

Rapport du projet de fin d'études

Conception et développement d'une plateforme de
freelance - lanci.tn

Mamoun Boussida Edriss Ben Afia

2020/2021

Dédicaces

En tout premier lieu, je remercie le bon Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.

Je dédie ce modeste travail en signe de respect et de reconnaissance :

À ma très chère mère Moufida

Aucune dédicace très chère maman, ne pourrait exprimer la profondeur des sentiments que j'éprouve pour vous. Tu m'as donné la vie, la tendresse et le courage pour réussir. Puisse Dieu, tout puissant vous combler de santé, de bonheur et vous procurer une longue vie.

À mon très cher père Mounir

Ce travail n'est que le fruit de vos grands efforts et votre immense soutien. Vos sacrifices innombrables et votre dévouement firent pour moi un encouragement. Que Dieu vous protège et vous accorde santé et bonheur.

À mes chères sœurs Maram et Maha

Je ne pourrais jamais imaginer la vie sans vous, vous comptez énormément pour moi, je vous estime beaucoup et je vous aime beaucoup. Je vous souhaite beaucoup de succès, de prospérité et une vie pleine de joie.

À Mes amis

En souvenir des moments heureux passés ensemble, avec mes voeux sincères de réussite, bonheur, santé et de prospérité.

À tous ceux qui ont participé à l'élaboration de ce travail et tous ceux qui nous sont chers, que dieu vous bénisse.

Boussida Mamoun

Dédicaces

Avec beaucoup de sentiments je dédie ce travail à :

Ma mère Houda

S'il y a une personne pour laquelle je ressens généralement un amour inconditionnel, que je remercie pour un millier des choses, c'est notre mère. Merci pour ton amour, ta bienveillance, ton instinct maternel infaillible. Je t'offre ce modeste travail pour te remercier pour ces sacrifices et pour l'affection dont tu m'as toujours entouré.

Mon père Medali

Si je devais revivre ma vie je choisirais toujours d'être ton fils, je choisirais toujours d'être de faire partie de cette famille. Sans ta poussée et tes encouragements constants je n'aurais pas la force et le courage de poursuivre mes rêves. Aucune dédicace ne peut exprimer mes sentiments envers vous Merci infiniment.

Mes deux sœurs Mariem et Malek

Pour leurs encouragements et leur soutien moral. Je leur souhaite tant de réussite dans leurs vies.

Mes chers amis, familles et enseignants

Et pour tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail. Que dieu vous garde.

Benafia Edriss

Remerciements

Nous tenons à remercier Mr. **Kmimech Mourad**, notre encadrant pédagogique à l'ISIMM, pour ses conseils, sa patience et son accueil. Il n'a pas cessé de nous guider et n'a épargné aucun effort pour nous aider à achever ce travail.

Nous tenons aussi à remercier Mr. **Ayech Hamza** (Co-founder & CTO at Halber agency). Votre disponibilité, votre gentillesse et votre grande patience nous ont marqués. Nous vous remercions pour toute l'aide que vous avez apportée.

Nous adressons nos profonds remerciements à Mr. **Hamel Lazhar** (président du jury) et Mme. **Hafiene Nesrine** pour avoir accepté d'évaluer notre modeste travail.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et nos amis qui nous ont toujours soutenu et encouragé au cours de la réalisation de ce projet de fin d'études.

Table des matières

Introduction générale	1
1 Contexte et objectifs du projet	2
1.1 Introduction	2
1.2 Contexte général du projet	2
1.2.1 Cadre du projet	2
1.2.2 Concepts fondamentaux	3
1.3 Etude de l'existant	3
1.3.1 Plateformes existantes	4
1.3.2 Critique de l'existant	8
1.3.3 Solution proposée	10
1.3.4 Objectifs du projet	11
1.4 Méthodologie de travail	12
1.4.1 Méthode classique (Modèle en cascade)	12
1.4.2 Agile Scrum	13
1.4.3 Comparaison des méthodologies et choix	13
1.5 Conclusion	14
2 Analyse et spécifications des besoins	15
2.1 Introduction	15
2.2 L'analyse des besoins	15
2.2.1 Identification des acteurs	15
2.2.2 Les besoins fonctionnels	16
2.2.3 Les besoins non fonctionnels	19
2.3 Spécification des besoins fonctionnels	19
2.3.1 Diagramme de cas d'utilisation général	20
2.3.2 Diagrammes de cas d'utilisation détaillés	21
2.3.3 Diagrammes de séquence système	23
2.4 Backlog du produit et pilotage du projet avec Agile Scrum	26
2.4.1 Backlog du produit	26
2.4.2 Pilotage du projet avec Agile Scrum	28

2.5 Conclusion	28
3 Initialisation du projet	31
3.1 Introduction	31
3.2 Architecture générale de la plateforme	31
3.2.1 Architecture logicielle	31
3.2.2 Architecture physique	33
3.3 Les frameworks de développement	33
3.3.1 Framework front-end	34
3.3.2 Framework back-end	35
3.4 REST API	35
3.5 Persistance des données	36
3.5.1 Choix de la base de données	36
3.5.2 Object-Document Mapping avec Mongoose	36
3.6 Les outils	36
3.6.1 Les outils de conception	36
3.6.2 Les outils de développement	38
3.6.3 Les outils de travail collaboratif	40
3.7 Conclusion	41
4 Mise en œuvre de la release 1	42
4.1 Introduction	42
4.2 Sprint 1 : Visite, création de compte et authentification	42
4.2.1 Sprint Planning	42
4.2.2 Conception	43
4.2.3 Sprint Review	46
4.2.4 Sprint Retrospective	48
4.3 Sprint 2 : Gestion des comptes et des profils	49
4.3.1 Sprint Planning	49
4.3.2 Conception	50
4.3.3 Sprint Review	50
4.3.4 Sprint Retrospective	55
4.4 Conclusion	55
5 Mise en œuvre de la release 2	56
5.1 Introduction	56
5.2 Sprint 3 : Annonce et offres	56
5.2.1 Sprint Planning	56
5.2.2 Sprint Backlog	56
5.2.3 Conception	57
5.2.4 Sprint Review	60

5.2.5 Sprint Retrospective	65
5.3 Conclusion	66
Conclusion générale	67
Webographie	69

Table des figures

1.1	Logo de l'entreprise d'accueil	3
1.2	Page d'accueil de la plateforme Freelance Tunisie	4
1.3	Page d'accueil de la plateforme iLancer	5
1.4	Page d'accueil de la plateforme Upwork	6
1.5	Page d'accueil de la plateforme Toptal	7
1.6	Page d'accueil de la plateforme Freelancer	8
1.7	Logo de notre plateforme lanci.tn	11
1.8	Illustration de modèle de développement en cascade	12
1.9	Cycle de vie de la méthodologie Agile Scrum	13
2.1	Diagramme de cas d'utilisation général	20
2.2	Diagramme de cas d'utilisation de visiteur	21
2.3	Diagramme de cas d'utilisation de l'utilisateur	22
2.4	Diagramme de cas d'utilisation de l'administrateur	23
2.5	Diagramme de séquence "Créer un compte"	24
2.6	Diagramme de séquence "Se connecter"	25
2.7	Diagramme de séquence "Gérer mes annonces"	26
3.1	L'architecture MVC	32
3.2	L'architecture MVC de lanci	33
3.3	L'architecture physique de la plateforme	33
3.4	Diagramme de fonctionnement d'une application React.js	35
3.5	Logo de Draw.io	37
3.6	Logo de Lucidchart	37
3.7	Logo de Visual Paradigm Online Diagrams	38
3.8	Logo de Visual Studio Code	38
3.9	Logo de Postman	39
3.10	Logo de Node.js	39
3.11	Logo de MongoDB Compass	39
3.12	Logo de Bootstrap	40
3.13	Logo de slack	40
3.14	Logo de Jira Software	40

3.15 Logo de Bitbucket	41
4.1 Diagramme MVC détaillé Sprint 1	44
4.2 Diagramme d'interaction "Se connecter"	45
4.3 Interface "Page d'accueil"	46
4.4 Interface "Contact administrateur"	47
4.5 Interface "Inscription"	47
4.6 Interface "Connexion"	48
4.8 Diagramme de séquences détaillé "Modifier profil freelance"	50
4.7 Diagramme MVC détaillé Sprint 2	51
4.9 Interface "Liste freelances"	52
4.10 Interface "Profil"	53
4.11 Interface "Gestion compte"	53
4.12 Interface "Gérer profil freelance" (Freelance)	54
4.13 Interface "Gérer freelances" (Admin)	55
5.1 Diagramme MVC détaillé Sprint 3	58
5.2 Diagramme de séquences détaillé "Ajouter annonce"	59
5.3 Diagramme de séquences détaillé "Proposer une offre"	60
5.4 Interface "Ajouter une annonce"	61
5.5 Interface "Gestion annonces"	61
5.6 Interface "Détails annonce"	62
5.7 Interface "Consulter annonces"	62
5.8 Interface "Proposer une offre"	63
5.9 Interface "Mes offres (Freelance)"	63
5.10 Interface "Offres sur mission" (Client)	64
5.11 Interface "Gérer annonces" (Admin)	65

Liste des tableaux

1.1	Comparaison entre différentes plateformes	9
2.1	Déscription du cas d'utilisation "Créer un compte"	24
2.2	Déscription du cas d'utilisation "Se connecter"	25
2.3	Déscription du cas d'utilisation "Gérer mes annonces"	27
2.4	Backlog du produit	29
2.5	Planification des releases et des sprints	30
4.1	Backlog du sprint 1	43
4.2	Backlog du sprint 2	49
5.1	Backlog du sprint 3	57

Acronymes

API Application Programming Interface. 35, 38

CDD Contrat à durée déterminée. 3, 4

CDI Contrat à durée indéterminée. 3, 4

CI/CD Continuous Integration & Continus Delivery. 41

CV Curriculum Vitae. 4, 55

JWT JSON Web Token. 50, 59, 60

MVC Model View Controller. 31, 32, 37, 43, 50, 57

ODM Object Document Mapper. 32, 36, 50, 59, 60

REST Representational State Transfer. 35

RWD Responsive Web Design. 19

SaaS Software as a Service. 41

SRP Separation of Responsibility Principle. 31

Introduction générale

Dès nos jours le travail libre (freelance) est devenu une source pour gagner sa vie. C'est un métier qui possède plusieurs avantages : quand on travaille de chez soi, la notion de flexibilité prend tout son sens. Vous obtenez la liberté de constituer vous-même votre emploi du temps et pouvez ainsi choisir vos horaires de travail.

C'est pourquoi nous pensons à créer un environnement virtuel qui permet de regrouper les prestataires qui cherchent à travailler indépendamment.

Dans ce contexte se dirige notre projet tant qu'il consiste à développer une plateforme web de freelance. Les freelances pourront trouver des annonces publiées par des différents clients et porteurs d'idées et ils les accompliront afin d'être payés.

Pour ce rapport nous avons choisi le cheminement suivant :

Le premier chapitre sous le nom « Contexte et objectifs du projet » où nous présenterons le cadre du projet ainsi que les concepts fondamentaux. Puis, on fait une étude et comparative de l'existant et nous terminons par la méthodologie du travail avec une justification du choix.

Le deuxième chapitre, intitulé « Analyse et spécifications des besoins », présente les besoins fonctionnels, non fonctionnels et techniques, ainsi que les diagrammes de cas d'utilisation et les diagrammes de séquences système.

Le troisième chapitre « Initialisation du projet » qui sera dédié à expliquer l'architecture générale de notre application, présenter les frameworks du développement et les outils utilisés et, à la fin, une partie aura pour objectif de présenter la manière de persistance des données.

Le deux chapitres 4 et 5 : chaque chapitre est réservée pour une release et chaque release sera composée des sprints, un sprint sera divisé en plusieurs parties afin de bien expliquer les parties de conception et de réalisation.

Chapitre 1

Contexte et objectifs du projet

1.1 Introduction

Ce chapitre est consacré à l'obtention d'une vue d'ensemble sur le projet. On va commencer par le contexte général. Ensuite, nous analysons quelques solutions existantes sur le marché en citant leurs avantages ainsi que leurs points faibles. Enfin, on va présenter notre solution qui va confronter les problèmes dégagés de la partie précédente et la méthodologie de travail utilisée.

1.2 Contexte général du projet

Dans cette section, on présente le cadre du projet ainsi que les définitions des certaines notions afin de mieux comprendre le contexte de notre projet.

1.2.1 Cadre du projet

Contexte Académique

Ce travail se déroule dans le cadre du projet de fin d'études à l'Institut Supérieur d'Informatique et de Mathématiques de Monastir (ISIMM) de l'université de Monastir afin d'obtenir le diplôme national de licence appliquée en informatique (systèmes informatiques et logiciels).

Entreprise d'accueil

Le projet est réalisé durant un stage de 4 mois au sein de **Halber Agency**. C'est une entreprise de service (agence digitale) basée à Monastir, et plus précisément, appartement 11, Immeuble Hadhri, Rue d'Alger. Son logo est illustré par la figure 1.1. Cette entreprise crée des solutions informatiques innovantes en offrant les services de conception et de réalisation des applications et des designs (interfaces

utilisateur) vifs et modernes pour web et mobile et avec un accent particulier sur la qualité de l'expérience utilisateur.

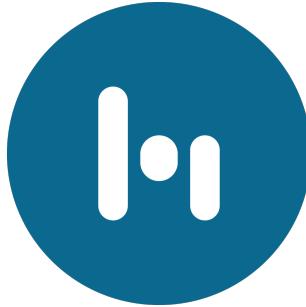


FIGURE 1.1 – Logo de l'entreprise d'accueil

1.2.2 Concepts fondamentaux

Le "Freelance"

Travailler en freelance signifie travailler en tant qu'indépendant. On parle de statut de travailleur indépendant (ou autonome). Un travailleur indépendant exerce des activités en son propre nom. En d'autres termes, c'est un collaborateur qui n'a pas de contrat à long terme comme les salariés, par ex. CDD ou CDI, mais réalise son activité professionnelle sous la forme de missions qu'il facture à ses clients. Il est autonome en termes de gestion organisationnelle, de sélection des clients et de tarification des services.

La "Gig Economy"

L'économie à la tâche (de l'anglais gig economy) désigne aussi sous le nom d'économie à la demande ou économie des petits boulots. C'est un système basé sur des emplois flexibles, temporaires ou indépendants. Cette économie est basée sur le travail à la tâche ou aussi sur le travail à la pièce. Des plateformes en ligne ont essayé depuis quelques années d'appliquer ce genre d'économie. Et c'est ce qu'on va discuter à la section suivante.

1.3 Etude de l'existant

Une bonne compréhension du marché de travail indépendant au monde et surtout en Tunisie est essentielle pour déterminer les points forts et faibles des futurs compétiteurs et pour tirer les défis qu'on va éventuellement rencontrer durant les phases de réalisation et de lancement du projet.

CHAPITRE 1. CONTEXTE ET OBJECTIFS DU PROJET

1.3.1 Plateformes existantes

Plateformes en Tunisie

Freelance Tunisie : est une plateforme de travail indépendant et d'emploi en Tunisie comme l'indique son nom. Fondée en 2017, elle offre un espace où les donneurs d'ordre peuvent poster des missions en plusieurs catégories en technologies de l'information telle que l'ingénierie, le développement des logiciels et le design. Ces missions peuvent être de type freelance ou autre (CDI, CDD, Stage, etc). Les demandes des candidats seront transférés par e-mail vers les recruteurs. La figure 1.2 désigne la page d'accueil de la plateforme.

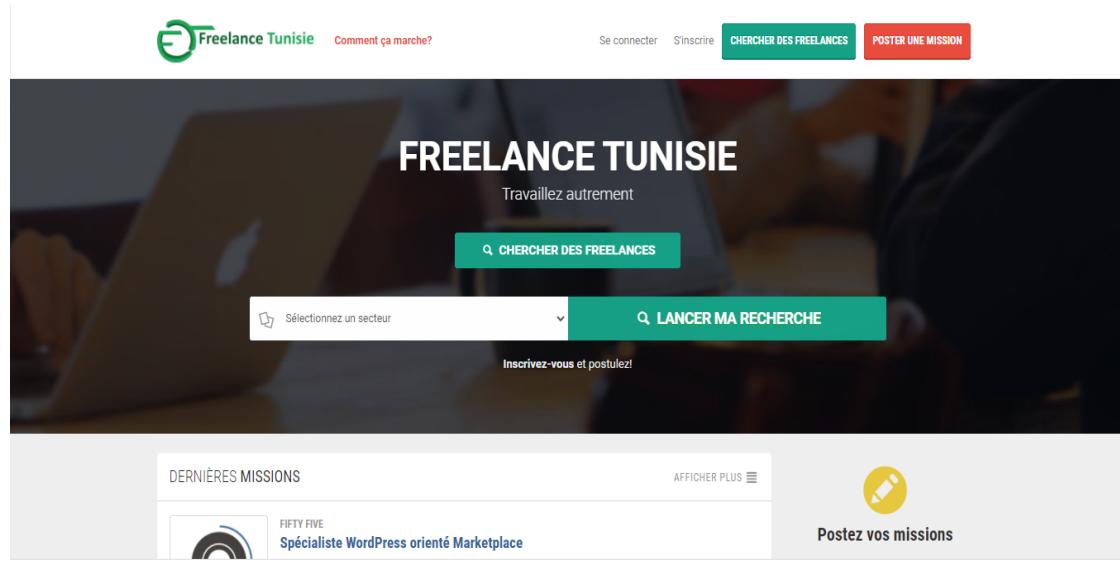


FIGURE 1.2 – Page d'accueil de la plateforme Freelance Tunisie

Le freelance/demandeur d'emploi peut former son profil et ajouter ses compétences ainsi qu'émettre son CV en ligne. Le service est entièrement gratuit et il va toujours rester sans frais comme son équipe l'a mentionné [6].

iLancer.tn : est une plateforme tunisienne de vente des services et la figure 1.3 désigne sa page d'accueil.

CHAPITRE 1. CONTEXTE ET OBJECTIFS DU PROJET

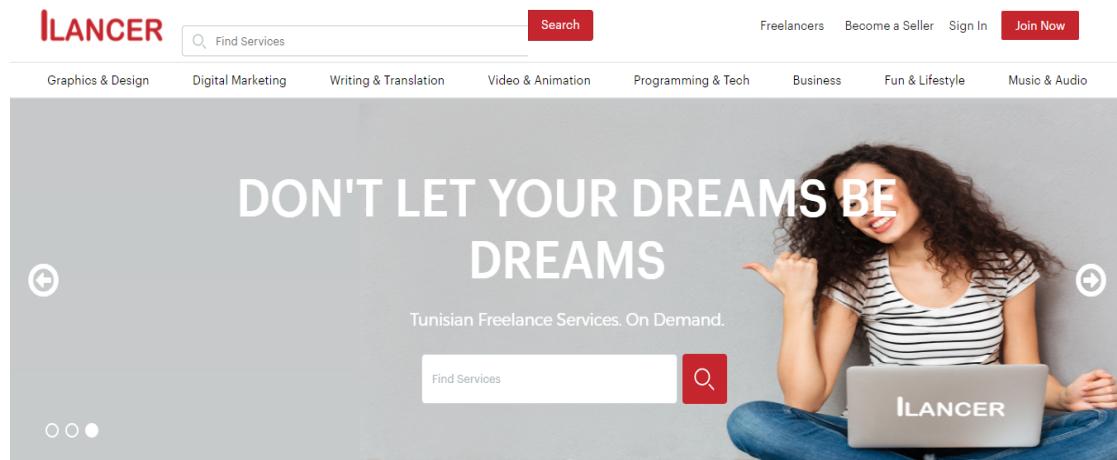


FIGURE 1.3 – Page d'accueil de la plateforme iLancer

L'idée principale consiste à offrir, en premier lieu, un espace aux travailleurs indépendants (vendeurs) où ils peuvent poster des prestations conformes à leurs compétences. Et en second lieu, un espace pour les acheteurs où il peuvent parcourir les services postés et sélectionner celui qui leur convient le mieux [10].

