

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE UNIVERSITE
DE SOUSSE

INSTITUT SUPERIEUR D'INFORMATIQUE
ET DES TECHNOLOGIES DE COMMUNICATION

المعهد العالي للإعلامية وتكنولوجيا الاتصالات



Rapport de stage de fin d'études

Présenté en vue de l'obtention du diplôme de Licence en Science de l'Informatique
(Computer Science)

Spécialité : Informatique et Multimédia

Une plateforme SaaS pour la mise en relation entre clients et architectes d'intérieur

Réalisé par :
Roua Mejdoub

Encadrant académique : M.Mohamed Faten Zhani

Encadrant professionnel : M.Amir Bouker

Société d'accueil



Année universitaire : 2023/2024

Institut Supérieur d'Informatique et des Technologies de Communication **ISITCOM**
Tél/Fax : +216 73 37 15 71 / +216 73 36 44 11

Table des matières

Introduction Générale	10
Chapitre 1 :	11
Cadre général du projet	11
1. Introduction	12
2. Organisme d'accueil.....	12
2.1 Présentation de Dundill	12
2.2 Fiche technique	12
2.3 Les Valeurs de Dundill.....	13
2.4 Organigramme :.....	13
2.4 Les Services	14
3. Présentation du Projet.....	14
3.1 Contexte Général	14
3.2 Problématique	14
3.3 Solution Proposée	14
4. Étude de l'Existant	14
4.1 Solutions Existante	14
4.2 Critiquedel'existant	19
5. Objectifs du projet	19
6. Méthodologie de gestion de projet	19
6.1. Introduction à la méthodologie Agile.....	19
6.2. Présentation des principaux frameworks Agile	19
6.3. Focus sur Scrum	20
6.4. Justification du choix de Scrum	21
6.5. Conclusion sur la méthodologie adoptée.....	22
7. Conclusion	22
Chapitre 2 :	24
Analyse et spécification des besoins	24
1. Introduction	25
2. Identification des besoins.....	25
2.1. Besoins Fonctionnels	25
2.2 Besoins Non Fonctionnels.....	26
3. Planification du projet avec Scrum.....	26

3.1. Equipe et rôles.....	26
3.2. Product backlog.....	26
3.3. Planification des sprints.....	28
4.Architecture générale de l'application	29
4.1 Architecture logique : Modèle MVC (Modèle-Vue-Contrôleur)	29
.....	29
4.2. Architecture physique (Déploiement et infrastructure).....	29
4.3. Justification des choix techniques	30
5.Méthode et langage de Modélisation utilisé	31
5.1. Définition d'UML et son utilité	31
5.2. Les diagrammes UML utilisés dans le projet	31
6.Conclusion	33
Chapitre 3 :	34
1.Introduction :	35
2.Backlog de Sprint 1	35
3.Spécification fonctionnelle :.....	36
3.1Diagramme de cas d'utilisation sprint 1	36
3.2. Analyse de cas d'utilisation « Créer compte »	36

Introduction Générale

Dans un contexte de digitalisation croissante, les attentes des particuliers et des professionnels concernant la création et la rénovation d'espaces intérieurs évoluent rapidement. L'avancée rapide des technologies web, l'accès facilité aux services en ligne, ainsi que le besoin d'optimiser les processus métiers, poussent le secteur de l'architecture d'intérieur à se transformer. Les plateformes digitales jouent un rôle clé en simplifiant la mise en relation, la gestion de projets et la collaboration à distance.

Ce projet s'inscrit dans cette dynamique d'innovation. Il consiste à développer une plateforme web SaaS (Software as a Service) destinée à faciliter la mise en relation entre clients et architectes d'intérieur. Cette solution vise à offrir un écosystème complet qui gère l'ensemble du processus, depuis le premier contact jusqu'à la réalisation du projet, incluant la gestion des devis, des abonnements et des échanges collaboratifs. La plateforme répond aux besoins spécifiques des utilisateurs tout en garantissant la sécurité et la confidentialité des données, grâce à une architecture flexible et des fonctionnalités avancées telles qu'une authentification sécurisée, un système de recommandation intelligent basé sur l'intelligence artificielle, une marketplace intégrée et un tableau Kanban.

L'objectif est de proposer une expérience utilisateur simple et intuitive, de faciliter la recherche et la sélection de professionnels qualifiés, et d'améliorer la gestion des projets pour tous les acteurs impliqués. De plus, le modèle SaaS assure évolutivité, accessibilité et maintenance aisée, tout en réduisant les coûts d'infrastructure pour les utilisateurs finaux.

Ce rapport présente le contexte et les enjeux du projet, l'analyse de l'existant, la méthodologie adoptée, ainsi que les choix techniques et fonctionnels qui ont guidé la conception et la réalisation de cette plateforme innovante.

Chapitre 1 : Cadre général du projet

1. Introduction

Ce chapitre a pour but de présenter l'organisme d'accueil et le contexte général du projet. Il décrit l'étude de l'existant, les limites qui se présentent pour déduire les objectifs, les solutions suggérées et la méthode utilisée.

