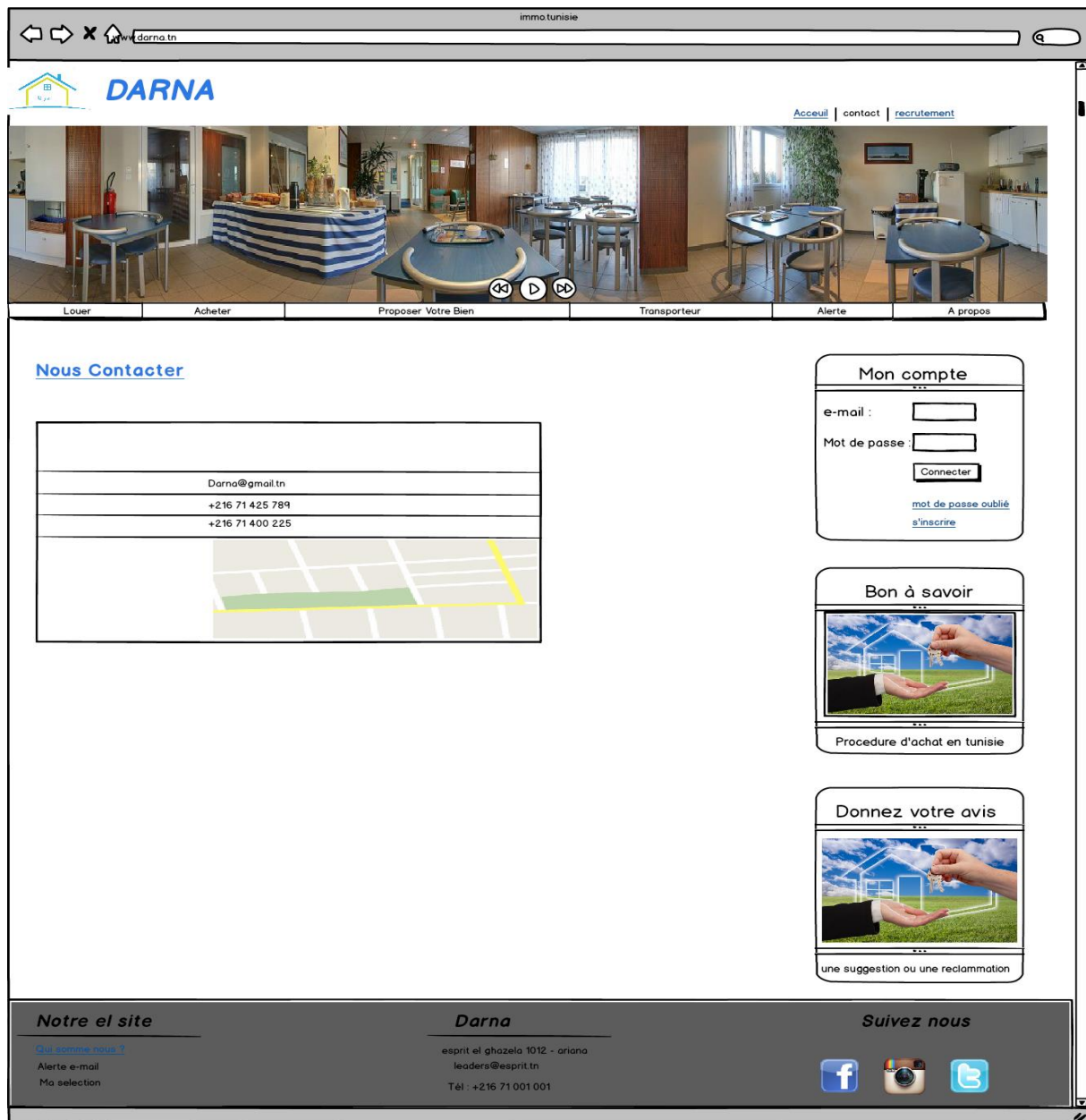


Figure 9 maquette de page contact web

d. consulter offre (exemple louer)