Plateformes internationales

Upwork : auparavant Elance-oDesk, est une plateforme américaine de recrutement (marché en ligne) spécialisée en travail indépendant, totalement en anglais, qui met en relation des freelances et des clients dans 180 pays [4]. La figure 1.4 désigne la page d'accueil de cette plateforme.

CHAPITRE 1. CONTEXTE ET OBJECTIFS DU PROJET

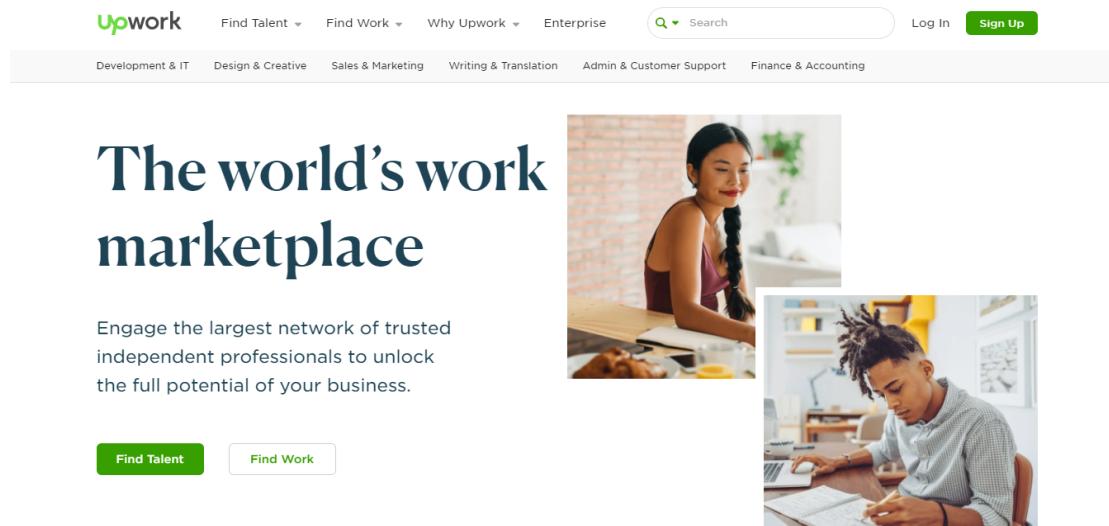


FIGURE 1.4 – Page d'accueil de la plateforme Upwork

Upwork est reconnue par la qualité élevée de ses freelances grâce à leur procédure de sélection (screening process) qui a pour but de mesurer l'aptitude professionnelle des nouveaux travailleurs lors de la création d'un nouveau compte sur la plateforme.

Toptal : est une société mondiale à distance fondée par Taso Du Val et Breanden Beneschott en 2010. Elle fournit une plateforme indépendante, connectant les entreprises avec des ingénieurs en logiciel, des concepteurs, des experts financiers, des chefs de produit et des chefs de projet. L'entreprise n'a pas de siège social. La figure 1.5 désigne la page d'accueil de Toptal.

CHAPITRE 1. CONTEXTE ET OBJECTIFS DU PROJET

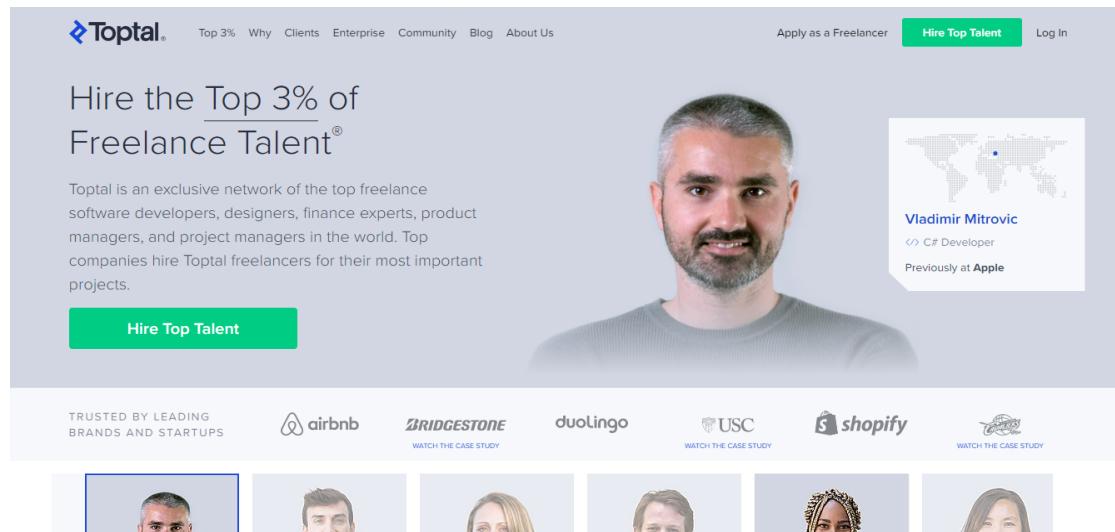


FIGURE 1.5 – Page d'accueil de la plateforme Toptal

De la même manière qu'Upwork, Toptal suit une procédure de sélection stricte afin de garder une bonne qualité de ces talents. Cette procédure [20] est divisée sur 5 phases :

1. Langue et personnalité (Language and Personality),
2. Examen approfondi des compétences (In-Depth Skill Review),
3. Sélection en direct (Live Screening),
4. Projets de test (Test Projects),
5. L'excellence continue (Continued Excellence).

Freelancer : est une plateforme australienne du marché indépendant qui permet aux employeurs potentiels de publier des emplois que les prestataires peuvent ensuite soumissionner. Fondé en 2009, son siège social est situé à Sydney, mais elle possède également des bureaux à Vancouver, Londres, Buenos Aires, Manille et Jakarta. La figure 1.6 désigne la page d'accueil de cette plateforme.

CHAPITRE 1. CONTEXTE ET OBJECTIFS DU PROJET

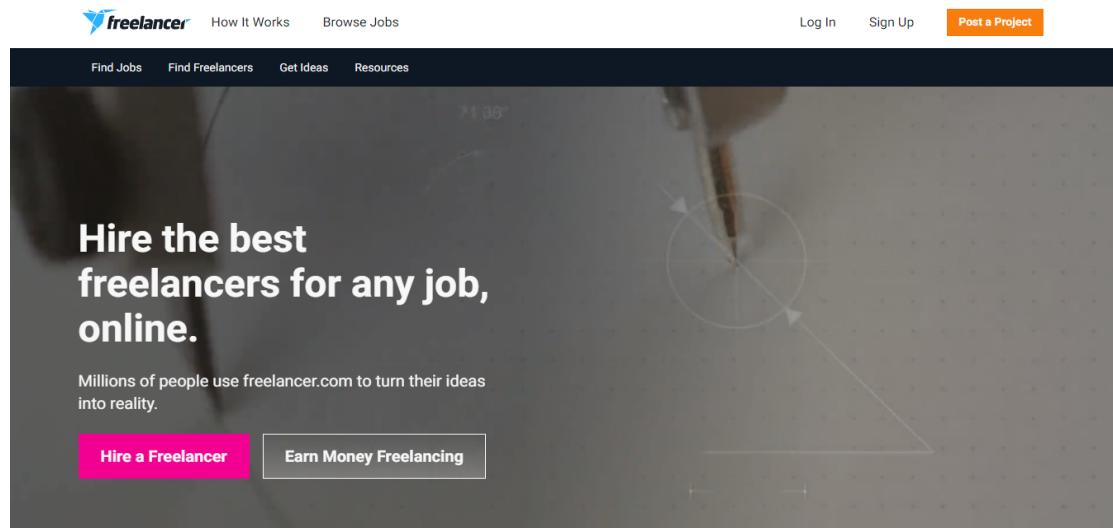


FIGURE 1.6 – Page d'accueil de la plateforme Freelancer

Freelancer.com est un acteur international largement implanté sur le marché mondial des prestations. La plateforme permet de mettre en relation les entreprises avec des freelances qui exercent plusieurs domaines. On peut citer : le développement des logiciels, l'ingénierie, la saisie de données, la rédaction, le design, le marketing, la comptabilité, la traduction et plusieurs d'autres. Les avantages de freelancer.com sont multiples, dont on peut citer : La variété des catégories, l'accessibilité et la facilité d'utilisation. Tout comme Upwork et Toptal, Freelancer.com évolue au niveau mondial, ce qui en fait une plateforme de freelance parfaitement adaptée pour les projets à distance en anglais.

1.3.2 Critique de l'existant

Tableau comparatif

Le tableau 1.1 résume les points forts et faibles des différentes plateformes de freelance que nous avons étudiées.

Critique de l'existant

Chez Upwork et Toptal, les nouveaux arrivants vont rencontrer un grand défi et c'est de faire leur preuve et de se positionner dans le marché (qui est hyper compétitif) à travers (et après) la procédure de sélection.

CHAPITRE 1. CONTEXTE ET OBJECTIFS DU PROJET

TABLE 1.1 – Comparaison entre différentes plateformes

	Upwork	Toptal	Freeelancer	Freelance	iLancer
Date de fondation	2015	2010	2009	2017	2020
Basée à					
Nombre de freelances	5MM	10M	50MM	N/A	N/A
Frais	-20%, -10% et -5%	Ça dépend de plusieurs facteurs	-5 \$ ou -10% (le plus haut) pour tous les projets à prix fixe	Aucun	N/A
Procédure de sélection	Oui	Oui	Non	Non	Non
Messagerie	Oui	Oui	Oui	Non	Oui
Notifications	Oui	Oui	Oui	Oui	Oui
Gestion du profil	Oui	Oui	Oui	Oui	Oui
Système de feedback	Oui	Oui	Oui	Non	Oui
Packs payants	Oui	Non	Oui	Non	Non
Paiement global	Oui	Oui	Oui	Non	Non

CHAPITRE 1. CONTEXTE ET OBJECTIFS DU PROJET

Un des problèmes récurrents pour la plupart des plateformes internationales est le taux de commission important. Ça représente un obstacle pour les professionnelles qui veulent se convertir vers le travail indépendant.

Les plateformes tunisiennes offrent les fonctionnalités nécessaires pour un site web de recherche d'emploi ou un marché des petites tâches afin qu'il soit utile et pratique. Cela ne nous empêche pas de dégager certains défauts :

- Freelance Tunisie ne prend pas en charge la communication entre les parties intervenantes et les prestataires envoient leurs demandes à travers un courriel vers l'entreprise/le client concerné,
- iLancer n'a pas mis la charte de ses frais de service en public,
- iLancer.tn et Freelance Tunisie n'imposent aucun test/procédure de sélection à l'inscription des nouveaux freelances/demandeurs d'emploi chez eux. Cela peut entraîner une qualité modeste des travailleurs, et par la suite, une perte de chemin vers le bon talent.

1.3.3 Solution proposée

Face aux défaillances des solutions déjà existantes l'échelle nationale et en apprenant des plateformes internationales, nous proposons de concevoir et réaliser une nouvelle plateforme de freelance sous le nom de “lanci.tn” avec le logo illustré à la figure 1.7 qui va être la première en sa manière de fonctionnement en Tunisie. Ainsi, notre projet doit être conforme à l'objectif de développement d'un site web convivial et ergonomique de travail indépendant qui offre les fonctionnalités suivantes :

- La consultation des pages publiques (“La page d'accueil”, “Comment ça marche ?”, “liste des freelances”, etc.) pour s'informer mieux à propos du concept général de la plateforme,
- La consultation des annonces des missions publiées par les différents clients ainsi que leurs profils pour les freelances et la possibilité d'ajouter les publications qui l'intéressent aux favoris,
- La recherche des annonces selon différents critères,
- La consultation des profils publics des freelances afin de trouver le bon talent pour accomplir la mission demandée,
- L'attribution de la réalisation de mission au freelance qui son offre a été accepté,

CHAPITRE 1. CONTEXTE ET OBJECTIFS DU PROJET

- La gestion de profils, des informations personnelles et des paramètres du compte des utilisateurs,
- L'échange des messages entre un donneur d'ordre et un travailleur indépendant,
- L'envoi des notifications aux utilisateurs à propos des nouvelles publications, de ses publications ou de les offres postées sur ses annonces.



FIGURE 1.7 – Logo de notre plateforme lanci.tn

1.3.4 Objectifs du projet

En se basant à la maturité que les grandes plateformes et marchés de services possèdent, on a fixé certains objectifs à atteindre pour notre application :

- Permettre aux prestataires de créer leurs profils avec toutes les informations nécessaires afin de construire une identité digitale sur la plateforme et attirer les clients potentiels,
- Permettre aux clients inscrits de poster leurs missions et afficher les données relatives à chacune d'eux et plus précisément les offres des prestataires et les prix proposés,
- Permettre aux prestataires de postuler sur ces annonces des missions et de mettre une offre avec le prix qui convient,
- Permettre aux clients d'accepter l'offre qu'il voit convenable et assez conforme aux spécifications de mission et à son budget avec le profil de freelance recherché,
- Permettre aux admins de gérer les comptes, les profils et aussi les annonces,
- Permettre à tout utilisateur de la plateforme d'effectuer une recherche sur la plateforme afin d'explorer les annonces des missions postées.

1.4 Méthodologie de travail

Dans cette section, on va citer quelques méthodologies de travail et de réalisation des projets informatiques classiques ainsi que modernes. On finira cette section par justifier le choix de la méthodologie qu'on va adopter tout au long de notre projet.

1.4.1 Méthode classique (Modèle en cascade)

On appelle modèle en cascade (waterfall model) un modèle de gestion séquentiel permettant de représenter les développements à travers des phases successives [15]. On doit le développement de ce modèle classique à l'informaticien Winston Walker Royce. En pratique, plusieurs versions du modèle en cascade sont utilisées. Les modèles les plus courants divisent les processus de développement en cinq phases :

- Analyse : planification, analyse et spécification des besoins,
- Conception : conception et spécification du système,
- Implémentation : programmation et tests des modules,
- Test : intégration du système, tests du système et de l'intégration,
- Exploitation : livraison, maintenance, amélioration.

La figure 1.8 illustre pourquoi le modèle de gestion linéaire est qualifié de « modèle en cascade ».

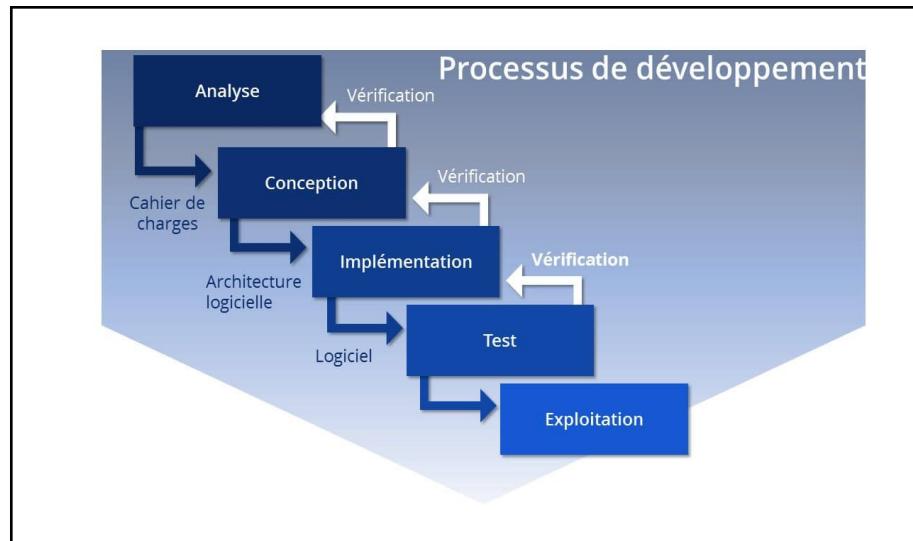


FIGURE 1.8 – Illustration de modèle de développement en cascade

1.4.2 Agile Scrum

Contrairement aux méthodologies classiques, on trouve les méthodes de développement itératives modernes. Parmi ces méthodes, nous pouvons distinguer les méthodes Agile [11] largement utilisées de nos jours à travers le monde.

Agile Scrum [16] est certainement la méthode de gestion de projet Agile la plus connue et la plus simple à comprendre. Souple et légère, elle peut être appliquée à des nombreuses situations.

Le principe de la méthode Agile Scrum est de concentrer l'équipe de développement sur un ensemble de fonctionnalités à réaliser d'une façon itérative dans des itérations d'une durée de deux à quatre semaines, appelées des Sprints. Chaque Sprint doit aboutir à la livraison d'un produit partiel. La figure 1.9 montre le cycle de vie de la méthodologie Agile Scrum.

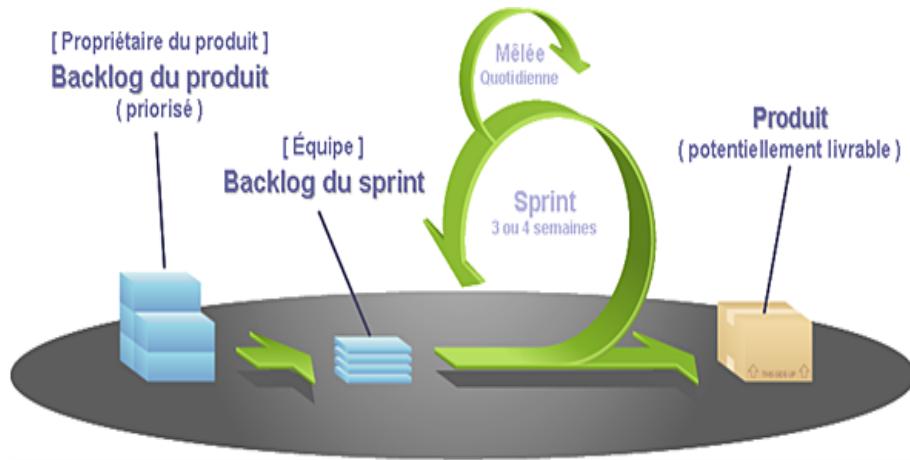


FIGURE 1.9 – Cycle de vie de la méthodologie Agile Scrum

1.4.3 Comparaison des méthodologies et choix

Afin de choisir entre une méthodologie de développement ou une autre, il faut mettre en évidence la nature du projet à élaborer, sa taille, ses exigences et ses perspectives. Pour les projets de petite durée et dont le domaine est maîtrisé, par exemple, le modèle en cascade s'avère largement suffisant. Lorsqu'il s'agit d'un projet où les données ne sont pas réunies dès le départ, où les besoins sont incomplets, il faut s'orienter vers une méthode itérative ou orientée prototypes.