2. Organisme d'accueil

2.1 Présentation de Dundill

Dundill se distingue en tant qu'entreprise axée sur les services informatiques et le développement de solutions adaptées aux besoins des entreprises. Avec une expertise diversifiée, elle se consacre à la création de sites web, à l'amélioration de la visibilité en ligne via le commerce électronique, au développement d'applications mobiles et web, elle est basée sur des valeurs :

- Innovation : Dundill s'engage continuellement à intégrer les nouvelles technologies
- Qualité : Dundill accorde une importance particulière à la qualité de ses produits et services.
- Collaboration : Dundill collabore étroitement avec ses clients pour une compréhension approfondie de leurs besoins, assurant ainsi des réponses efficaces.
- Adaptabilité : l'adaptabilité est une caractéristique fondamentale chez Dundill. Nous accueillons les changements, les retours et les ajustements afin de garantir que nos produits finaux correspondent parfaitement aux attentes de nos clients.



Figure 1 : Logo de Dundill

2.2 Fiche technique

Ce tableau illustre la fiche technique de Dundill :

Dénomination	Dundill
Année de création	2022
Siège social	Immeuble Gammoun Akouda Sousse
Direction générale	Ala Eddin Bouker
Site web	https://dundill.tn
Téléphone	+216 54 133 050
Secteur d'activité	Service informatique
Catégorie	Services informatique
Forme juridique	S.A.R.L

Tableau 1 : Fiche technique Dundill

2.3 Les Valeurs de Dundill

Dundill est guidée par des valeurs fondamentales qui orientent ses opérations et ses relations avec les clients et les développeurs :

- Excellence : Engagement à fournir des solutions de haute qualité qui répondent aux besoins spécifiques des clients.
- Innovation : Utilisation des technologies de pointe pour offrir des services à la pointe du progrès.
- Intégrité : Maintien d'une transparence totale dans toutes les interactions et transactions.
- Collaboration : Promotion d'un environnement de travail inclusif où chaque voix est valorisée.
- Engagement envers la satisfaction du client : Priorité donnée à la compréhension et à la satisfaction des besoins des clients.

2.4 Organigramme :



ORGANIGRAMME DE NOTRE ÉQUIPE DUNDILL

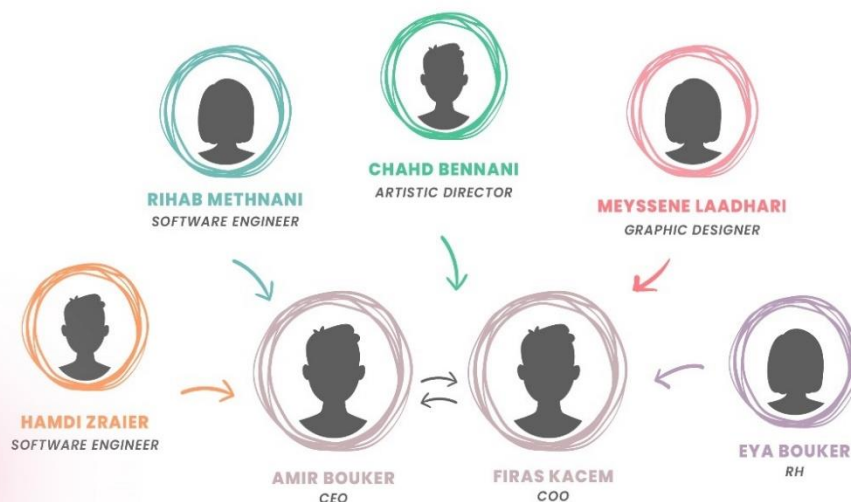


Figure 2 : Organigramme de Dundill

2.4 Les Services

Dundill offre une gamme complète de services visant à faciliter le recrutement et l'intégration de développeurs à distance :

- Recrutement de Développeurs à Distance : Identification et sélection rigoureuse de développeurs qualifiés pour répondre aux exigences technologiques des clients.
- Solutions Digitales Personnalisées : Développement de sites web dynamiques, d'applications mobiles intuitives et de solutions basées sur l'intelligence artificielle et la blockchain.
- Conseil en Stratégie Digitale : Fourniture de conseils stratégiques pour aider les entreprises à naviguer dans le paysage numérique en constante évolution.
- Support et Maintenance : Offre d'un soutien continu pour assurer le bon fonctionnement et la mise à jour des solutions technologiques mises en place.

Grâce à son expertise et à son engagement envers l'excellence, Dundill joue un rôle clé dans la transformation numérique des entreprises, en les aidant à atteindre leurs objectifs grâce à des solutions technologiques innovantes et sur mesure.

3. Présentation du Projet

3.1 Contexte Général

Les architectes d'intérieur et les designers ont besoin d'une plateforme pour présenter leurs services, leurs portfolios et leurs expertises de manière professionnelle. Les clients, quant à eux, cherchent des moyens efficaces pour trouver des professionnels qui correspondent à leurs besoins spécifiques. Cette plateforme vise à combler ce besoin en offrant un espace centralisé où les architectes peuvent exposer leurs travaux et où les clients peuvent filtrer et choisir les professionnels en fonction de leurs critères.