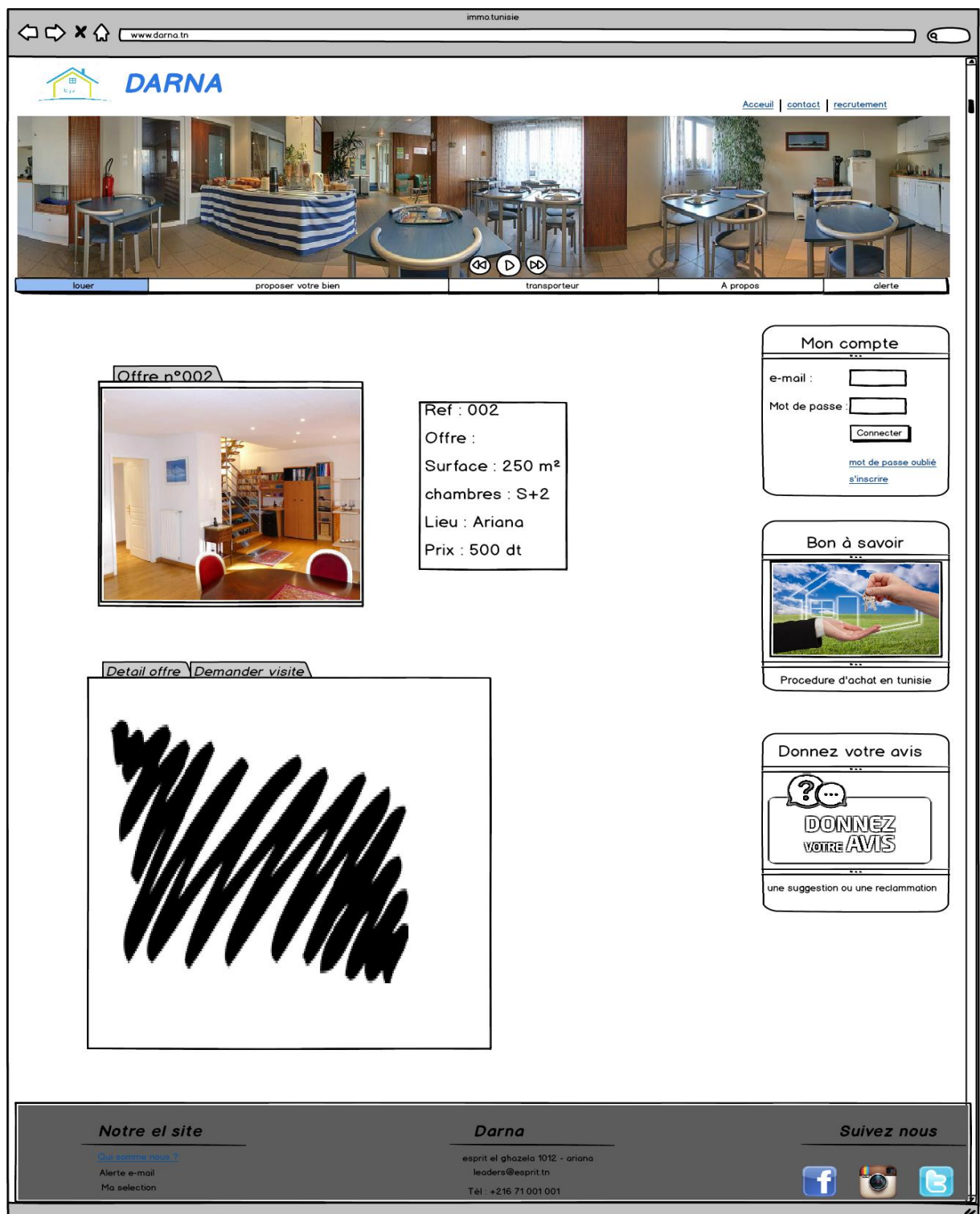


Figure 10 Maquette de la page consulter offre

e. Inscription client :

The image shows a wireframe of a web registration form overlaid on a background image of a modern interior. The browser's address bar at the top displays 'www.darna.tn'. The registration form includes the following fields and elements:

- Prenom ***: A single-line text input field.
- Nom ***: A single-line text input field.
- Adresse Email ***: A single-line text input field.
- Confirmer Adresse ***: A single-line text input field.
- Mot de passe ***: A single-line text input field.
- Confirmer mot de passe ***: A single-line text input field.
- Telephone**: A single-line text input field.
- Enregistrez-vous**: A button with a right-pointing arrow.