Une méthode Agile est menée dans un esprit collaboratif et s'adapte aux approches incrémentales. Elle engendre des produits de haute qualité tout en tenant compte de l'évolution des besoins du client. Une méthode Agile assure une meilleure communication avec le client et une meilleure visibilité du produit livrable. Elle permet aussi de gérer la qualité en continu et de détecter des problèmes le plus tôt au fur et à mesure, permettant ainsi d'entreprendre des actions correctrices sans trop de pénalités dans les coûts et les délais.

La nature de notre projet doit être évolutive. Et dont tous les besoins n'ont pas encore été totalement identifiés, nous étions orientés vers une méthode de type Agile et plus particulièrement Scrum.

1.5 Conclusion

Dans ce chapitre, nous avons commencé par une présentation du contexte général. Puis, on a montré notre étude de l'existant avec une comparaison entre différentes plateformes et leurs fonctionnalités. Par la suite, on a présenté certaines méthodes de travail existantes et nous avons pu choisir celle que nous avons adopté. Le chapitre suivant sera consacré à l'analyse et la spécification des besoins de notre application.

Chapitre 2

Analyse et spécifications des besoins

2.1 Introduction

Après avoir présenté le contexte général, on va consacrer ce chapitre pour analyser et expliquer les exigences de l'application. Dans un premier temps, on va présenter les acteurs qui vont interagir avec le système. Ensuite, on continue vers l'étude des besoins fonctionnels et non fonctionnels. Puis, on va créer les diagrammes nécessaires en fonction de ces exigences. Enfin, on va présenter le Backlog produit et la chronologie des sprints.

2.2 L'analyse des besoins

Dans cette partie, nous déterminerons les acteurs et leurs rôles. Plus tard, nous expliquerons les différentes exigences fonctionnelles et non fonctionnelles que notre application cherche à satisfaire.

2.2.1 Identification des acteurs

On débute l'étude d'interaction par dégager les acteurs qui vont s'interagir avec le système :

- Acteur 1 : **un visiteur** qui peut consulter la page d'accueil, contacter l'administration du site et créer un compte en tant qu'**un client ou un freelance**,
- Acteur 2 : **un utilisateur** qui a les caractéristiques mutuelles de client et de freelance (*Veuillez noter que cet acteur est utilisé juste pour le regroupement des actions et d'interactions, il ne sera pas donc élaboré en code*),
- Acteur 3 : **un client** qui possède un compte qui lui permet de bénéficier des fonctionnalités conformes à son type de compte,

- Acteur 4 : **un freelance** qui possède un compte qui lui permet de bénéficier des fonctionnalités conformes à son type de compte,
- Acteur 5 : **un administrateur** qui gère le contenu (les annonces des missions) et les utilisateurs de la plateforme.

2.2.2 Les besoins fonctionnels

Généralement formulés sous forme d'exigences fonctionnelles, les besoins fonctionnels sont l'expression de ce que le service délivré par le projet devrait être ou faire.

L'application doit permettre le visiteur de :

- **Consulter contenu** : le visiteur peut consulter le contenu public de la plateforme comme la page d'accueil et la page "Comment ça marche?",
- **Contacter l'administrateur** : la mise en contact est établi avec un formulaire du contact qui envoie le message vers un e-mail à l'équipe administrative (à l'adresse admin@lanci.tn),
- **Créer un compte** : s'inscrire à la plateforme en tant qu'un nouvel utilisateur (client ou freelance).

L'application doit permettre l'utilisateur (client ou freelance) de :

- **Se connecter** : se connecter à la plateforme avec son nom d'utilisateur (pseudo) et son mot de passe,
- **Consulter les annonces des missions** : consulter les annonces publiées par les différents clients et effectuer une recherche selon ses titres,
- **Gérer son compte** : gérer ses paramètres du compte (modifier son email et/ou son mot de passe),
- **Gérer son profil** : afin de bien gérer son profil, l'utilisateur peut :
 - Consulter son profil,
 - Modifier sa photo de profil,
 - Modifier sa biographie,
 - Modifier son titre,
 - Gérer (ajouter et supprimer) les langues qu'il parle,

CHAPITRE 2. ANALYSE ET SPÉCIFICATIONS DES BESOINS

- Modifier son nom et/ou son prénom,
 - Modifier sa ville,
 - Modifier sa date de naissance,
 - Modifier son téléphone.
- **Consulter ses messages :** consulter les conversations qu'il a avec la possibilité de répondre avec un message. Il sera aussi possible de supprimer une conversation,
 - **Consulter ses notifications :** consulter les notifications qu'il avait reçues,
 - **Se déconnecter :** se déconnecter de la plateforme.

L'application doit permettre le client de :

- **Gérer ses annonces des missions :** Afin de bien manager ses annonces des missions, le client pourra :
 - Consulter ses annonces,
 - Ajouter une annonce,
 - Modifier une annonce,
 - Marquer une annonce comme terminée : après l'accomplissement d'une mission avec succès par le prestataire, le client doit marquer l'annonce comme terminée afin de procéder aux étapes suivantes,
 - Supprimer une annonce.
- **Consulter les offres :** un client doit être capable de lister les propositions postées sur chacune de ses missions et il peut réagir par l'acceptation ou par le rejet,
- **Consulter les freelances :** une page va être consacrée à l'affichage des freelances inscrits à la plateforme. Elle va aider le client de trouver le bon talent et il pourra contacter le prestataire qui lui intéresse directement par un message,
- **Alimenter son solde :** alimenter le compte à travers les méthodes disponibles.

CHAPITRE 2. ANALYSE ET SPÉCIFICATIONS DES BESOINS

L’application doit permettre le freelance de :

- **Gérer son profil** : afin de mieux gérer son profil, le freelance peut :
 - Gérer ses formations :
 - * Ajouter une formation,
 - * Modifier une formation,
 - * Supprimer une formation.
 - Gérer ses expériences professionnelles :
 - * Ajouter une expérience professionnelle,
 - * Modifier une expérience professionnelle,
 - * Supprimer une expérience professionnelle.
 - Gérer ses liens vers les réseaux sociaux et les plateformes professionnelles : Facebook, Instagram, Twitter, LinkedIn, Medium, Dribbble, Behance, GitHub, GitLab, lien vers son site web personnel ou son portfolio,
 - Gérer (ajouter et supprimer) ses compétences.
- **Proposer une offre sur une annonce de mission** : Postuler sur la mission qu’il trouve la plus conforme à ses compétences et son taux horaire,
- **Consulter les annonces des missions** : Consulter les annonces publiées par les clients. Il peut aussi consulter l’annonce qui l’intéresse séparément et l’ajoute à ses favoris s’il veut,
- **Consulter ses favoris** : Lister les annonces qu’il a mis aux favoris,
- **Gérer ses offres** :
 - Consulter les offres postées sur les différentes annonces,
 - Supprimer une offre.
- **Retirer son argent** : Utiliser les méthodes de paiements de la plateforme afin de recevoir son argent bien mérité.

L’application doit permettre l’administrateur de :

- **Gérer son compte d’administrateur** : Consulter et modifier ses informations de connexion (son mot de passe),
- **Gérer les utilisateurs** : Effectuer les actions classiques de gestion des collections au système :

- Consulter la liste des utilisateurs (que ce soit clients ou freelances),
 - Modifier les informations d'un utilisateur,
 - Supprimer un utilisateur.
- **Gérer les annonces des missions :** D'une manière pareille au cas d'utilisation précédente il peut :
 - Consulter les annonces,
 - Modifier une annonce,
 - Supprimer une annonce.

2.2.3 Les besoins non fonctionnels

Il s'agit des conditions nécessaires permettant d'assurer le bon fonctionnement et la continuité du système et d'améliorer la qualité des services délivrés vers l'utilisateur. Notre système doit répondre aux besoins suivants :

- **L'extensibilité :** L'extensibilité ou la scalabilité désigne la capacité d'un produit à s'adapter à un changement d'ordre de grandeur de la demande (montée en charge), en particulier sa capacité à maintenir ses fonctionnalités et ses performances en cas de forte demande [8]. Dans le cadre de notre travail, l'application devra être extensible et évolutive. On pourra avoir la possibilité d'ajouter ou de modifier des fonctionnalités afin de répondre à la grande demande si ça sera le cas.
- **La sécurité :** L'application devra être sécurisée. D'autres mots, le contenu ne devra être accessible que par un identifiant et un mot de passe. Cela garantira la protection des ressources et ensuite l'attribution d'accès aux informations en fonction du rôle choisi ou attribué à chaque type d'utilisateur.
- **La responsivité :** on a essayé d'appliquer le principe de RWD (Responsive Web Design) et c'est une approche de conception web qui vise à élaborer le site en offrant une expérience de lecture et de navigation optimales pour l'utilisateur quelle que soit sa gamme d'appareil.

2.3 Spécification des besoins fonctionnels

Dans cette partie nous allons préciser les différentes fonctionnalités présentes dans notre application. C'est pour cette raison qu'on va utiliser les cas d'utilisation et les user stories d'une façon complémentaire.

Un cas d'utilisation définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles.

2.3.1 Diagramme de cas d'utilisation général

La figure 2.1 illustre le diagramme de cas d'utilisation général.

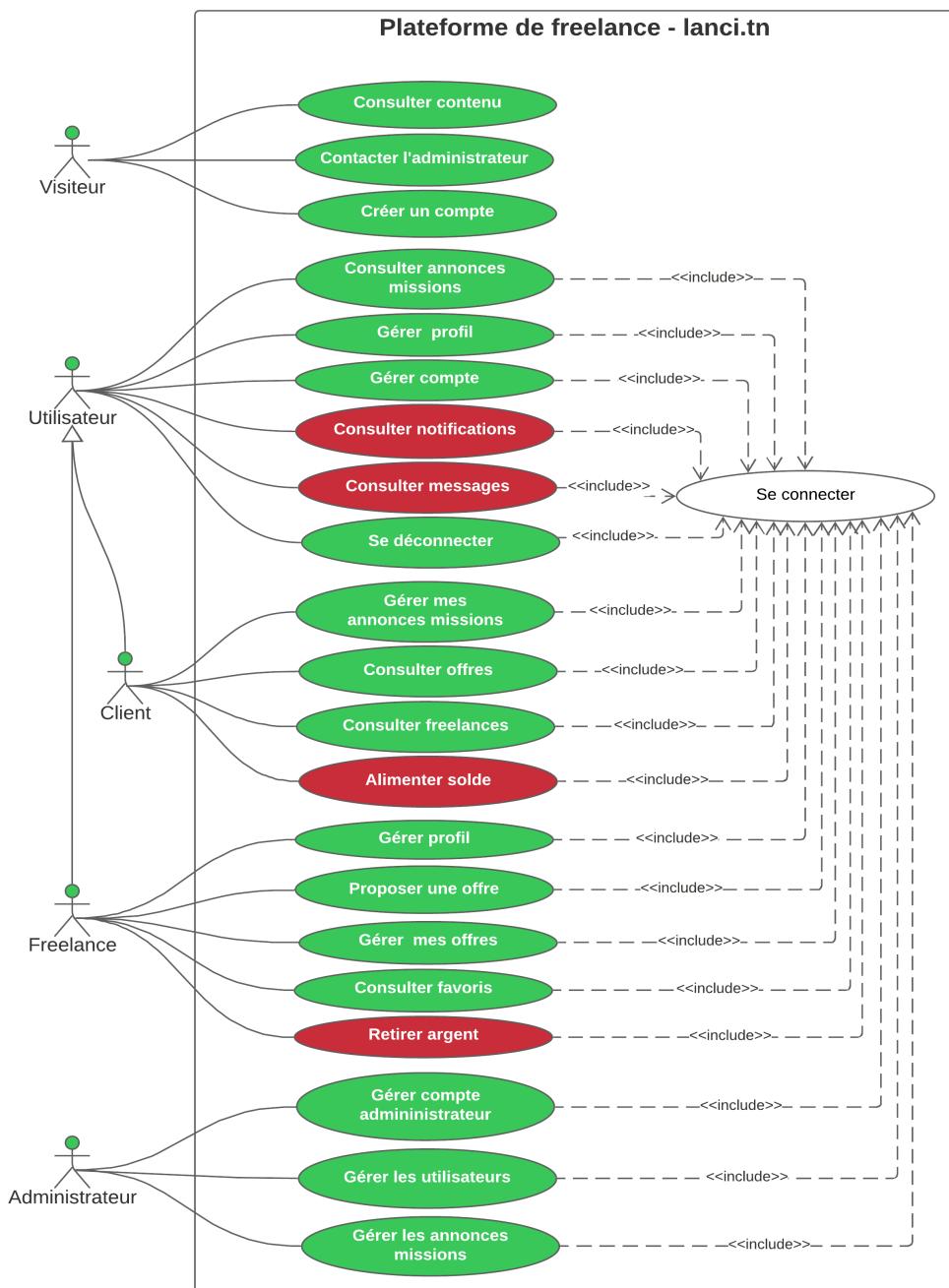


FIGURE 2.1 – Diagramme de cas d'utilisation général

Veuillez noter que les actions colorées en rouge sont les parties qu'on a pas réussie à achever.

2.3.2 Diagrammes de cas d'utilisation détaillés

Diagramme de cas d'utilisation détaillé sur l'acteur "Visiteur"

Le visiteur, dont la figure 2.2 représente son cas d'utilisation détaillé avec le système, peut consulter la page d'accueil de la plateforme pour mieux connaître son contenu, contacter l'administration du site à l'aide d'un formulaire prédéfini. Il pourra créer un compte pour s'inscrire à la plateforme. Il existe deux types de comptes : le compte pour un client et le compte pour un freelance.

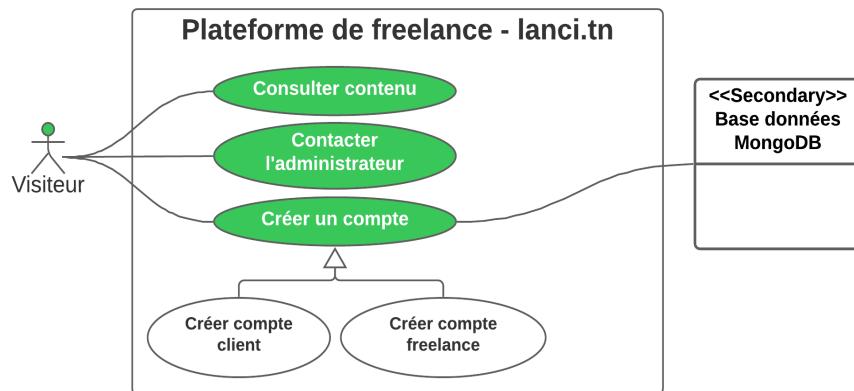


FIGURE 2.2 – Diagramme de cas d'utilisation de visiteur

Diagramme de cas d'utilisation détaillé sur l'acteur "Utilisateur"

La figure 2.3 représente le diagramme de cas d'utilisation détaillé de l'utilisateur (*Veuillez noter que toutes les actions de ce diagramme manipules d'une façon ou une autre la base de données MongoDB. Mais, Afin de garantir la bonne clarté du diagramme, on a éliminé cet acteur secondaire*).

CHAPITRE 2. ANALYSE ET SPÉCIFICATIONS DES BESOINS

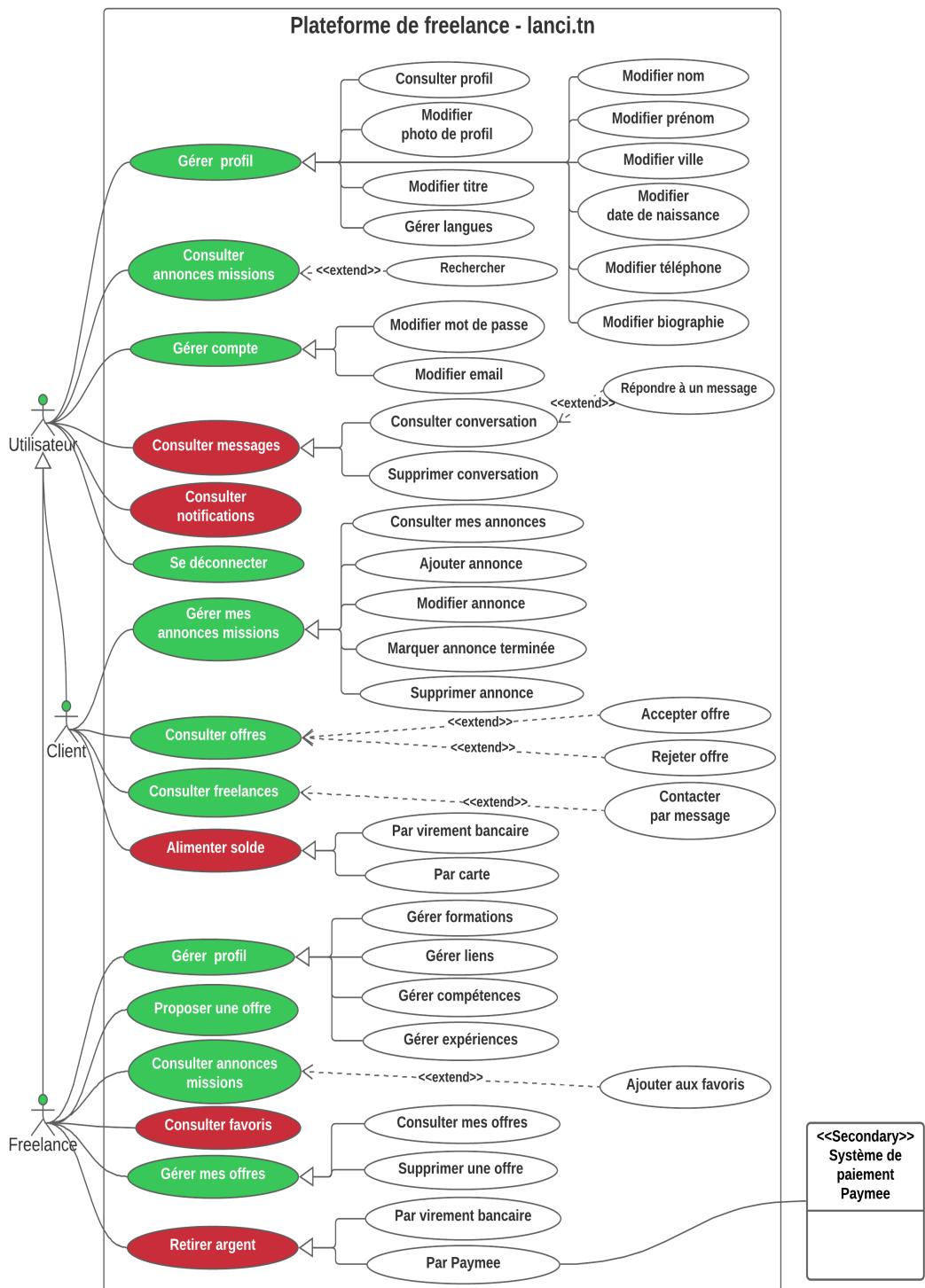


FIGURE 2.3 – Diagramme de cas d'utilisation de l'utilisateur

Diagramme de cas d'utilisation détaillé sur l'acteur "Administrateur"

la figure 2.4 représente le diagramme de cas d'utilisation détaillé de l'administrateur de la plateforme

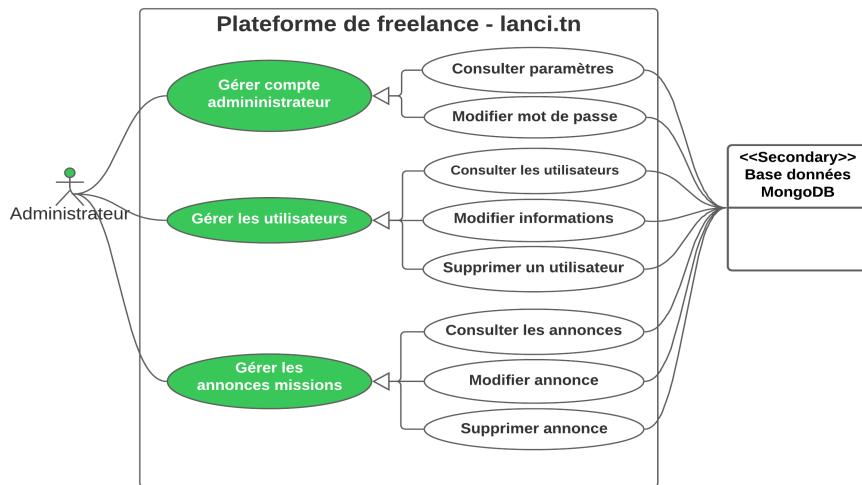


FIGURE 2.4 – Diagramme de cas d'utilisation de l'administrateur

2.3.3 Diagrammes de séquence système

Le comportement interactif entre les acteurs et le système peut être décrit par un diagramme d'interaction (par ex. diagramme de séquences) système. Dans cette partie, vous trouverez les diagrammes de séquences système des principaux cas d'utilisation et une description pour chacune.