Each text input field is represented by a rectangular box. The labels for the first four fields include an asterisk (*) to denote required information.

Figure 11 Maquette de la page inscription Client web

f. Page Transporteur :



Figure 12 mquette de la page transporteur

2. Maquettes de L'application mobile :

a. Authentification



Figure 13 maquette de la page authentification Mobile

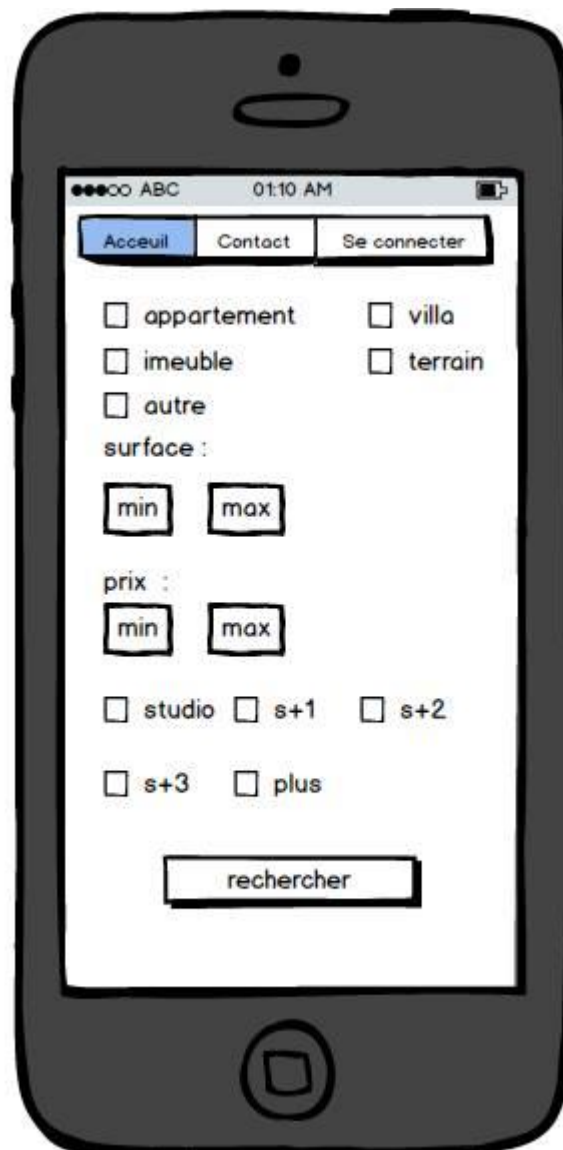
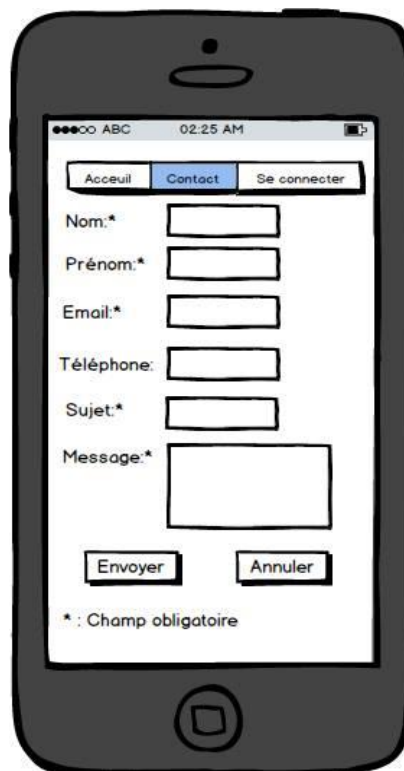
b. Recherche offre

Figure 14 Recherche Offre Mobile

c. Contact

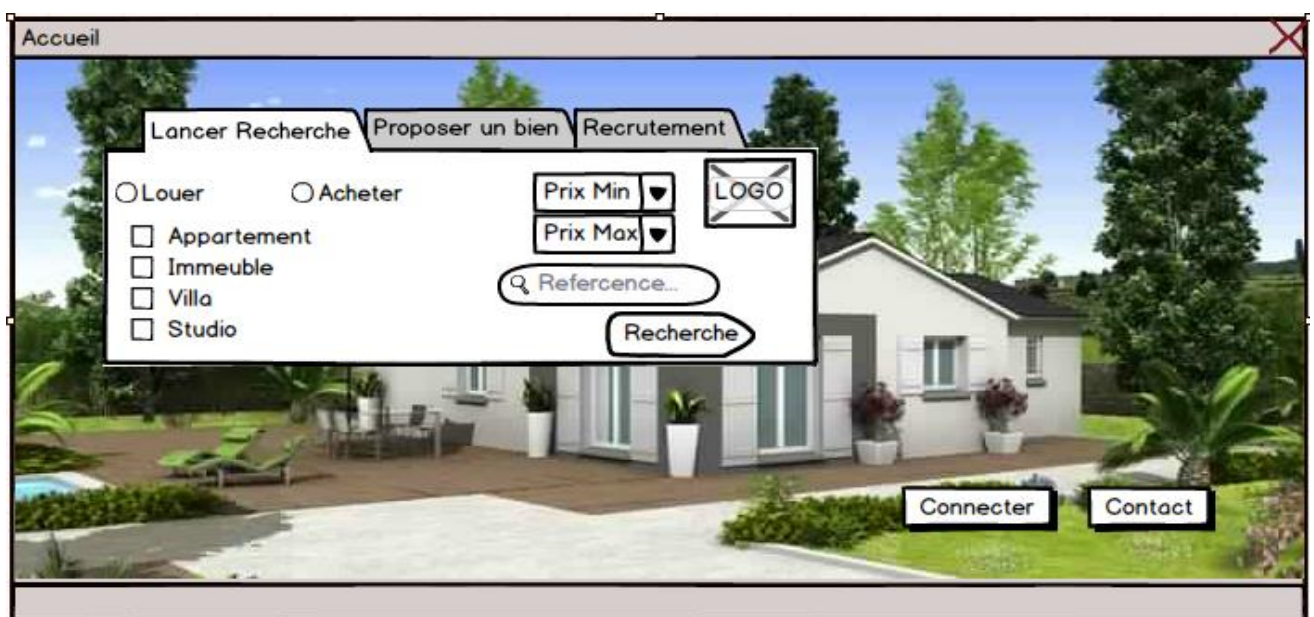


Maquette de la page contact mobile. L'interface est présentée sur un smartphone. En haut, une barre d'état affiche 'ABC' et '02:25 AM'. Le menu de navigation contient 'Accueil', 'Contact' (sélectionné) et 'Se connecter'. Le formulaire de contact comprend des champs obligatoires (marqués d'un astérisque) pour : Nom, Prénom, Email, Téléphone, Sujet, et Message. Les boutons 'Envoyer' et 'Annuler' sont situés en bas du formulaire. Une légende indique : '* : Champ obligatoire'.