Diagramme de séquence “Créer un compte” déclenché par un visiteur

Le tableau 2.1 et La figure 2.5 décrivent le cas d'utilisation “Créer un compte”

Diagramme de séquence “Se connecter” déclenché par un visiteur

Le tableau 2.2 décrit le diagramme de séquence “Se connecter” illustré par la figure 2.6.

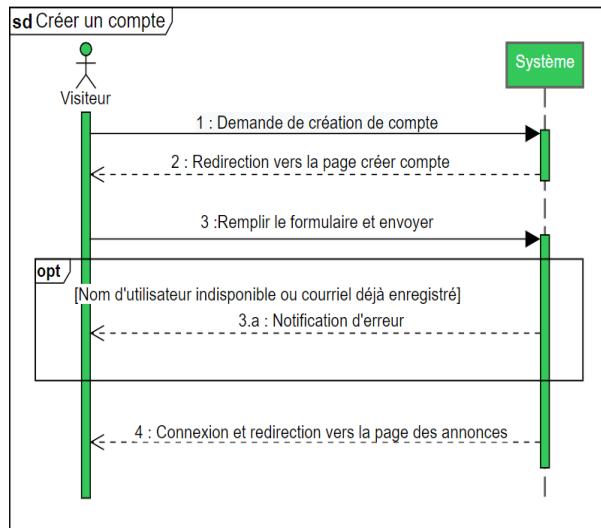


FIGURE 2.5 – Diagramme de séquence "Créer un compte"

TABLE 2.1 – Description du cas d'utilisation "Créer un compte"

Cas d'utilisation	Créer un compte
Intérêt	La création d'un compte sur la plateforme pour devenir un utilisateur (freelance ou client).
Acteur	Visiteur
Précondition	Le visiteur doit être sur une des pages publiques du site
Scénario nominal	<ol style="list-style-type: none"> 1. Le visiteur demande la création du compte. 2. Le système fait la redirection vers la page créer compte et lui affiche un formulaire d'inscription 3. Le Visiteur remplit le formulaire 4. Le système autorise la connexion et la redirection se fait vers la page mes annonces
Scénario alternatif	3.a. Si le nom d'utilisateur est indisponible (existe déjà au système), le système répond par une notification d'erreur
Postcondition	Un nouvel utilisateur est ajouté
Exception	Les erreurs sont générées par le système et un message est affiché pour toute entrée invalide.

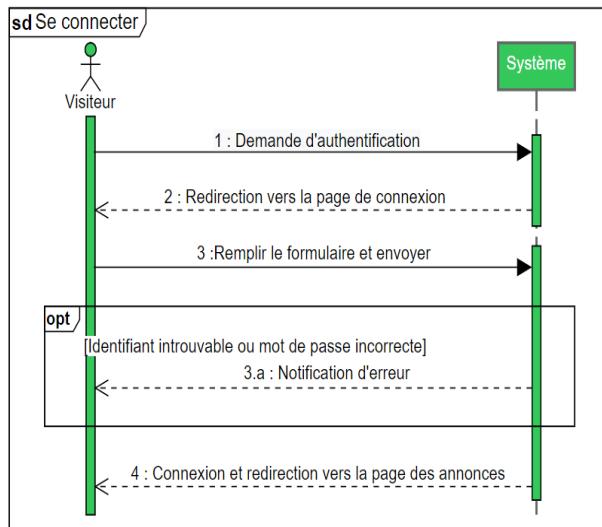


FIGURE 2.6 – Diagramme de séquence "Se connecter"

TABLE 2.2 – Description du cas d'utilisation "Se connecter"

Cas d'utilisation	Se connecter
Intérêt	L'authentification
Acteur	Visiteur
Précondition	Le visiteur est sur le site
Scénario nominal	1. Le visiteur demande de se connecter 2. Le système fait la redirection vers la page de connexion et lui affiche un formulaire d'authentification. 3. Le visiteur remplit le formulaire 4. Le système autorise la connexion et la redirection se fait vers la page des annonces.
Scénario alternatif	3.a. Si l'identifiant est introuvable ou le mot de passe est incorrecte, le système répond par une notification d'erreur
Postcondition	L'utilisateur se connecte à son compte
Exception	Les erreurs sont gérées par le système et une notification est affiché pour toute entrée invalide

Diagramme de séquence “Gérer mes annonces missions” déclenché par un client

Le tableau 2.3 et la figure 2.7 décrivent le cas d'utilisation “Gérer mes annonces” déclenché par un client.

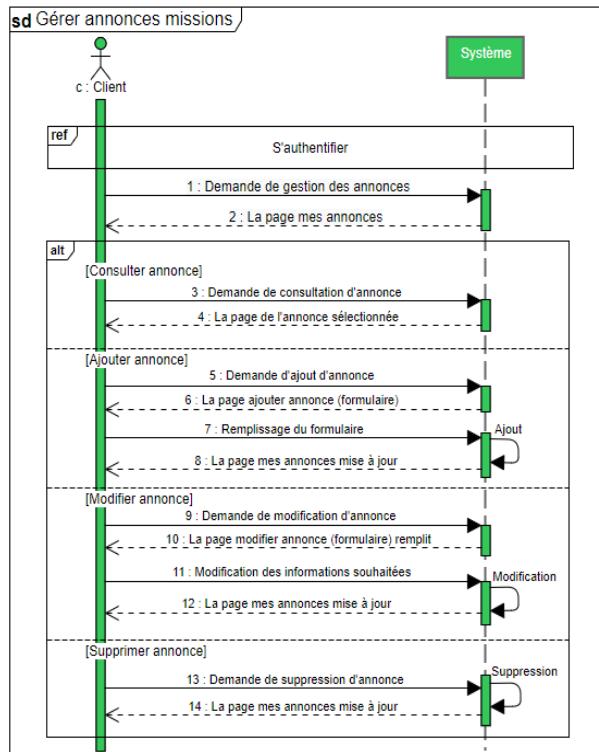


FIGURE 2.7 – Diagramme de séquence "Gérer mes annonces"

2.4 Backlog du produit et pilotage du projet avec Agile Scrum

2.4.1 Backlog du produit

Le Backlog du produit est une liste ordonnée par priorité des besoins fonctionnels du client qui sont appelés des histoires utilisateurs (User stories), la notion de backlog produit est notamment liée à l’application de la méthode Scrum. Ils représentent une description brève de l’exigence fonctionnelle formulée en une phrase sous la forme suivante : "En tant qu'un <acteur>, je veux <un objectif> pour que <une raison>".

CHAPITRE 2. ANALYSE ET SPÉCIFICATIONS DES BESOINS

TABLE 2.3 – Description du cas d'utilisation "Gérer mes annonces"

Cas d'utilisation	Gérer annonces missions
Intérêt	Gérer annonces missions
Acteur	Client
Précondition	Un client est connecté et possède au moins une annonce
Scénario nominal	<p>1. Le client demande la page de ses annonces 2. Le système fait la redirection vers la page mes annonces (annonces du client)</p> <p>-Consulter annonce : 3. Le client demande la consultation d'une annonce 4. Le système fait la redirection vers la page de l'annonce demandée</p> <p>-Ajouter annonce : 5. Le client demande d'ajouter une annonce 6. Le système autorise la redirection vers la page ajout annonce 7. Le client remplit le formulaire et valide 8. Le système ajoute l'annonce et renvoie vers la page mes annonces</p> <p>-Modifier annonce : 9. Le client demande la modification d'une annonce 10. Le système renvoie le formulaire rempli avec les anciennes informations 11. Le client modifie les informations souhaitées 12. Le système applique la modification et fait la redirection vers la page mes annonces</p> <p>-Supprimer annonce : 13. Le client demande la suppression d'une annonce. 14. Le système supprime l'annonce actualise les données de la page mes annonces</p>
Postcondition	La liste des annonces est mise à jour
Exception	Les erreurs sont gérées par le système et un message est affiché pour les entrées invalides

Le Backlog du produit présenté dans le tableau 2.4 comprend les champs suivants :

Thème : Représente le thème du user story.

User story : C'est l'histoire utilisateur décrivant un besoin fonctionnel désiré par le client.

Priorité : Représenté par la méthode de priorisation de Jira dans laquelle chaque user story est priorisée comme l'un des suivants :

- **Highest** : La fonctionnalité doit être faite.
- **High** : La fonctionnalité devrait être faite dans la mesure du possible.
- **Medium** : La fonctionnalité a un degré de priorité moyenne.
- **Low** : La fonctionnalité pourrait être faite si elle n'a pas d'impact sur les autres tâches.
- **Lowest** : La fonctionnalité ne sera pas faite cette fois, mais sera faite plus tard.

2.4.2 Pilotage du projet avec Agile Scrum

Release : La release correspond à la livraison d'une version de l'application. Il s'agit de la période allant du début des travaux de cette version à la livraison, elle a traversé une série de sprints continus.

Sprint : C'est le terme utilisé pour l'itération dans Scrum. C'est un bloc de temps dont le but est de créer un incrément potentiel de produit livrable. Son objectif doit être défini en termes commerciaux plutôt qu'en termes techniques afin que les membres extérieurs à l'équipe puissent le comprendre. Pour notre projet, nous avons choisi de réaliser deux releases décrites par le tableau 2.5. La première release est nommée « Version 1.0 », la deuxième est intitulée « Version 2.0 ».

2.5 Conclusion

Une description détaillée des besoins facilite énormément la gestion du projet ainsi que sa réalisation et contribue efficacement à sa réussite. Au cours de ce chapitre, on a identifié d'abord les principaux acteurs et les besoins fonctionnels et non fonctionnels de notre application. Après, on a présenté les diagrammes de cas d'utilisation et les diagrammes de séquence système. Enfin, on a montré notre Backlog produit et la planification des sprints et des releases. Le chapitre suivant sera consacré au sprint 0 qui est accordé à l'initialisation de projet.

CHAPITRE 2. ANALYSE ET SPÉCIFICATIONS DES BESOINS

TABLE 2.4 – Backlog du produit

Thème	User story	Priorité
Création compte	En tant que visiteur, je veux créer un compte pour bénéficier du contenu de la plateforme.	Highest
Connexion	En tant qu'utilisateur ou admin, je veux m'authentifier pour accéder à mon espace personnel.	Highest
Gestion compte	En tant qu'utilisateur, je veux modifier mes paramètres afin d'avoir un contrôle sur mon compte.	Highest
Gestion Profil	En tant qu'utilisateur, je veux gérer mon profil afin d'avoir une meilleure apparence.	Highest
Se déconnecter	En tant qu'utilisateur, je veux terminer ma session intentionnellement.	Highest
Consultation annonce service	En tant qu'utilisateur, je veux parcourir les annonces publiées afin d'avoir une idée sur le marché.	High
Gestion des annonces services	En tant qu'un client, je veux gérer les annonces que je publie à la plateforme (créer, modifier, supprimer).	High
Consultation des offres	En tant qu'un client, je veux consulter les offres proposées sur l'une de mes annonces.	High
Consultation des profils des freelances	En tant qu'un client, je veux parcourir les freelances et consulter le profil qui m'intéresse.	High
Proposer une offre sur une annonce service	En tant qu'un freelance, je veux poser ma candidature sur l'annonce d'une mission.	High
Gestion des comptes	En tant qu'un admin, je veux gérer les comptes des utilisateurs (modifier et supprimer).	High
Gestion compte administrateur	En tant qu'un admin, je veux gérer mes paramètres de connexion.	High
Gestion des annonces services	En tant qu'un admin, je veux gérer les annonces publiées.	High
Gestion des profils utilisateurs	En tant qu'un admin, je veux modifier les profils des utilisateurs.	High
Messagerie	En tant qu'utilisateur, je veux envoyer des messages et parcourir mes discussions afin d'avoir une bonne compréhension de mission.	Low
Consultation des notifications	En tant qu'utilisateur, je veux être notifié pour rester au courant des nouveautés.	Low
Gestion paiement	En tant qu'utilisateur, je veux choisir ma méthode de paiement préféré (payer ou être payé)	Lowest

CHAPITRE 2. ANALYSE ET SPÉCIFICATIONS DES BESOINS

TABLE 2.5 – Planification des releases et des sprints

Release	Sprint	Tâches à faire
Release 1 : Version 1.0	Sprint 1	<ul style="list-style-type: none">- Consulter contenu public- Créer un compte- Se connecter- Se déconnecter
	Sprint 2	<ul style="list-style-type: none">- Gérer profil par le client- Gérer profil par le freelance- Gérer compte par l'utilisateur- Gérer compte administratif par l'administrateur- Gérer les utilisateurs par l'administrateur
Release 2 : Version 2.0	Sprint 3	<ul style="list-style-type: none">Consulter les annonces par tout le mondePoster une annonce mission par le clientGérer annonces par le clientProposer une offre sur une annonce par le freelanceGérer les offres par le freelanceGérer les annonces par l'administrateur

Chapitre 3

Initialisation du projet

3.1 Introduction

Dans ce chapitre, nous nous intéressons à l'initialisation du projet. Nous commençons par la présentation de l'architecture générale de notre application web. Par la suite, on va citer les frameworks de développement et la manière de garantir la persistance des données de plateforme. Enfin, on va nommer les outils de conception, de développement et de travail collaboratif qu'on a utilisé au cours de la réalisation.

3.2 Architecture générale de la plateforme

3.2.1 Architecture logicielle

Les applications web modernes sont de plus en plus grandes et complexes. Afin de réduire la complexité, nous utilisons le principe de séparation des responsabilités (SRP : Separation of Responsibility Principle). L'utilisation de SRP garantit un couplage lâche, une maintenabilité et une testabilité. Le modèle MVC(Model View Controller) qu'on va utiliser pour développer notre application incarne cette philosophie de séparation des responsabilités. Les différentes parties de l'architecture MVC sont illustrées par la figure 3.1.

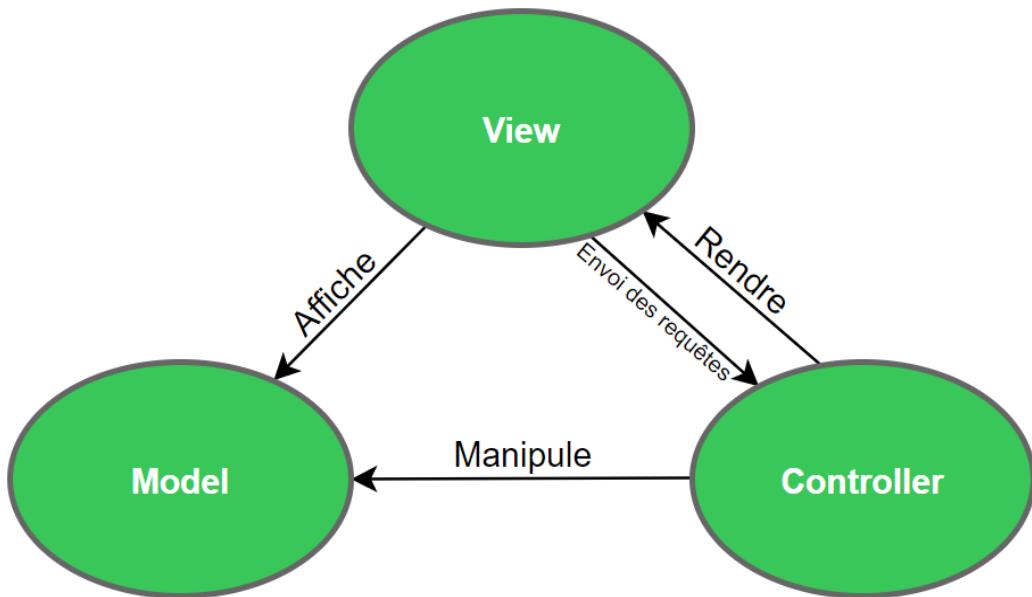


FIGURE 3.1 – L’architecture MVC

Les couches d’une application à une architecture MVC sont :

- **Le modèle (Model)** : les composants du modèle sont responsables du domaine de données de l’application. Les objets du modèle sont responsables du stockage, de la récupération et de la mise à jour des données de la base de données. Dans notre cas, c’est le rôle de MongoDB à travers l’ODM Mongoose,
- **La vue (View)** : c’est l’interface utilisateur de notre application. Dans la plupart des applications Web modernes, la couche de vue est généralement remplacée par une autre application à page unique (Single page application). Dans notre cas, on a utilisé React.js,
- **Les contrôleurs (Controllers)** : ils sont responsables de la gestion des interactions avec les utilisateurs. Ils interagissent avec les modèles pour récupérer des informations et finalement répondre aux requêtes des utilisateurs. On a utilisé Expressjs pour créer, maintenir et acheminer vers nos contrôleurs.

Une représentation de haut niveau de l’architecture logicielle en appliquant MVC à notre système est illustrée par la figure 3.2.

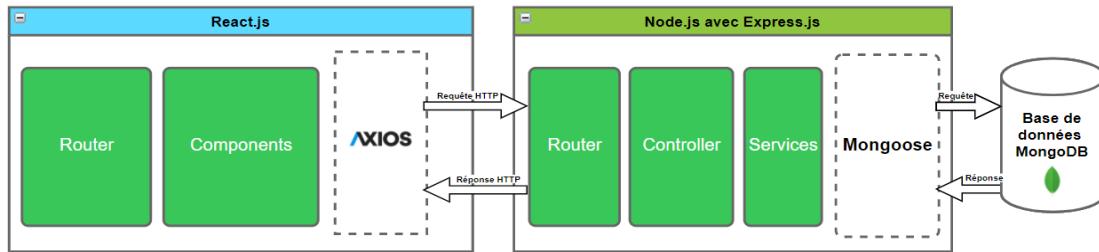


FIGURE 3.2 – L'architecture MVC de l'application

3.2.2 Architecture physique

La figure 3.3 illustre l'architecture physique de l'application

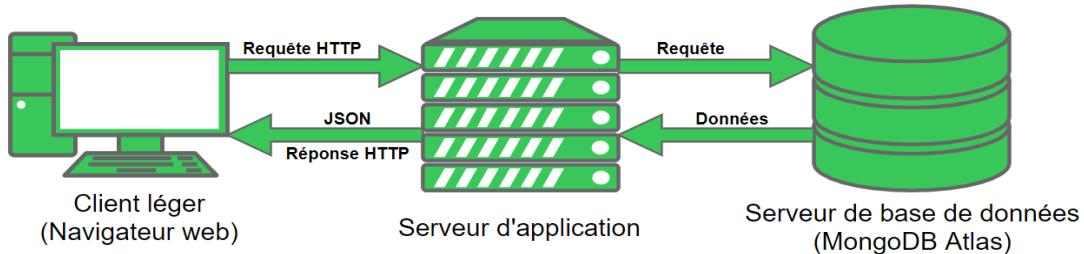


FIGURE 3.3 – L'architecture physique de la plateforme

- **Client léger** : Le navigateur web installé sur le poste client qui est responsable de l'affichage et qui assure la communication avec l'utilisateur et le serveur d'application.
- **Serveur de l'application** : Le serveur applicatif qui est chargé de la partie fonctionnelle et de la logique de l'application.
- **Serveur base de données** : Ce serveur va permettre de stocker et de restituer les données. On a utilisé MongoDB Atlas dans notre projet en phase de développement.

3.3 Les frameworks de développement

Les frameworks sont des ensembles de composants permettant de mettre en place rapidement les fondations d'une application tout en fixant des standards aux développeurs travaillant sur le projet, ce qui facilite le travail collaboratif.

3.3.1 Framework front-end

Les frameworks et les bibliothèques Javascript sont nombreux, et plusieurs d'entre eux sont destinés à la création des interfaces pour le Web. On peut citer par exemple : jQuery, Angular (de la part de Google), React (Facebook), Vue, Ember, etc.

Après avoir fait une analyse des spécifications de notre projet et également la spécificité de chacune des frameworks populaires, on a choisi React.js.

Présentation et spécificité de la bibliothèque React.js

React.js est une bibliothèque JavaScript open source développée en 2013 par Facebook. Le principal avantage de cette bibliothèque est de faciliter la création du contenu web mono-page.

Le principe est de créer des composants réutilisables, les ensembles des composants créés constituent l'interface de l'application. Avec React, on peut créer des vues dynamiques, produire des interfaces utilisateurs simples pour chaque état de votre application, et lorsque vos données changeront, React mettra à jour, de façon automatique, les composants qui auront besoin d'être rafraîchi.

La principale caractéristique de React.js est l'assemblage de composants. D'autres mots, des composants écrits par des personnes différentes devraient fonctionner correctement ensemble. Nous tenons à ce que vous puissiez ajouter des fonctionnalités à un composant sans que ça impacte le reste de la base de code. Par exemple, il devrait être possible d'introduire un état local dans un composant sans changer les composants qui l'utilisent. Dans le même esprit, il devrait être possible d'ajouter du code d'initialisation et de nettoyage à n'importe quel composant lorsque c'est nécessaire.

Diagramme de fonctionnement d'une application React.js

La figure 3.4 illustre le modèle que chaque application construite avec React.js adopte.

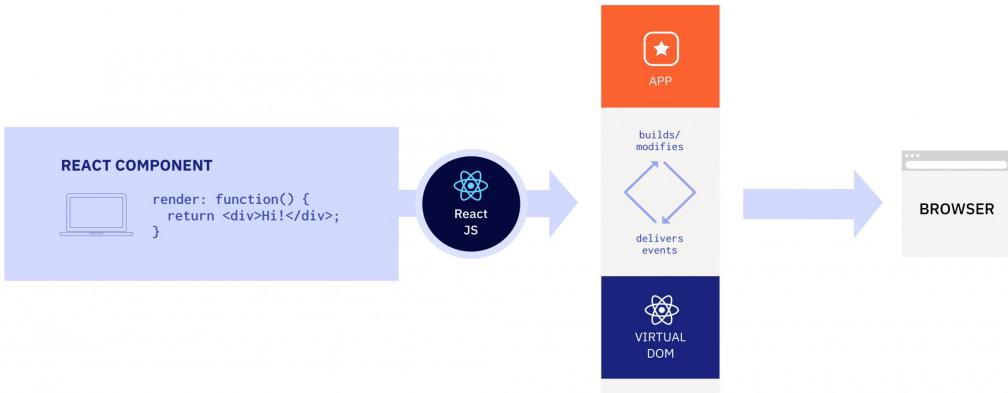


FIGURE 3.4 – Diagramme de fonctionnement d'une application React.js

3.3.2 Framework back-end

Présentation et spécificité d'Express.js

Express est une infrastructure d'application (framework), écrite en JavaScript et hébergée dans l'environnement d'exécution Node.js [5].

Minimaliste et très léger, il apporte peu de surcouches pour garder des performances optimales et une exécution rapide. Express ne fournit que des fonctionnalités d'application web (et mobile) fondamentales, mais celles-ci sont extrêmement robustes et ne prennent pas le dessus sur les fonctionnalités natives de Node.js.

3.4 REST API

Une API REST (également appelée API RESTful) est une interface de programmation d'application (API ou API web) qui respecte les contraintes du style d'architecture REST et permet d'interagir avec les services web RESTful. L'architecture REST (Representational State Transfer) a été créée par l'informaticien Roy Fielding [18]. Cette architecture est utilisée dans le projet pour élaborer la communication entre les grandes parties de l'application (Back-end et Front-end).

3.5 Persistance des données

3.5.1 Choix de la base de données

Afin de gérer les données de l'application, nous avons utilisé le NoSQL (not only SQL), avec MongoDB.

Le plus grand avantage de MongoDB par rapport au SQL, c'est sa capacité à gérer des systèmes de données complexes. On peut avoir des listes, des objets encapsulés sans avoir de soucis. Les données sous MongoDB sont représentées en des documents. Ce fonctionnement facilite grandement le développement d'applications qui gèrent beaucoup de données. Le NoSQL impose beaucoup moins de contrainte et est plus léger. De plus, il est plus adapté aux méthodologies de travail itératives et orientés prototypes que les bases de données SQL classiques. Un autre grand avantage est la possibilité d'accéder et d'utiliser les données le plus tôt possible. MongoDB ne nécessite pas de schéma prédéfini. Elle permet notamment d'avoir une architecture distribuée inter-serveur. Les documents sont créés sans les contraintes de relations et peuvent fonctionner indépendamment.

3.5.2 Object-Document Mapping avec Mongoose

Mongoose est un framework JavaScript populaire couramment utilisé dans les applications Node.js avec une base de données MongoDB. C'est en fait un mapeur de document objet (ODM). Cela signifie que Mongoose nous permet de définir des objets avec un schéma fortement typé mappé sur un document MongoDB. Mongoose fournit une quantité incroyable de fonctionnalités pour créer et utiliser des schémas. Ce framework contient actuellement plusieurs SchemaTypes sur lesquels les données sont enregistrées dans la base de données. Elles sont : String, Number, Date, Buffer, Boolean, Mixed, ObjectId, Array, Decimal128, Map et Schema.

3.6 Les outils

3.6.1 Les outils de conception

Draw.io (Diagrams.net) [1] : est une application gratuite en ligne, qui permet de dessiner des diagrammes ou des organigrammes. Cet outil nous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter. Nous avons utilisé Draw.io pour la conception des diagrammes de cas d'utilisation. La figure 3.5 montre le logo de Draw.io



FIGURE 3.5 – Logo de Draw.io

Lucidchart [9] : est une plateforme de collaboration en ligne, basée sur le cloud, permettant la création de diagrammes et la visualisation de données, et autres schémas conceptuels. La startup hébergeant Lucidchart a été lancée en décembre 2008, et est basée à Salt Lake City. Cette solution constitue une alternative à Microsoft Visio, avec lequel Lucidchart est compatible. Nous avons utilisé Lucidchart pour la création des diagrammes MVC détaillés. La figure 3.6 ci-dessous illustre le logo de la plateforme Lucidchart.

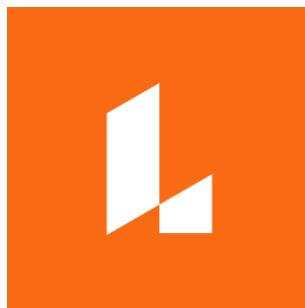


FIGURE 3.6 – Logo de Lucidchart

Visual Paradigm Online Diagrams [7] : est un outil de création de diagrammes basé sur le Web (en ligne) qui prend en charge un grand nombre de diagrammes. Elle propose une variété des formes triées par type de diagramme. Visual paradigm nous a aidé à la création des diagrammes de séquences. La figure 3.7 représente le logo de cet outil.



FIGURE 3.7 – Logo de Visual Paradigm Online Diagrams

3.6.2 Les outils de développement

Visual Studio Code [19] : est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. La figure 3.8 présente le logo de VS Code.

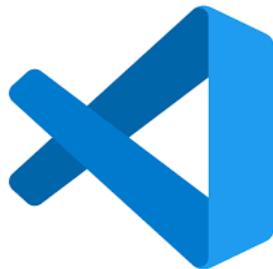


FIGURE 3.8 – Logo de Visual Studio Code

Postman [13] : permet de construire et d'exécuter des requêtes HTTP. Il facilite la procédure de tester les API et les services web, il est aussi largement utilisé pour la documentation. La figure 3.9 montre le logo de Postman.



POSTMAN

FIGURE 3.9 – Logo de Postman

Node.js [21] : est une plateforme logicielle libre en JavaScript, orientée vers les applications réseau événementielles hautement concurrentes qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle V8, la librairie libuv pour sa boucle d'évènements et implémente sous licence MIT les spécifications CommonJS. La figure 3.10 démontre le logo de Node.js.



FIGURE 3.10 – Logo de Node.js

MongoDB Compass [2] : propose une interface graphique pour MongoDB, nous permettant d'envisager de nouvelles perspectives de gestion de notre base de données NoSQL orientés documents. MongoDB Compass nous permet de garder la main sur nos données confidentielles ainsi que de nous connecter à un serveur MongoDB local ou également à MongoDB Atlas. La figure 3.11 démontre le logo de MongoDB Compass.

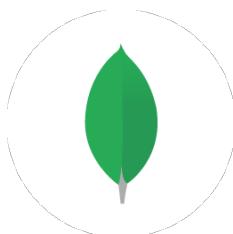


FIGURE 3.11 – Logo de MongoDB Compass

Bootstrap [12] : est une librairie open source qui permet de facilement concevoir une page Web avec du HTML, CSS et du javascript. Bootstrap est exclusivement utilisée pour le Front-end. Cet outil sert à faciliter le design de l'interface utilisateur afin de générer un site web adaptatif à la taille de l'écran. Cette librairie aide le développeur à économiser du temps en réduisant le temps nécessaire pour éditer les fichiers CSS. La figure 3.12 illustre le logo de Bootstrap.



FIGURE 3.12 – Logo de Bootstrap

3.6.3 Les outils de travail collaboratif

Slack [17] : est un outil de communication pour les entreprises qui désirent être plus efficaces. Slack a vu le jour il y a 8 ans, en août 2013. Slack est un acronyme pour « Searchable Log of All Conversation and Knowledge ». Cet outil nous a aidé énormément à la communication et la gestion des tâches grâce à la fonctionnalité d'intégration avec Jira. La figure 3.13 désigne le logo de slack.



FIGURE 3.13 – Logo de slack

Jira Software [14] : Développé par l'entreprise australienne Atlassian, est un logiciel principalement dédié à la gestion de projet de développement d'applications en mode agile. La figure 3.14 désigne le logo de Jira Software.



FIGURE 3.14 – Logo de Jira Software

Bitbucket [3] : est un service SaaS (Software as a Service), rebaptisé depuis Bitbucket Cloud, qui est bien plus qu'un simple outil de gestion de versions logicielles basé sur les systèmes open source de contrôle de révisions Git et Mercurial. Édité par la société australienne Atlassian, il englobe également un espace de gestion de projet, un gestionnaire de planification, un environnement de collaboration autour du code source, mais aussi des fonctions de test et de CI/CD, pour intégration et livraison continue. La figure 3.15 démontre le logo de Bitbucket.



FIGURE 3.15 – Logo de Bitbucket

3.7 Conclusion

Dans ce chapitre, nous avons présenté l'architecture logicielle et physique du site web ainsi que la manière comment garantir la persistance des données de l'application. Nous avons aussi démontré les frameworks de développement et les outils de conception. Le chapitre suivant sera consacré à la mise en œuvre du premier release de notre projet.

Chapitre 4

Mise en œuvre de la release 1

4.1 Introduction

Ce chapitre fait l'objet de l'introduction de la première version du projet, qui est intitulée "Version 1.0". Cette version se compose de deux sprints, chacune va prendre 2 semaines à travailler sur et elle va être consacrée, en gros, à l'authentification et la gestion des comptes.

4.2 Sprint 1 : Visite, création de compte et authentification

4.2.1 Sprint Planning

Sprint Goal

Durant ce sprint, nous allons travailler à réaliser les fonctionnalités permettant au visiteur de consulter le contenu du site, contacter l'administration et créer un compte pour rejoindre la plateforme et les fonctionnalités permettant à l'utilisateur de se connecter ou de se déconnecter.

Sprint Backlog

Le Backlog du premier sprint est illustré par le tableau 4.1 contenant une liste des tâches triées par ordre chronologique.

TABLE 4.1 – Backlog du sprint 1

Rang	Use case	User story	Estimation
1	Consulter contenu	En tant que visiteur, je veux consulter le contenu de la plateforme (page d'accueil, comment ça marche ?, ...) afin de s'informer.	16h
2	Contacter administrateur	En tant que visiteur, je veux contacter l'administration de la plateforme pour avoir des réponses à mes interrogations.	16h
3	Créer un compte	En tant que visiteur, je veux créer un compte pour bénéficier du contenu de la plateforme.	32h
4	Se connecter	En tant qu'utilisateur ou admin, je veux m'authentifier pour accéder à mon espace personnel.	24h
5	Se déconnecter	En tant qu'utilisateur, je veux terminer ma session intentionnellement.	8h

4.2.2 Conception

Diagramme MVC détaillé

La figure 4.1 illustre le diagramme MVC détaillé de 1er sprint. Il s'agit d'un diagramme composé de 3 couloirs :

- Le Model : où on retrouve le schéma de base de données nécessaire pour ce sprint.
- Le Controller : où on a cité les contrôleurs construits durant le sprint et contenant les différentes méthodes utilisées.
- Le View : où on trouve la liste des interfaces nécessaires pour accomplir le travail.

Diagramme d'interaction “Se connecter”

La figure 4.2 illustre le diagramme d'interaction du cas d'utilisation “Se connecter” qui est déclenché par un visiteur du site.