Figure 15 Maquette de la page contact Mobile

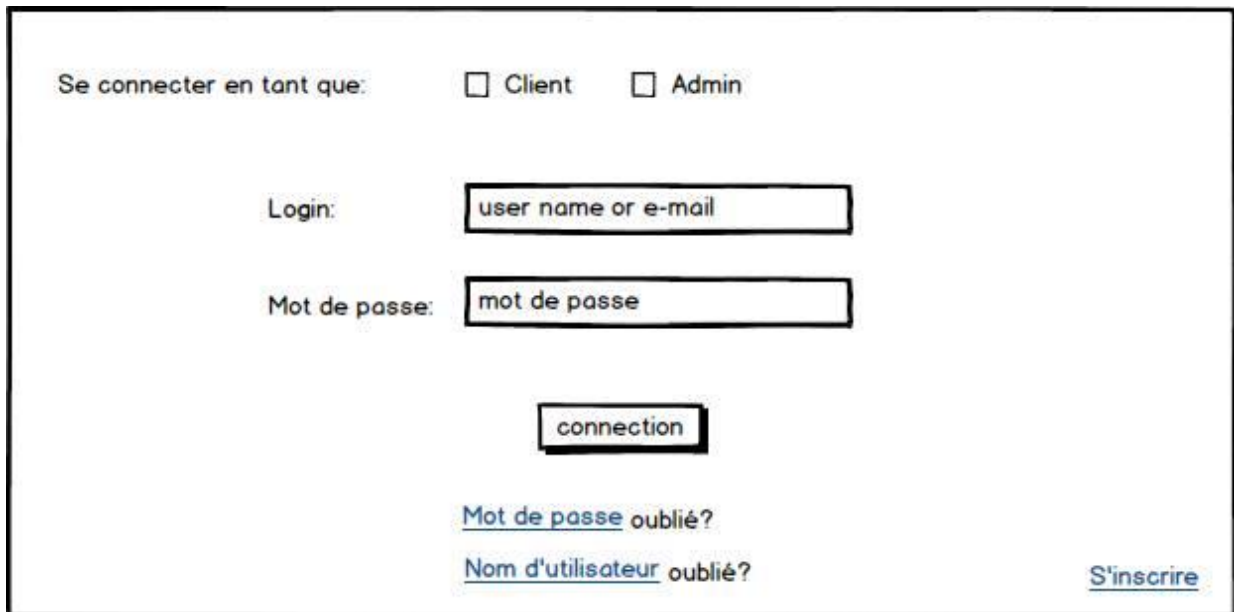
3. Maquettes de L'application Java :

a. Accueil



Maquette de la page accueil de l'application Java. L'interface est présentée dans une fenêtre intitulée 'Accueil'. Le menu de navigation contient 'Lancer Recherche', 'Proposer un bien' et 'Recrutement'. Le formulaire de recherche comprend des options pour 'Louer' et 'Acheter', des cases à cocher pour 'Appartement', 'Immeuble', 'Villa' et 'Studio', des champs pour 'Prix Min' et 'Prix Max', un champ de recherche 'Reference...' avec un bouton 'Recherche', et un bouton 'Recherche' principal. Un bouton 'Connecter' est visible en bas à droite. Le fond de l'interface est une image d'une maison moderne.

Figure 16 Maquette de la page accueil de l'application Java

b. Authentification :

Se connecter en tant que: ☐ Client ☐ Admin

Login:

Mot de passe:

[Mot de passe](#) oublié?

[Nom d'utilisateur](#) oublié?

[S'inscrire](#)

Figure 17 Maquette de page authentification Java

c. Gérer les offres pour l'administrateur

Administrateur : Gérer les offres

Connexion:

Catégorie:

Type d'annonce: ☐ Vente ☐ Location

Titre:

Text de l'annonce:

Prix:

Photo:

Figure 18 Maquette de page gerer les offres Java

d. Modifier les offres pour l'administrateur

Administrateur: Gérer les offres

Ajouter Modifier Supprimer

Rechercher:

Prix:

Disponibilité:

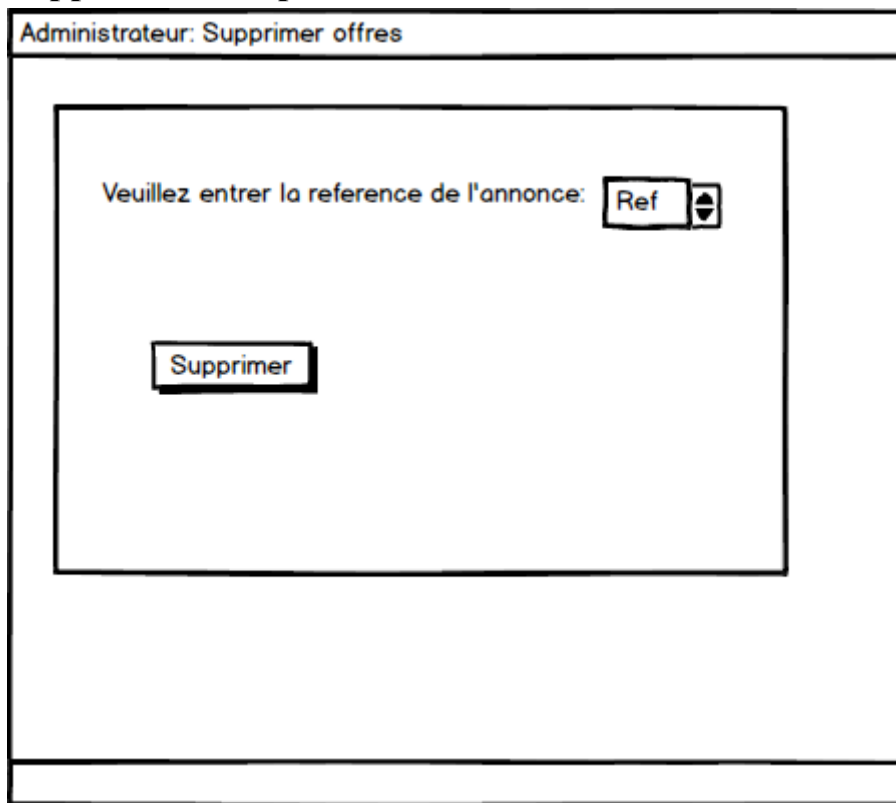
Type d'annonce: ☐ Vente ☐ Location

Ajouter photos:

Enregistrer modification Annuler

Figure 19 Maquette de page modification des offres Java

e. **Supprimer offre pour l'administrateur**



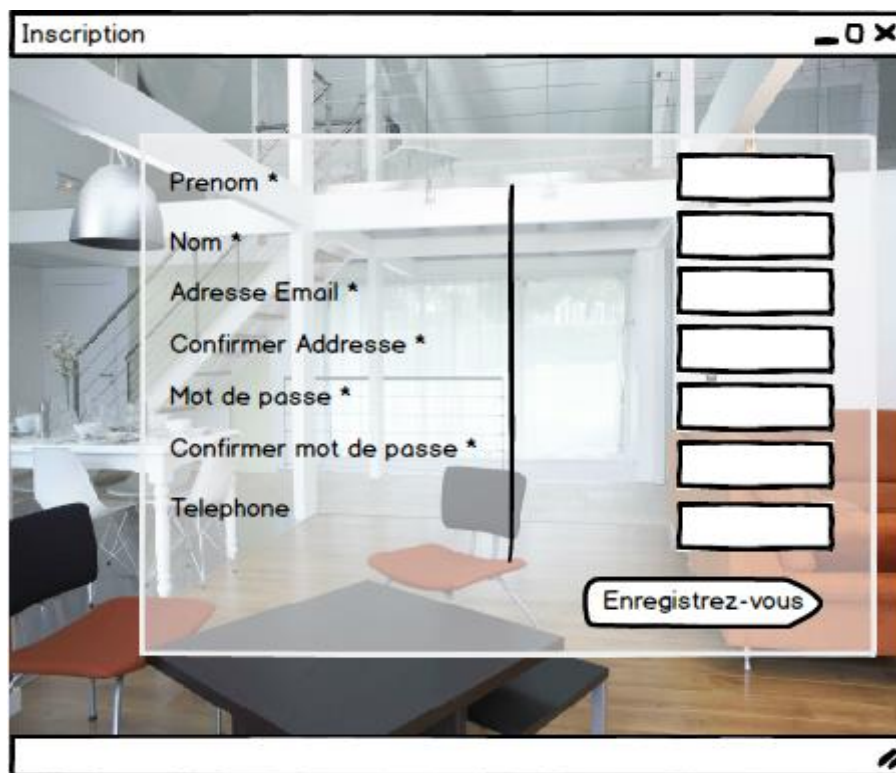
Administrateur: Supprimer offres

Veuillez entrer la reference de l'annonce: Ref

The image shows a Java Swing window titled "Administrateur: Supprimer offres". Inside the window, there is a rectangular panel containing the text "Veuillez entrer la reference de l'annonce:". To the right of this text is a text input field with the placeholder text "Ref" and a small vertical scroll arrow icon on its right side. Below the input field, centered within the panel, is a button labeled "Supprimer".