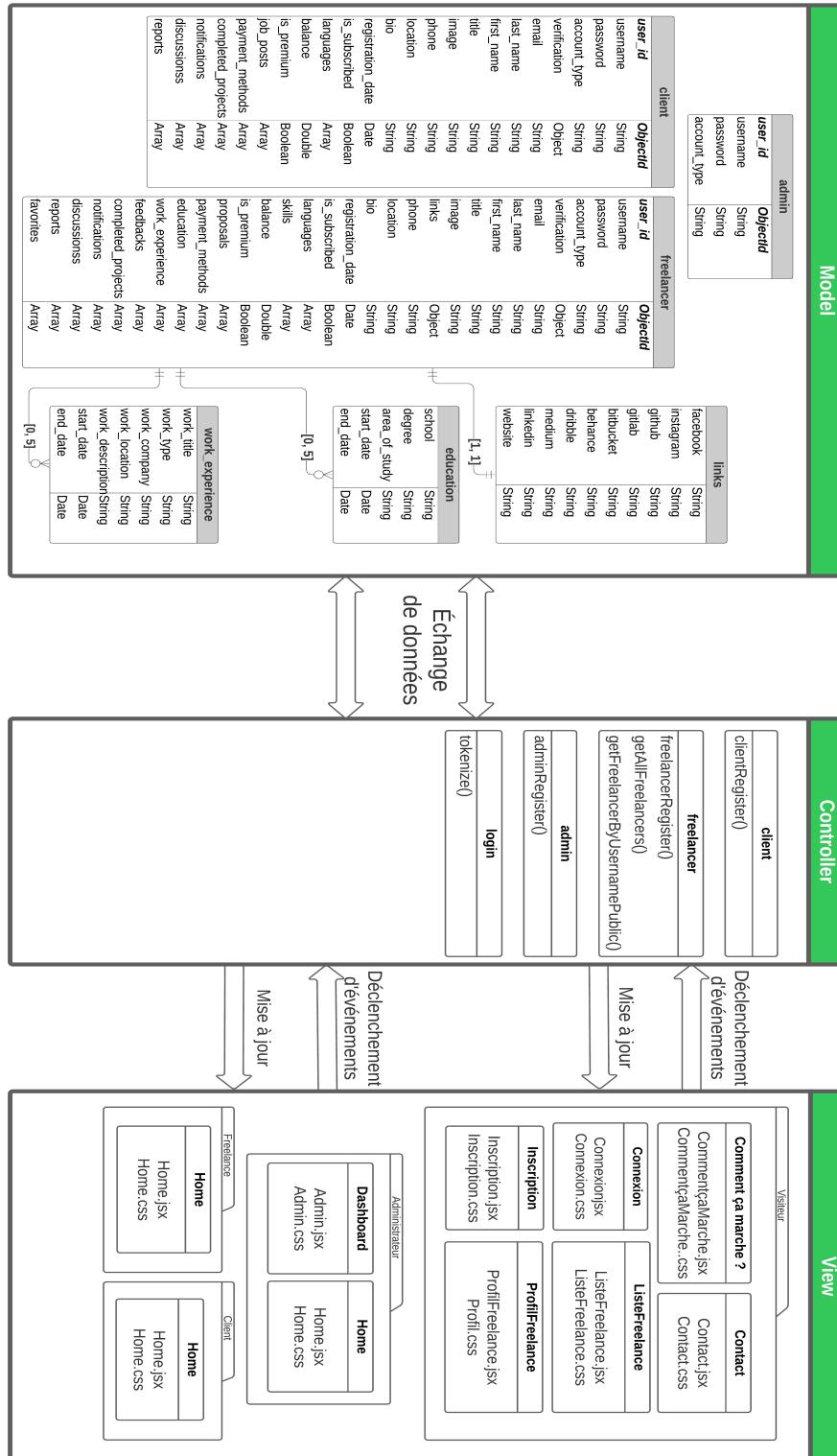


FIGURE 4.1 – Diagramme MVC détaillé Sprint 1

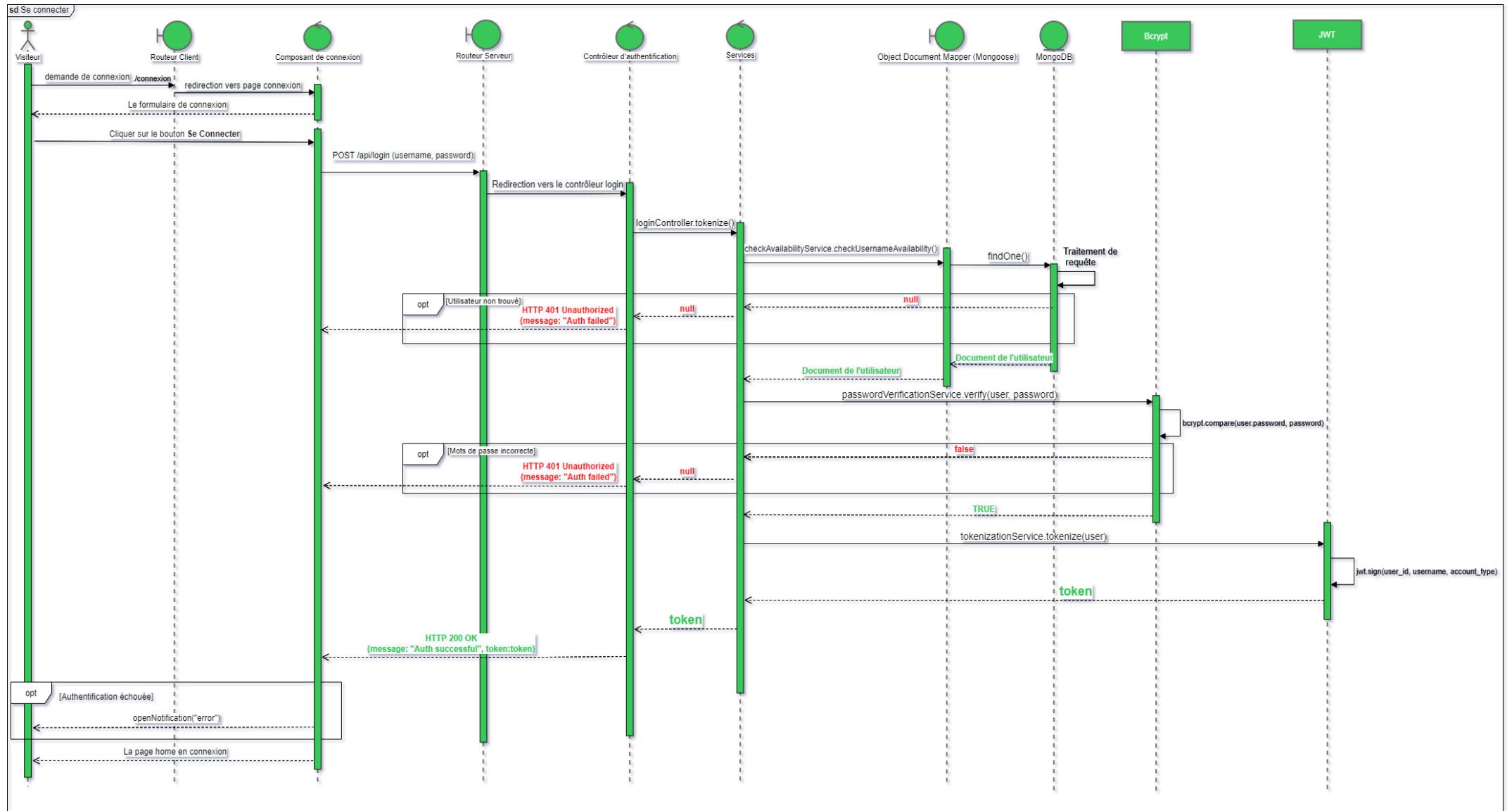


FIGURE 4.2 – Diagramme d'interaction "Se connecter"

L'acteur visiteur demande le formulaire d'inscription et ce formulaire est rendu par la vue “ Connexion”. Après avoir entré les coordonnées, une requête de type POST est envoyée au back-end et la fonction “tokenize()” sera exécutée. Elle retourne un status 200 et fait la redirection vers la page d'accueil si les données sont valides. Sinon, le status “401 Unauthorized” est retourné et le visiteur est invité à saisir de nouveau ses coordonnées.

4.2.3 Sprint Review

Afin de monter les résultats de ce sprint, nous allons présenter quelques captures des interfaces graphiques de l'application.

Interface "Page d'accueil"

La page d'accueil (figure 4.3) est consultée par tout visiteur de la plateforme. Elle contient des informations qui peuvent être utiles pour que l'internaute comprenne le concept général du site.

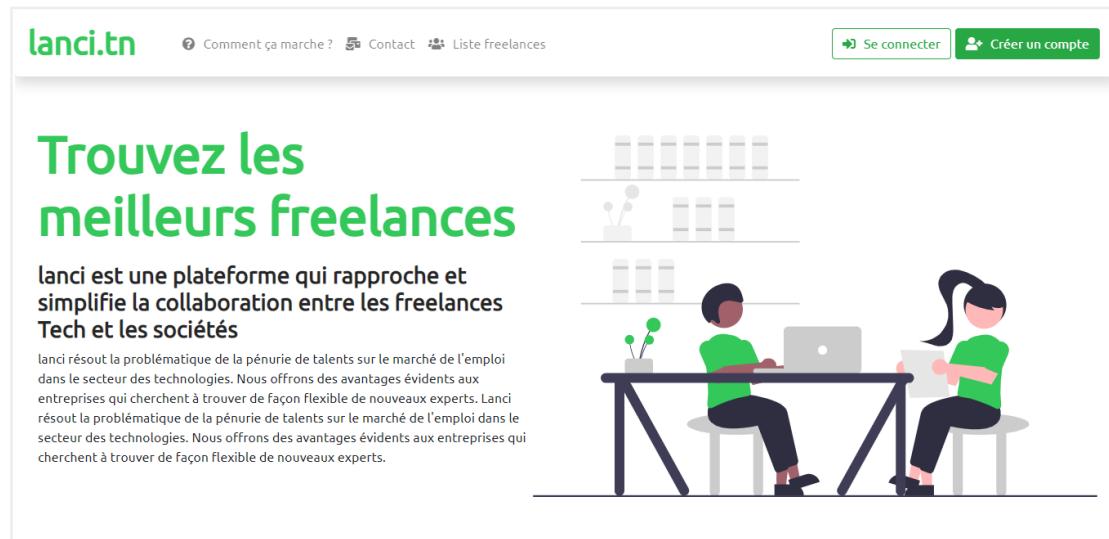
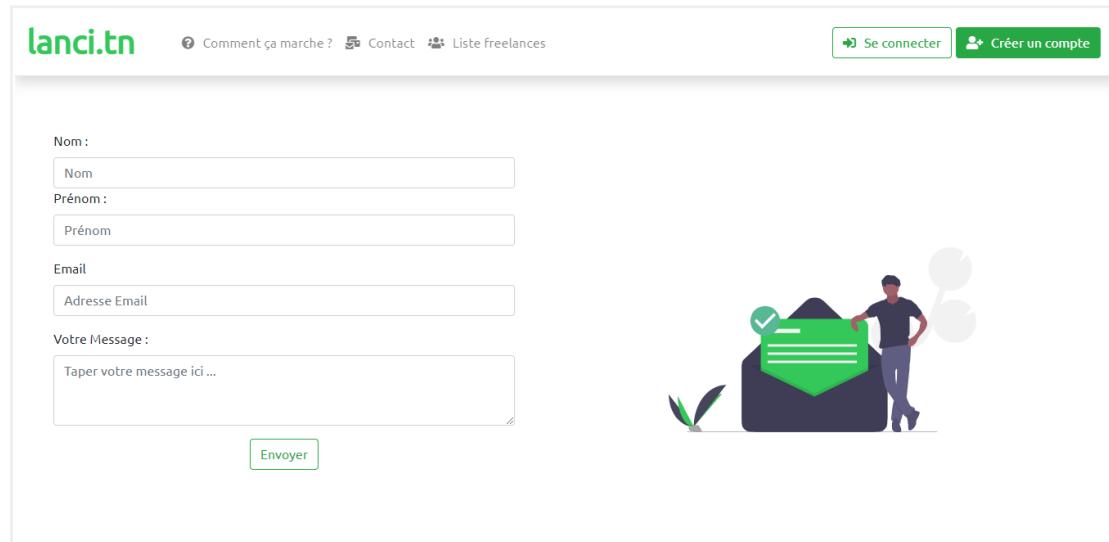


FIGURE 4.3 – Interface "Page d'accueil"

Interface “Contact administrateur”

Cette interface sert à contacter l'administrateur du site, les données remplies dans le formulaire seront envoyées par mail à l'admin (admin@lanci.tn). Cette interface est illustrée par la figure 4.4.

CHAPITRE 4. MISE EN ŒUVRE DE LA RELEASE 1

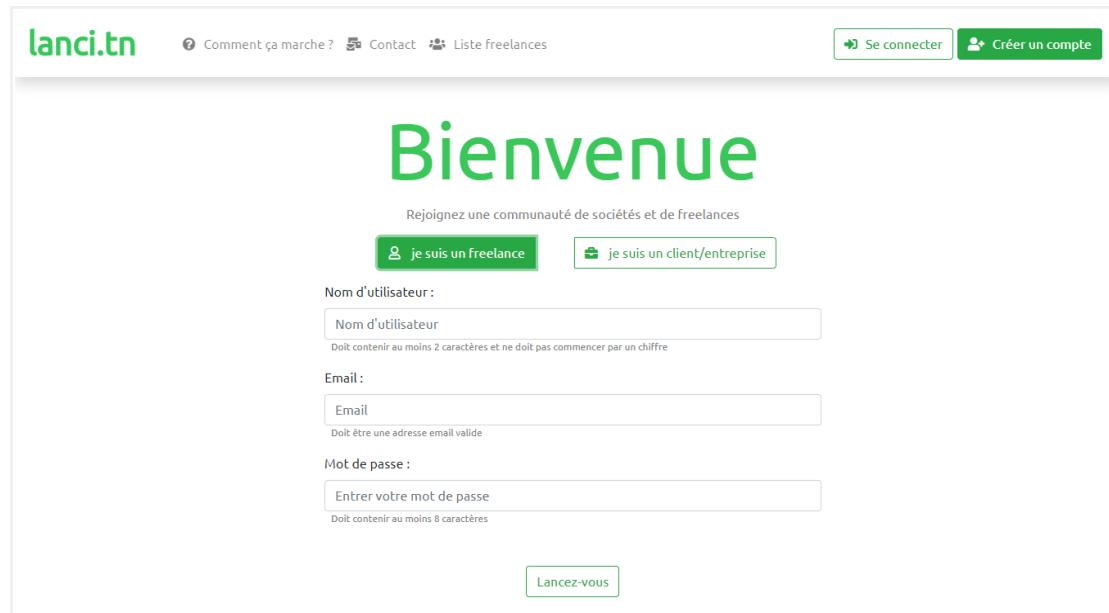


The screenshot shows the 'Contact administrateur' (Administrator Contact) page of the lanci.tn website. At the top right are links for 'Se connecter' (Log in) and 'Créer un compte' (Create account). The main form contains fields for 'Nom' (Name), 'Prénom' (First name), 'Email', and 'Votre Message' (Your message). Below the message field is a placeholder 'Taper votre message ici ...'. A green 'Envoyer' (Send) button is at the bottom left. To the right is a decorative illustration of a person standing next to a large envelope with a checkmark.

FIGURE 4.4 – Interface “Contact administrateur”

Interface “Inscription”

La figure 4.5 ci-dessous montre le formulaire de création d'un compte. Le visiteur est invité à remplir ce formulaire pour joindre la plateforme soit en tant que freelance ou bien en tant qu'un client/entreprise.



The screenshot shows the 'Inscription' (Sign-up) page of the lanci.tn website. At the top right are links for 'Se connecter' (Log in) and 'Créer un compte' (Create account). The main area features a large green 'Bienvenue' (Welcome) heading. Below it is a sub-headline 'Rejoignez une communauté de sociétés et de freelances' (Join a community of companies and freelancers). Two buttons are present: 'je suis un freelance' (I am a freelancer) and 'je suis un client/entreprise' (I am a client/enterprise). The form includes fields for 'Nom d'utilisateur' (Username), 'Email', and 'Mot de passe' (Password). Each field has a descriptive placeholder and a note below it. A green 'Lancez-vous' (Launch) button is at the bottom.

FIGURE 4.5 – Interface “Inscription”

Interface “Connexion”

La page Connexion illustrée ci-dessous (figure 4.6) contient un formulaire de connexion. L’abonné doit entrer son nom d’utilisateur et son mot de passe afin de continuer.

The screenshot shows the lanci.tn login page. At the top, there is a navigation bar with links for 'Comment ça marche ?' (How it works), 'Contact', 'Liste freelances', 'Se connecter' (Log in), and 'Créer un compte' (Create account). The main heading 'Salut, ça va ?' is displayed prominently. Below it, a sub-heading says 'Bienvenue sur lanci.tn , veuillez vous connecter à votre compte.' A form is present for entering login credentials: 'Nom d'utilisateur:' (User name:) followed by an input field containing 'Nom d'utilisateur', 'Mot de passe:' (Password:) followed by an input field containing '*****', and a link 'Mot de passe oublié?' (Forgot password?). A 'Se Connecter' (Log in) button is located below the password field. At the bottom of the form, a message reads 'Vous n'avez pas encore un compte ! Créez le votre par ici' (You don't have an account yet! Create your own here) next to a 'Créer un compte' button.

FIGURE 4.6 – Interface “Connexion”

4.2.4 Sprint Retrospective

Ce qui a bien fonctionné

Dans ce premier sprint, nous avons effectué l’analyse, la conception et la réalisation du premier sprint. Les nouvelles fonctionnalités sont :

- Le visiteur peut s’inscrire, contacter l’administrateur et se connecter. Il est aussi capable de consulter le contenu public de la plateforme tel que la page “Comment ça marche ?” et “À propos”,
- Le client, le freelance ou l’admin connecté peuvent se déconnecter.

Ce qui est à améliorer

Dans ce sprint, nous avons implémenté la fonctionnalité d’inscription à la plate-forme. Cette fonctionnalité peut être améliorée par l’ajout de la possibilité de créer son compte en se basant sur les informations récupérées de son compte Facebook, Google ou LinkedIn. Par conséquent, l’authentification aussi sera possible à travers d’autres comptes.

TABLE 4.2 – Backlog du sprint 2

Rang	Use case	User story	Estimation
1	Gestion compte	En tant qu'utilisateur, je veux modifier mes paramètres afin d'avoir un contrôle sur mon compte.	16h
2	Gérer Profils	En tant qu'admin, je veux gérer les profils des utilisateurs afin d'avoir un contrôle sur les données.	16h
3	Consultation freelances	En tant que visiteur ou utilisateur, je veux parcourir les freelances et consulter le profil qui m'intéresse.	8h
4	Gérer Profil freelance	En tant que freelance, je veux gérer mon propre profil pour ajouter plus d'informations sur ma carrière.	16h
5	Gérer Profil client	En tant que client, je veux gérer mon propre profil pour modifier mes informations.	16h
6	Gérer les utilisateurs	En tant qu'administrateur, je veux gérer les comptes des utilisateurs (modifier et supprimer).	16h
7	Gérer compte admin	En tant qu'admin, je veux gérer mon compte en modifiant mon mot de passe.	8h

En plus, une fonctionnalité de réinitialisation du mot de passe en cas d'oubli par l'utilisateur sera vraiment utile.

4.3 Sprint 2 : Gestion des comptes et des profils

4.3.1 Sprint Planning

Sprint Goal

L'objectif de ce sprint consiste à implémenter les fonctionnalités permettant à l'utilisateur de gérer son compte et son profil, à l'administrateur du site de gérer les utilisateurs de la plateforme, ainsi, au client de consulter la liste des freelances, et enfin, le freelance de mieux gérer son profil en modifiant ses informations spéciales.

Sprint Backlog

Le Backlog du sprint 2 est illustré par le tableau 4.2 contenant une liste des tâches identifiées qui devront être réalisées avant la fin du sprint.

4.3.2 Conception

Diagramme MVC détaillé

La figure 4.7 représente le diagramme MVC détaillé de 2ème sprint.

Diagramme de séquence détaillé “Modifier profil freelance”

La figure 4.8 illustre le diagramme de séquences “Modifier profil freelance” qui présente l’interaction entre le freelance et le système lors de la modification de ses informations sur son profil.

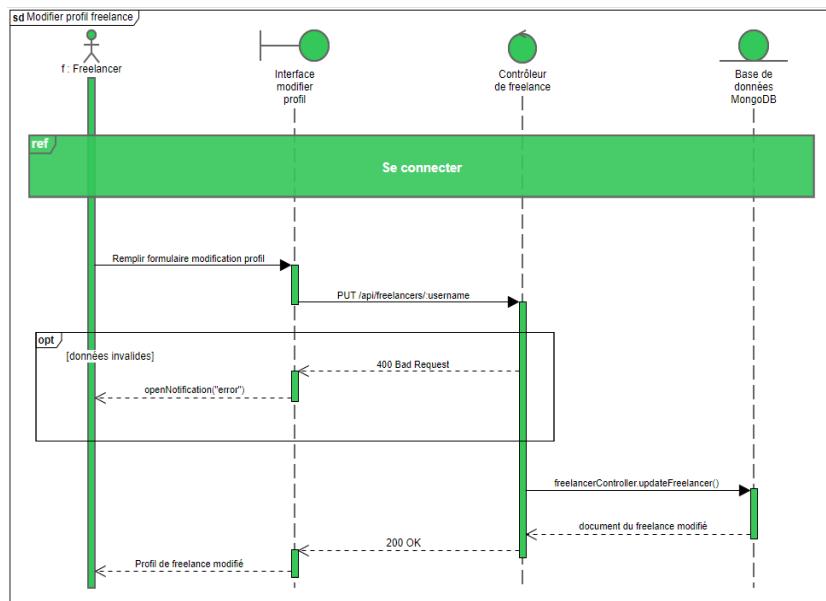


FIGURE 4.8 – Diagramme de séquences détaillé “Modifier profil freelance”

Le freelance interagit avec la vue “Modifier profil” afin de modifier ses informations. Après le remplissage du formulaire et la validation, la demande est envoyée vers la partie back-end qui vérifie tout d’abord que l’acteur a l’autorisation d’exécuter cette requête grâce au mécanisme JWT. Ensuite, la fonction “ updateFreelancer()” sera exécutée. En effet, l’ODM (Object Document Mapper) Mongoose traite le changement sur MongoDB à l’intérieur de cette fonction.

4.3.3 Sprint Review

Dans cette partie nous allons mettre en valeur les parties du projet réalisées dans ce sprint en présentant quelques interfaces.

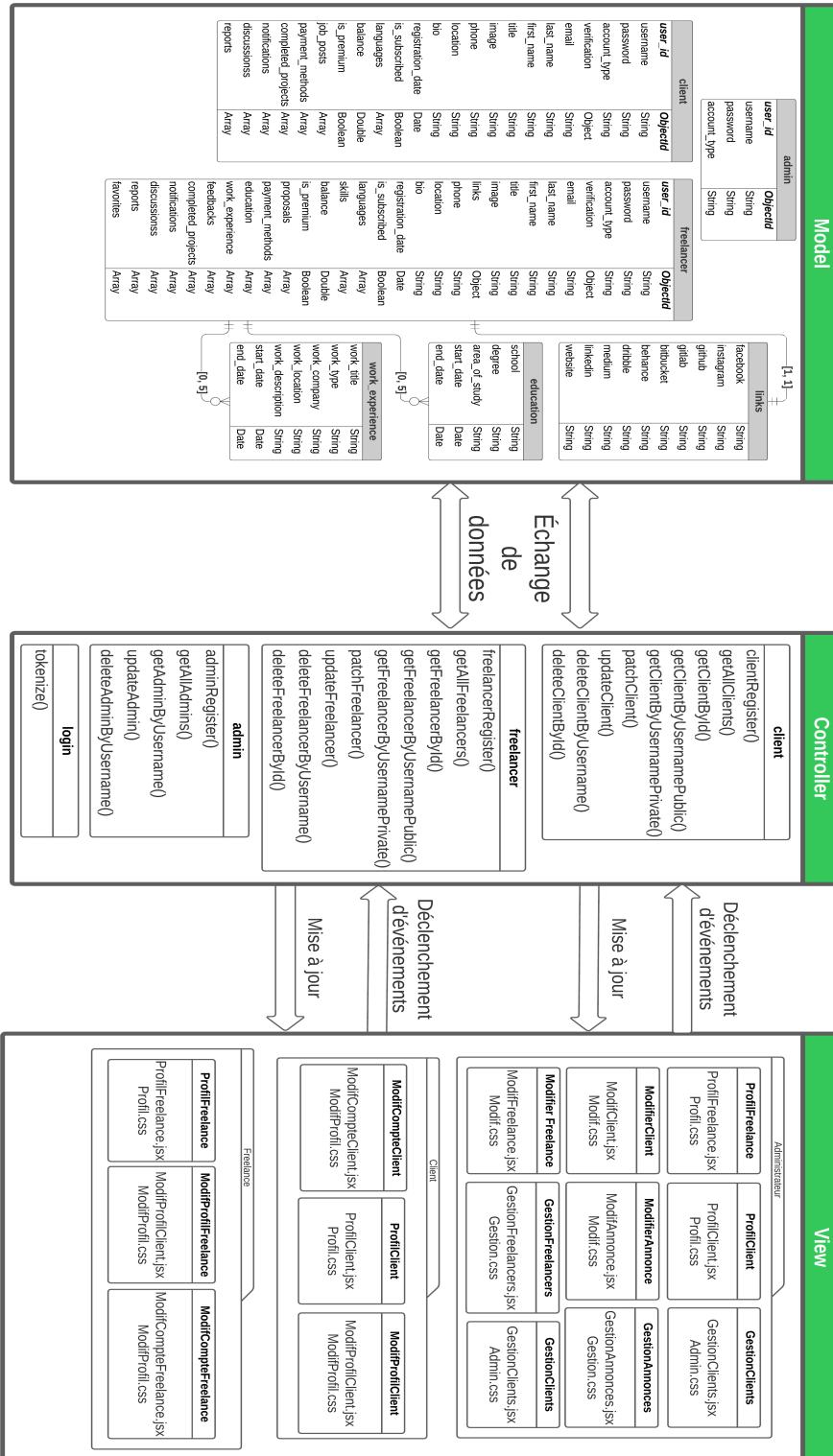


FIGURE 4.7 – Diagramme MVC détaillé Sprint 2

Interface “Liste freelances”

Cette interface (figure 4.9) illustre la liste des freelances inscrits à la plateforme. L'internaute peut consulter le profil de chaque freelance séparément.

The screenshot shows a web application interface for lance.tn. At the top, there is a navigation bar with links: "Comment ça marche?", "Contact", "Liste freelances", "Se connecter" (login), and "Créer un compte" (create account). Below the navigation, there is a grid of nine cards, each representing a freelance profile:

Profile Picture	Name	Occupation
	Fathi B.	Développeur Mobile (Flutter)
	Hasni M.	Data scientist
	Oussama B.	Data analyst
	Salah S.	Développeur d'applications mobiles (React Native)
	Samira B.	Motion graphic designer
	Wassim B.	Développeur C#
	Wissem B.	Développeur PHP
	Youssef Y.	Développeur Java
	Zied Z.	Graphic designer

Each card includes a small "eye" icon and a "chat" icon at the bottom right.

FIGURE 4.9 – Interface "Liste freelances"

Interface “Profil”

La figure 4.10 ci-dessous illustre l'interface profil d'un utilisateur (client ou freelance) qui est accessible à tout internaute.

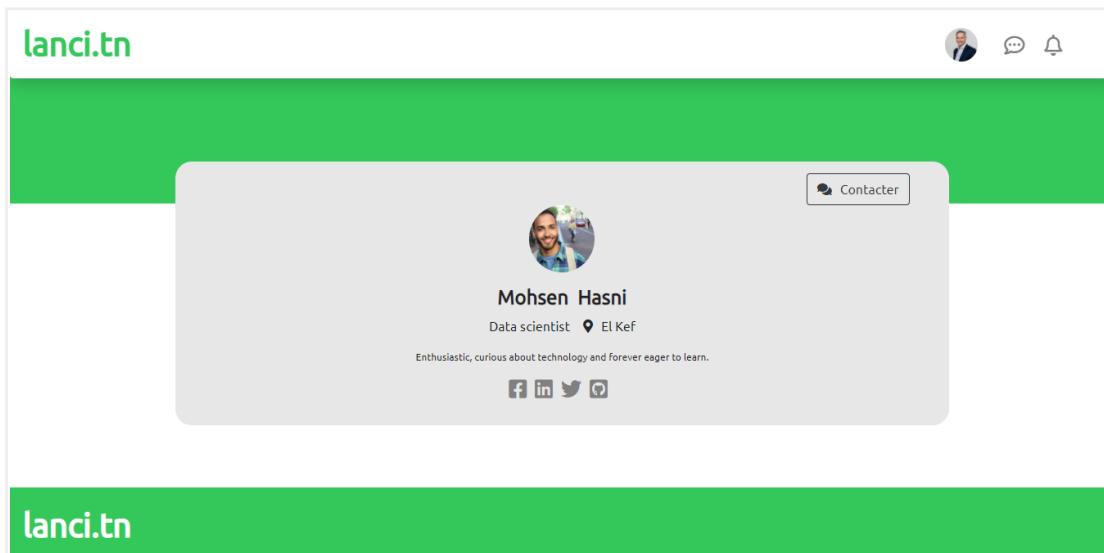


FIGURE 4.10 – Interface “Profil”

Interface “Gestion compte”

Cette interface (figure 4.11) illustre la page où l’utilisateur peut gérer son compte. Il lui permet de changer son e-mail ou son mot de passe.

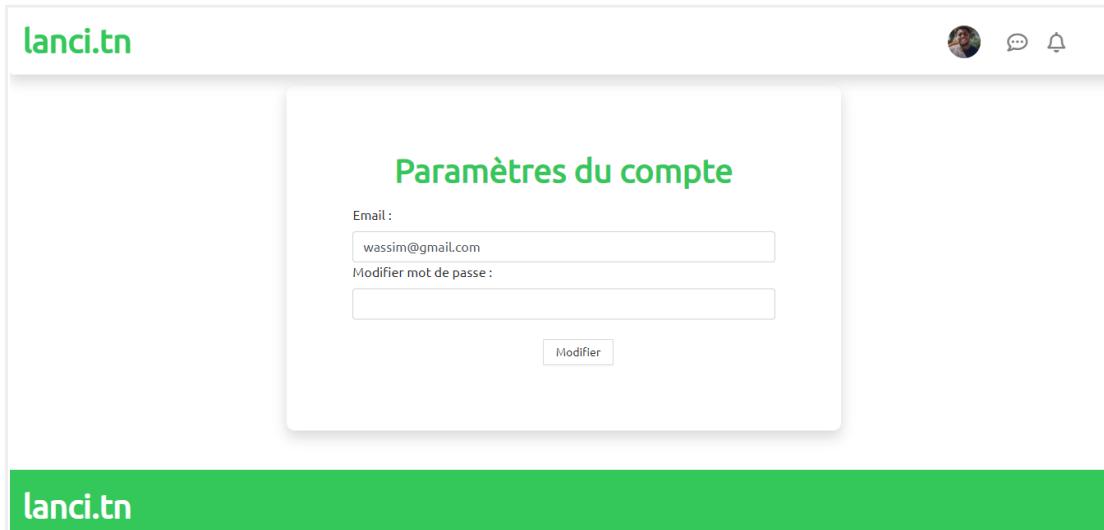


FIGURE 4.11 – Interface “Gestion compte”

Interface “Gérer profil freelance” (Freelance)

La figure 4.12 illustre la page modification des informations du profil qu’un freelance peut l’utiliser pour modifier ses informations générales.

The screenshot shows the lance.tn platform's profile management interface. On the left, there is a form for updating personal information:

- Nom d'utilisateur: samira
- Photo de profil: A placeholder image with a "Choisir une image" button.
- Nom: Ben Mostfa
- Prénom: Samira
- Date naissance: 1996-06-15
- Téléphone: 29292929
- Ville: Tunis
- Titre: (empty dropdown)

On the right, there is a "Liste des liens:" section displaying social media and professional links:

Icone	Link
	Facebook
	https://linkedin.com/in/samira
	https://twitter.com/samira
	Instagram
	GitHub
	GitLab
	Bitbucket
	https://behance.com/samira
	https://dribbble.com/samira
	Medium

FIGURE 4.12 – Interface “Gérer profil freelance” (Freelance)

Interface “Gérer freelances” (Admin)

La figure 4.13 ci-dessous montre la page de gestion des freelances par un administrateur de site.

id	Nom d'utilisateur	Email	Action
60d7667e14e24e2c9d762e8c	fathi	fathi@gmail.com	Consulter Modifier Supprimer
60d747f614e24e2c9d762a3b	mohsen	mohsen@gmail.com	Consulter Modifier Supprimer
60d767e814e24e2c9d762ef6	oussama	oussama@gmail.com	Consulter Modifier Supprimer
60d749bf14e24e2c9d762a84	salah	salah@gmail.com	Consulter Modifier Supprimer
60d768b914e24e2c9d762f0a	samira	samira@gmail.com	Consulter Modifier Supprimer
60d74e0c14e24e2c9d762bd4	wassim	wassim@gmail.com	Consulter Modifier Supprimer

FIGURE 4.13 – Interface “Gérer freelances” (Admin)

4.3.4 Sprint Retrospective

Ce qui a bien fonctionné

Au long de ce sprint, nous avons réalisé toutes les tâches. À ce niveau-là,

- L’administrateur peut gérer les utilisateurs (consultation, modification et suppression). Il peut aussi modifier les paramètres de son propre compte.
- Toute freelance ou client peut gérer son compte et son profil.
- Le client peut consulter la liste des freelances et les deux (client ou freelance) peuvent consulter le profil qui les intéresse séparément.

Ce qui est à améliorer

On vise à ajouter le téléchargement (upload) du Curriculum Vitae (CV) 2.0. Ce CV nouvelle génération est multimédia, personnalisable, dynamique, évolutif et multisupports (Texte, vidéo...).

4.4 Conclusion

Pendant cette release, nous avons effectué l’analyse, la conception et la réalisation de sprint 1 “Visite, création de compte et authentification” et de sprint 2 “Gestion des comptes et des profils”. À la fin de ce dernier sprint, les principaux objectifs ont été bien achevés.

Chapitre 5

Mise en œuvre de la release 2

5.1 Introduction

Dans ce chapitre, on s'intéresse à produire la version 2.0 du projet. Cette version se concentre sur les annonces des missions et est composé du sprint 3 qui avait une durée de deux semaines.

5.2 Sprint 3 : Annonce et offres

5.2.1 Sprint Planning

Sprint Goal

Durant ce sprint, nous allons réaliser les fonctionnalités permettants de gérer les annonces des missions par le client propriétaire d'annonce ou par l'administrateur de la plateforme. La consultation des annonces est effectuée par tous les utilisateurs du site. Un freelance peut proposer une offre sur une annonce. Il a la possibilité de gérer ses offres (consulter ou supprimer). Enfin, le client pourra consulter la liste des offres sur l'une de ses annonces et décide l'approbation ou le rejet de cette offre.

5.2.2 Sprint Backlog

Le Backlog du troisième sprint est illustré par le tableau 5.1 contenant une liste des tâches triés par ordre chronologique.

TABLE 5.1 – Backlog du sprint 3

Rang	Use case	User story	Estimation
1	Gérer mes annonces missions	En tant que client, je veux gérer mes annonces (ajouter, consulter, modifier ou supprimer).	16h
2	Consulter annonces missions	En tant qu'utilisateur, je veux consulter les annonces de missions pour avoir une idée sur le marché interne de la plateforme.	8h
3	Proposer une offre sur annonce mission	En tant que freelance je veux proposer une offre sur une annonce afin de gagner du travail.	16h
4	Consulter offres	En tant que client je veux consulter les offres proposées sur l'une de mes missions afin de prendre une décision (accepter ou rejeter une offre).	8h
5	Gérer mes offres	En tant que freelance, je veux gérer mes offres (consulter ou supprimer une offre).	24h
6	Gérer les annonces missions	En tant qu'admin, je veux gérer les annonces missions (consulter, modifier ou supprimer).	24h

5.2.3 Conception

Diagramme MVC détaillé

La figure 5.1 illustre le diagramme MVC détaillé du 3ème sprint.

CHAPITRE 5. MISE EN ŒUVRE DE LA RELEASE 2

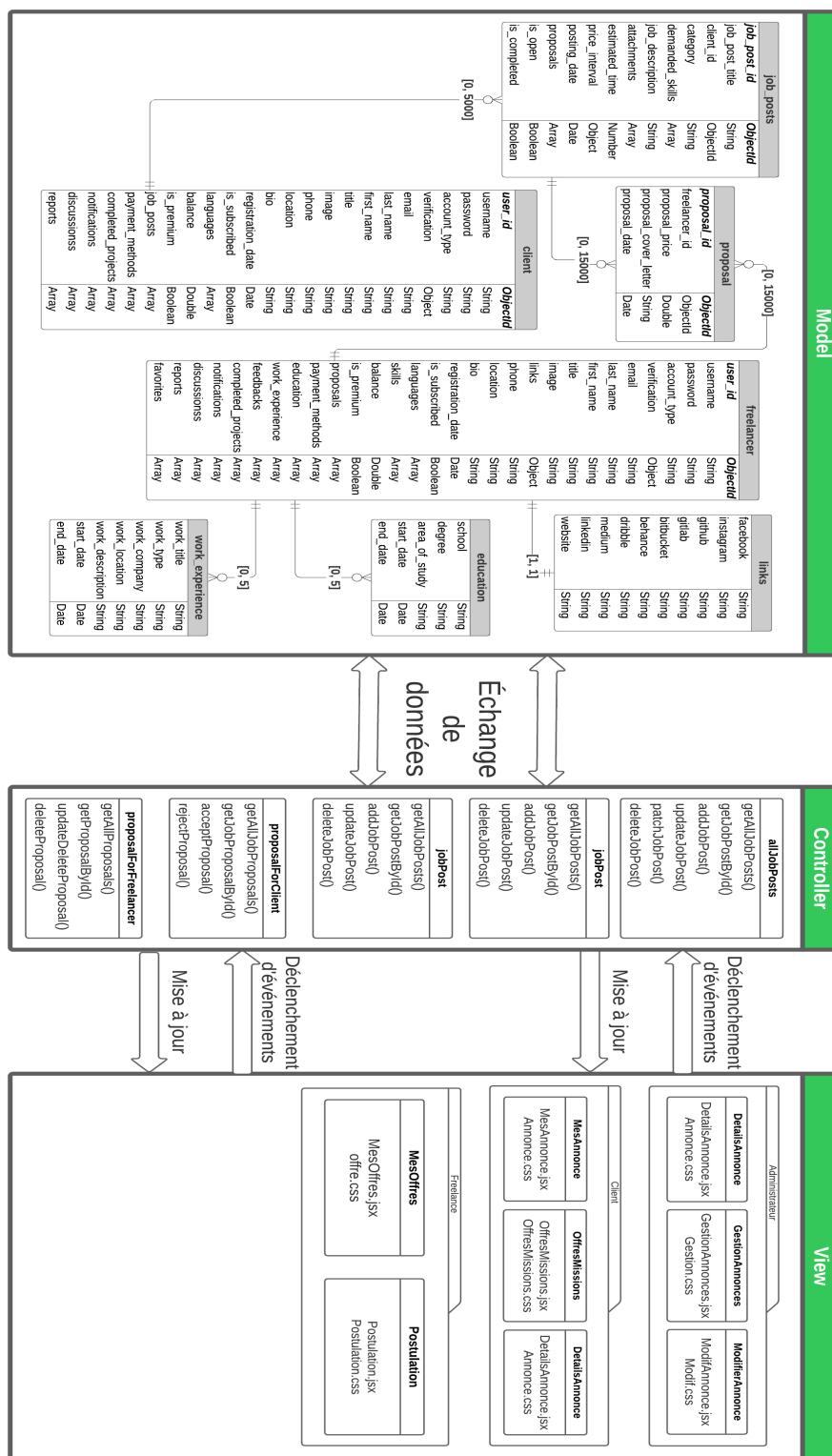


FIGURE 5.1 – Diagramme MVC détaillé Sprint 3

Diagramme de séquence détaillé “Ajouter annonce”

La figure 5.2 illustre le diagramme de séquences “Ajouter annonce” qui illustre l’interaction entre le client et le système lors de l’ajout d’une nouvelle annonce.

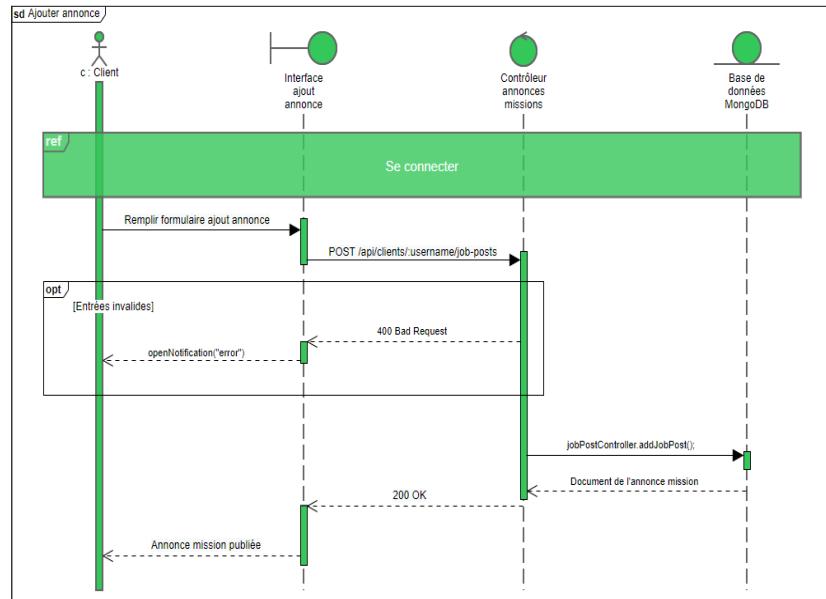


FIGURE 5.2 – Diagramme de séquences détaillé “Ajouter annonce”

Le client interagit avec la vue “Ajout annonce” en demandant l’ajout d’une annonce à ce diagramme. Après l’entrée des données demandées, la demande est envoyée vers la partie back-end pour vérifier l’autorisation de l’acteur grâce au mécanisme JWT. Ensuite, la fonction “addJobPost()” est exécutée. L’ODM Mongoose map la requête et traite l’ajout sur MongoDB. Enfin, l’annonce est ajoutée et la redirection se fait vers la page des annonces.