Figure 20 Maquette de la page supprimer offre Java

g. Formulaire d'inscription pour le client:



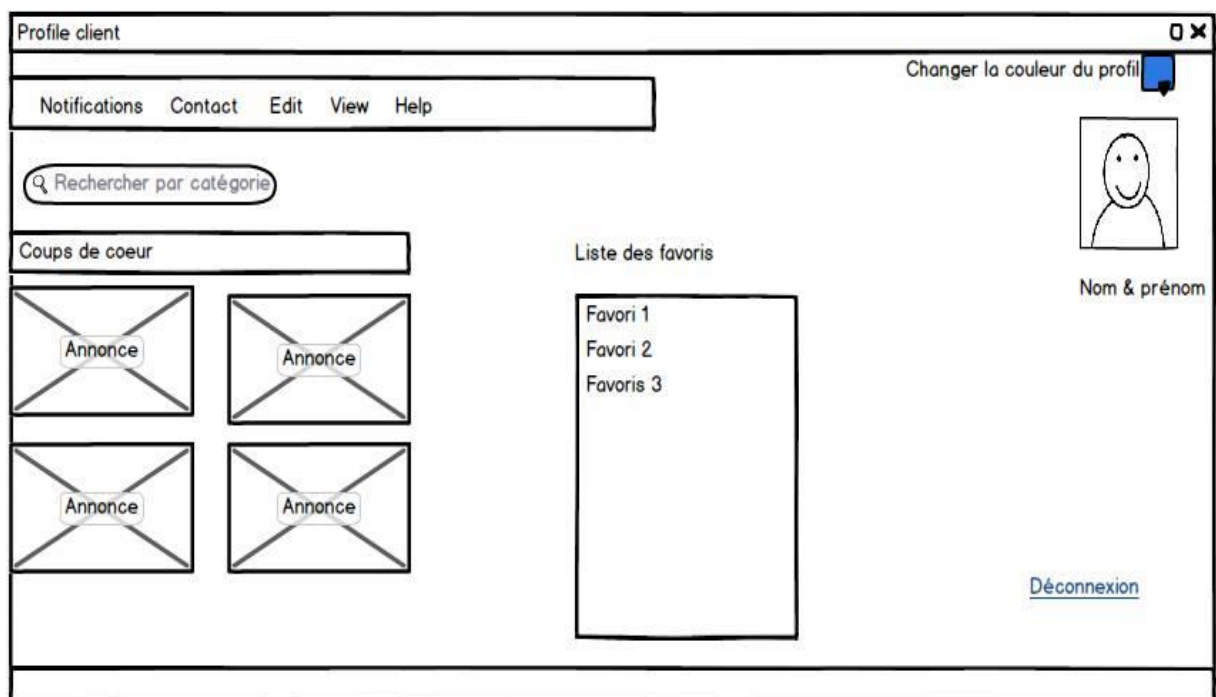
Maquette de la page d'inscription Java. Le formulaire est superposé sur une image de fond d'un intérieur moderne. Les champs de saisie sont :

- Prenom *
- Nom *
- Adresse Email *
- Confirmer Adresse *
- Mot de passe *
- Confirmer mot de passe *
- Telephone

Un bouton "Enregistrez-vous" est situé en bas à droite du formulaire.

Figure 22 Maquette de page Inscription Java

h. Profil client :



Maquette de la page Profil Client. Le titre de la page est "Profile client".

Barre de navigation : Notifications Contact Edit View Help

Recherche : Rechercher par catégorie

Coups de coeur

Liste des favoris

Changer la couleur du profil

Nom & prénom

Déconnexion

Le contenu principal est divisé en deux colonnes. La colonne de gauche contient quatre cartes "Annonce" (chacune avec une croix diagonale). La colonne de droite contient une liste des favoris (Favori 1, Favori 2, Favori 3) et un bouton "Déconnexion".

Figure 23 Maquette de la page Profil Client

IX. Conclusion :

Dans ce rapport du sprint 0, nous avons pu présenter notre projet après une étude de l'existant, élaborer nos choix techniques que nous avons adoptés, ainsi que l'environnement d'implémentation et quelques interfaces de notre application.

Et ce n'est qu'on ayant l'esprit d'équipe et la notion de s'entraider et appliquer la méthode Scrum.

Dans le prochain sprint « Sprint 1 », nous élaborerons le Backlog sprint 1 qui décrira plus en détails les fonctionnalités qui le concerne.