Diagramme de séquence détaillé “Proposer une offre”

La figure 5.3 illustre le diagramme de séquences “Proposer une offre” qui illustre l’interaction entre le freelance et le système lors de la proposition d’une offre sur une annonce de mission.

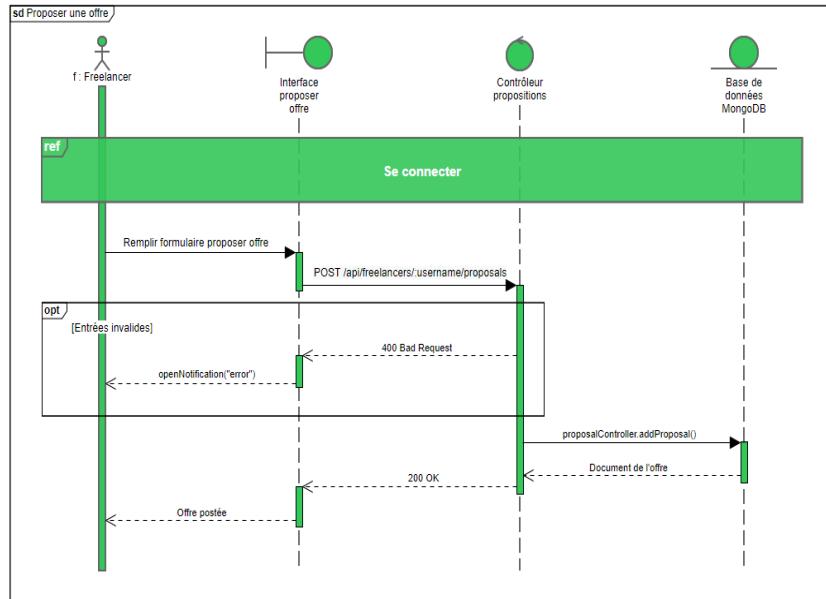


FIGURE 5.3 – Diagramme de séquences détaillé “Proposer une offre”

Le freelance interagit avec la vue “Proposer une offre” en demandant en remplissant une formulaire sur la page “détails annonces”. Après l’entrée des données demandées (valides), la demande est envoyée vers la partie back-end pour vérifier l’autorisation de l’acteur avec JWT. Ensuite, la fonction “addProposal()” est exécutée. L’ODM map la requête et traite l’ajout de l’offre sur MongoDB d’une manière similaire à l’ajout d’annonce à la partie précédente. Enfin, la proposition est ajoutée et une redirection vers la page home se produit.

5.2.4 Sprint Review

Interface “Ajouter une annonce”

L’interface présentée à la figure 5.4 montre la page d’ajout d’une nouvelle annonce de mission par un client.

CHAPITRE 5. MISE EN ŒUVRE DE LA RELEASE 2

Ajouter une annonce

Titre de l'annonce :

Description :

Décrivez votre mission ...

Marge de prix :

DT Prix minimum DT Prix maximum

Durée :

La durée de la mission (en jours)

Catégorie :

Choisir une catégorie

Entrer les compétences requises :

Compétences

Créer annonce

FIGURE 5.4 – Interface “Ajouter une annonce”

Interface “Gestion annonces (gérer mes annonces missions)”

L’interface présentée à la figure 5.5 montre la page de gestion des annonces.

Titre	Catégorie	Nb. offres	Action
Application de crypto monnaie	Développement Mobile	1	<button>Consulter</button> <button>Consulter offres</button> <button>Modifier</button> <button>supprimer</button>
Application de traduction	Développement Web	0	<button>Consulter</button> <button>Consulter offres</button> <button>Modifier</button> <button>supprimer</button>
Application de gestion de stock	Développement Mobile	0	<button>Consulter</button> <button>Consulter offres</button> <button>Modifier</button> <button>supprimer</button>
Application de réservation	Développement Mobile	1	<button>Consulter</button> <button>Consulter offres</button> <button>Modifier</button> <button>supprimer</button>

FIGURE 5.5 – Interface "Gestion annonces"

Interface “Détails annonce”

Cette interface figure 5.6 présente la page détails d’une annonce séparément avec toutes ses données pertinentes.

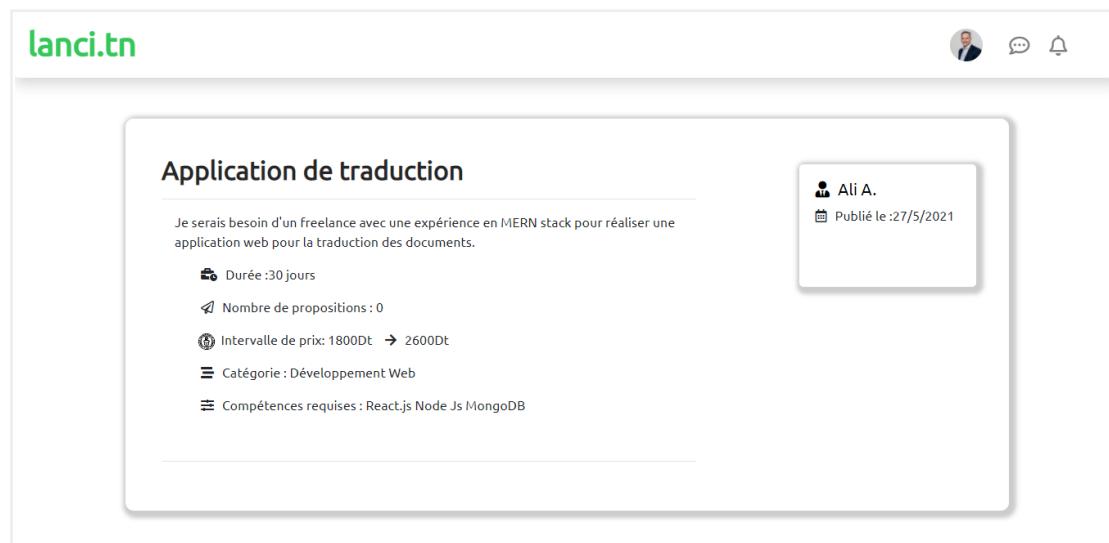


FIGURE 5.6 – Interface “Détails annonce”

Interface “Consulter annonces”

L’interface à la figure 5.7 montre la page home après la connexion où on trouve la liste de toutes les annonces publiées.

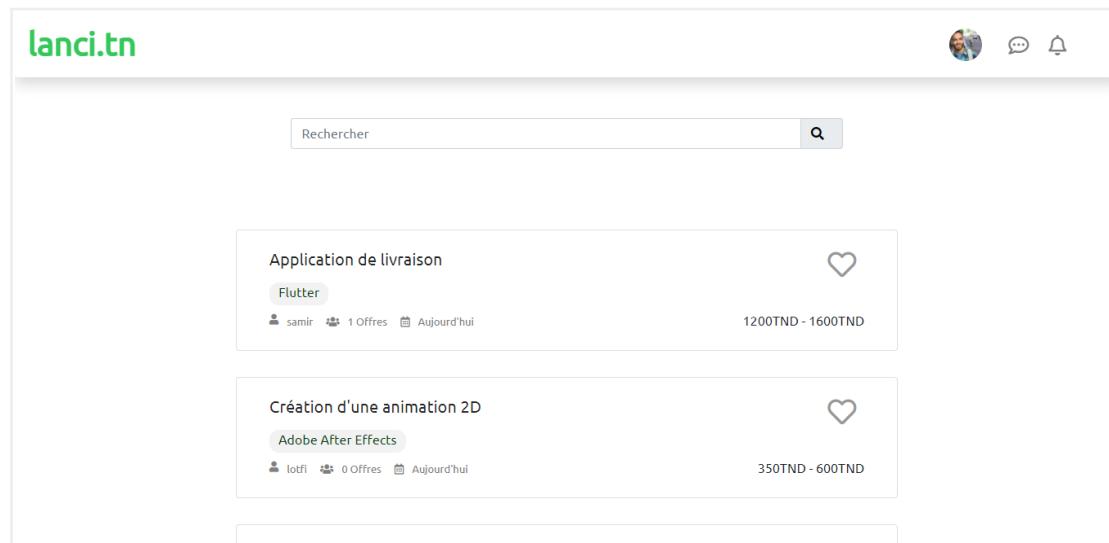


FIGURE 5.7 – Interface “Consulter annonces”

Interface “Proposer une offre”

La figure 5.8 montre l’interface de proposition d’une offre par un freelance (formulaire sous la page détails annonce).

The screenshot shows a web-based application interface for lanceur (freelancer). At the top, there's a navigation bar with the lanci.tn logo, user profile icon, and notification icons. Below the header, the main content area has a title "Votre offre" (Your offer). It contains several input fields: "Prix proposé:" (Price proposed), "Votre message:" (Your message), and "Date de livraison:" (Delivery date). There's also a text area labeled "Taper votre message ici ..." (Type your message here). A "Postuler" (Apply) button is located at the bottom of the form. The entire interface is set against a light gray background.

FIGURE 5.8 – Interface “Proposer une offre”

Interface “Mes offres (Freelance)”

La figure 5.9 est l’interface de gestion des offres d’un freelance.

The screenshot shows a web-based application interface for lanceur (freelancer). At the top, there's a navigation bar with the lanci.tn logo, user profile icon, and notification icons. Below the header, the main content area has a title "Mes offres" (My offers). It displays a table of submitted offers. The table has columns: "Annonce" (Advertisement), "Prix proposé" (Offered price), "Date de proposition" (Offer date), "Date de livraison" (Delivery date), and "Action" (Action). Two rows of data are shown: one for "Application de réservation" (Offered price 1400, Date 2021-06-26T23:35:50.082Z, Delivery date 2021-06-20T00:00:00.000Z) and another for "Application de crypto monnaie" (Offered price 2200, Date 2021-06-26T23:35:50.082Z, Delivery date 2021-06-29T00:00:00.000Z). Each row has two buttons: "Consulter annonce" (View offer) and "supprimer" (Delete). At the bottom of the table, there are navigation arrows for pagination. The entire interface is set against a light gray background.

FIGURE 5.9 – Interface “Mes offres (Freelance)”

Interface “Offres sur mission” (Client)

L’interface à la figure 5.10 présente la page des offres sur une mission qu’un client peut la consulter pour prendre sa décision.

Freelance	Message	Prix proposé	Action
salah	Je suis un développeur d'applications mobiles (React Native) et je serais content de réaliser votre application. En effet, j'ai travaillé dans un projet pareil il y a un ans.	1400	Consulter freelance Accepter Rejeter
fathi	Je suis un développeur d'application mobile avec une expérience de 2 ans. J'ai développé une application similaire pour la réservation.	1850	Consulter freelance Accepter Rejeter

FIGURE 5.10 – Interface “Offres sur mission” (Client)

Interface “Gérer annonces” (Admin)

L’interface à la figure 5.11 présente la page de gestion des annonces des mission par un administrateur.

id	Titre	Client	Action
60d7bb35d6b6da182d29f079	Application de livraison	samir	Consulter Modifier supprimer
60d7bc29d6b6da182d29f08c	Création d'une animation 2D	lotfi	Consulter Modifier supprimer
60d7bfe5d6b6da182d29f146	Application de crypto monnaie	ali	Consulter Modifier supprimer
60d7c3a8d6b6da182d29f1cb	Application de traduction	ali	Consulter Modifier supprimer
60d770ee7f4d917fc3f4965	Application de gestion de stock	ali	Consulter Modifier supprimer
60d7632214e24e2c9d762dc3	Application de réservation	ali	Consulter Modifier supprimer

FIGURE 5.11 – Interface “Gérer annonces” (Admin)

5.2.5 Sprint Retrospective

Ce qui a bien fonctionné

Au cours du sprint 3, nous avons réalisé toutes les tâches comme prévu. Les nouvelles fonctionnalités sont :

- Le client peut gérer ses missions (consulter, ajouter, modifier, supprimer). Il peut aussi consulter les offres postées sur chacune de ses annonces et faire son choix (accepter, rejeter).
- Le freelance peut proposer une offre à une annonce de mission. Il peut aussi gérer toutes ses offres sur les différentes annonces.
- L'administrateur peut gérer les annonces publiées sur la plateforme.

Ce qui est à améliorer

Nous pensons que la plateforme sera plus sécurisée et fiable si on ajoute un suivi de la réalisation d'une mission (tracking system) pour que le client soit au courant de l'avancement de son travail d'une part, et pour que le freelance prouve la bonne qualité de son service.

5.3 Conclusion

À la fin de ce sprint, on a terminé le release 2. Les principales fonctionnalités ont été réalisées avec succès, la version atteinte de l'application est stable et offre les fonctionnalités basiques qu'on doit retrouver dans ce type de plateforme.

Conclusion générale

Le présent rapport est réalisé dans le cadre de notre projet de fin d'études en vue de l'obtention du diplôme de Licence Appliquée Informatique (Systèmes informatiques et logiciels).

Ce travail avait pour but de concevoir et développer une application Web de travail indépendant (freelance). Afin d'accomplir notre mission, nous avons détaillé les différentes étapes d'analyse, de conception et de réalisation de ce système tous au long de 2 releases sous la méthodologie Agile Scrum.

La publication des annonces des missions en freelance, la proposition d'offre par le prestataire, la gestion des compte et des profil et enfin, l'administration de contenu sont les fonctionnalités prévues de notre application et ils ont été bien réalisées. On a achevé ce point de travail tout en respectant l'aspect sécuritaire et ergonomique et en préparant la documentation nécessaire. Cependant, nous n'avons pas terminé l'implémentation de certaines fonctionnalités tel que la messagerie et le paiement en ligne.

Ce travail était d'une importance considérable dans la mesure où il nous a permis d'approfondir nos connaissances pratiques en termes de conception et développement web à travers une panoplie de technologies comme Node.js, React.js et MongoDB et d'apprendre les bonnes pratiques de programmation notamment la sécurisation des données et la documentation des codes sources.

Ce projet était une véritable opportunité pour apprécier et enrichir notre formation puisqu'il nous a aidé à consolider nos acquis scientifiques et technologiques. Ainsi, il nous a aidé à être beaucoup plus à l'aise de travailler au sein d'une équipe et d'échanger, constamment, les idées pour avancer et pour s'évoluer d'une manière efficace.

Webographie

- [1] *diagrams.net - FREE FLOWCHART MAKER et diagrams ONLINE. Flowchart Maker & Online Diagram Software.* Diagrams.net. 2021. URL : <https://app.diagrams.net/> (visité le 21/03/2021).
- [2] *Compass.* MongoDB. 2021. URL : <https://www.mongodb.com/products/compass> (visité le 20/03/2021).
- [3] Antoine CROCHET-DAMAIS. *Bitbucket : l'outil Git de gestion de projet et de livraison continue.* Journaldunet.fr. Sept. 2019. URL : <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1443816-bitbucket-l-outil-git-de-gestion-de-projet-et-de-livraison-continue/> (visité le 20/03/2021).
- [4] *Eligibility to Join and Use Upwork.* Upwork Customer Service & Support Upwork Help Center. 2015. URL : <https://support.upwork.com/hc/en-us/articles/211067778-Eligibility-to-Join-and-Use-Upwork>.
- [5] *Express Web Framework (Node.js/JavaScript) - Apprendre le développement web / MDN.* Mozilla.org. 2021. URL : https://developer.mozilla.org/fr/docs/Learn/Server-side/Express_Nodejs (visité le 25/03/2021).
- [6] *Freelance Tunisie Comment ça marche ?* Freelancetunisie.net. 2017. URL : https://www.freelancetunisie.net/Home/Comment_Ca_Marche.
- [7] *Ideal Modeling & Diagramming Tool for Agile Team Collaboration.* Visual-paradigm.com. 2021. URL : <https://www.visual-paradigm.com/> (visité le 20/03/2021).
- [8] *Le grand dictionnaire terminologique.* Gouv.qc.ca. 2018. URL : <http://gdt.oqlf.gouv.qc.ca/resultat.aspx?terme=scalability> (visité le 02/04/2021).
- [9] *Lucidchart overview / Lucidchart.* Lucidchart.com. 2021. URL : <https://www.lucidchart.com/pages/product> (visité le 20/03/2021).
- [10] *MA. iLancer - How It Works.* Ilancer.tn. 2020. URL : <https://www.ilancer.tn/how-it-works>.

WEBOGRAPHIE

- [11] *Manifesto for Agile Software Development*. Agilemanifesto.org. URL : <https://agilemanifesto.org/> (visité le 15/03/2021).
- [12] Mark OTTO. *Bootstrap*. Getbootstrap.com. 2021. URL : <https://getbootstrap.com/> (visité le 20/03/2021).
- [13] *Postman / The Collaboration Platform for API Development*. Postman. 2021. URL : <https://www.postman.com/> (visité le 20/03/2021).
- [14] La RÉDACTION. *Jira (gratuit) : de la gestion de projet à la digital workplace*. Journaldunet.fr. Sept. 2019. URL : <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1443890-jira-de-la-gestion-de-projet-agile-digital-workplace/> (visité le 02/04/2021).
- [15] 1&1 IONOS SARL. *Modèle en cascade*. IONOS Digitalguide. 1970. URL : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/modele-en-cascade/> (visité le 14/03/2021).
- [16] *Scrum Guide / Scrum Guides*. Scrumguides.org. 2020. URL : <https://scrumguides.org/scrum-guide.html> (visité le 15/03/2021).
- [17] SLACK. *Slack*. Slack. 2021. URL : <https://slack.com/intl/en-tn/> (visité le 20/03/2021).
- [18] *Une API REST, qu'est-ce que c'est ?* Redhat.com. 2021. URL : <https://www.redhat.com/fr/topics/api/what-is-a-rest-api> (visité le 02/04/2021).
- [19] *Visual Studio Code*. Visualstudio.com. Avr. 2016. URL : <https://code.visualstudio.com/> (visité le 20/03/2021).
- [20] *Why 3% : How Toptal Finds Top Talent*. Toptal.com. 2021. URL : <https://www.toptal.com/top-3-percent>.
- [21] Contributeurs aux projets WIKIMEDIA. *Node Js*. Wikipedia.org. Sept. 2010. URL : <https://fr.wikipedia.org/wiki/Node.js> (visité le 20/03/2021).

Résumé

Ce rapport a été rédigé dans le cadre de réalisation d'un projet de fin d'étude en vue d'obtenir le diplôme national de Licence Appliquée en Informatique. Il consiste à concevoir et développer une plateforme web de freelance. L'objectif de cette plateforme est d'assurer la mise en relation entre, d'une part, les porteurs de projets/d'idées qui ont des missions, qui peuvent être de différents types et échelles, et, d'autre part, des freelances qui disposent les qualifications et les compétences nécessaires pour les accomplir.

Mots clés : Freelance, Travail indépendant, React.js, Node.js, MongoDB.

Abstract

This report was prepared as part of a graduation project in order to get an Applied Bachelor's Degree in Computer Systems and Software. This project's main goal is to design and develop a freelance web platform which aims to connect, on one hand, clients who have ideas or projects (with several types and shapes) and want them to be implemented. And, on another hand, competitive, skilled and creative freelancers and individuals who can satisfy this demand.

Keywords : Freelance, Self-employment, React.js, Node.js, MongoDB.