3.2 Problématique

Les particuliers ont du mal à trouver des architectes d'intérieur qualifiés et adaptés à leurs besoins. Les architectes, de leur côté, ont besoin d'une plateforme pour gérer efficacement leurs projets et acquérir de nouveaux clients.

3.3 Solution Proposée

Visiona une plateforme innovante conçue pour simplifier cette mise en relation. Elle permet aux clients de spécifier leurs besoins, de recevoir des devis et de comparer différents architectes. Pour les architectes, Visiona offre un moyen efficace de gérer les demandes et de suivre leurs projets.

4. Étude de l'Existant

4.1 Solutions Existantes

Afin d'être en mesure de proposer une solution fiable et adaptée aux attentes des utilisateurs, nous avons commencé par analyser les difficultés rencontrées par toute personne souhaitant collaborer avec un architecte d'intérieur, que ce soit pour un projet personnel ou professionnel. Il était essentiel de se documenter sur les solutions existantes afin de comprendre les limites des plateformes actuelles et d'identifier les axes d'amélioration possibles. Cette étape nous a permis d'évaluer l'éventail des outils déjà disponibles, d'en cerner les points forts et les points faibles, et ainsi de justifier la conception d'une nouvelle application plus complète et intelligente. Voici donc une illustration de quelques sites web représentatifs dans ce domaine.



Figure 3 – Logos de quelques plateformes analysées

4.1.1 Houzz

Plateforme internationale de référence en architecture d'intérieur, Houzz permet aux utilisateurs de consulter des portfolios d'architectes, de s'inspirer à travers des idées de design et d'acheter des produits pour la maison.

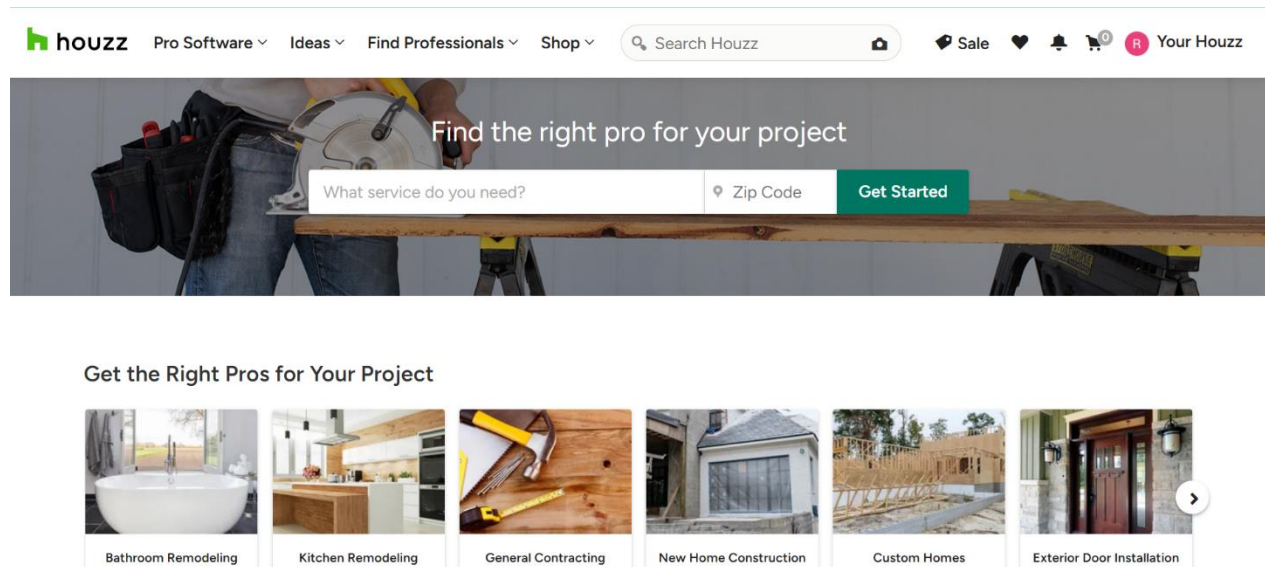


Figure 4– Page d'accueil de la plateforme Houzz

Avantages	Inconvénients
Très grande base de données d'architectes et de designers	Pas de système de matching intelligent
Design attrayant et expérience utilisateur fluide	Peu de suivi personnalisé entre client et professionnel
Marketplace intégrée pour l'achat de produits	Manque de gestion de projet (suivi, tâches, livrables)

• Tableau 2 : Synthèse des fonctionnalités de Houzz

4.1.2 Archidvisor

Plateforme française dédiée à la mise en relation avec des architectes certifiés. Elle propose un formulaire de besoin, une sélection manuelle par l'équipe, puis une mise en relation.

The screenshot displays the Archidvisor website interface for project description. At the top, the Archidvisor logo is on the left, and navigation links 'Mon projet', 'Mon bien', and 'Localisation' are in the center. Below this, a section titled 'Quel est votre projet ?' features six cards with icons and labels: 'Construction neuve', 'Rénovation', 'Aménagement intérieur', 'Extension', 'Surélévation', and 'Aménagement extérieur'.

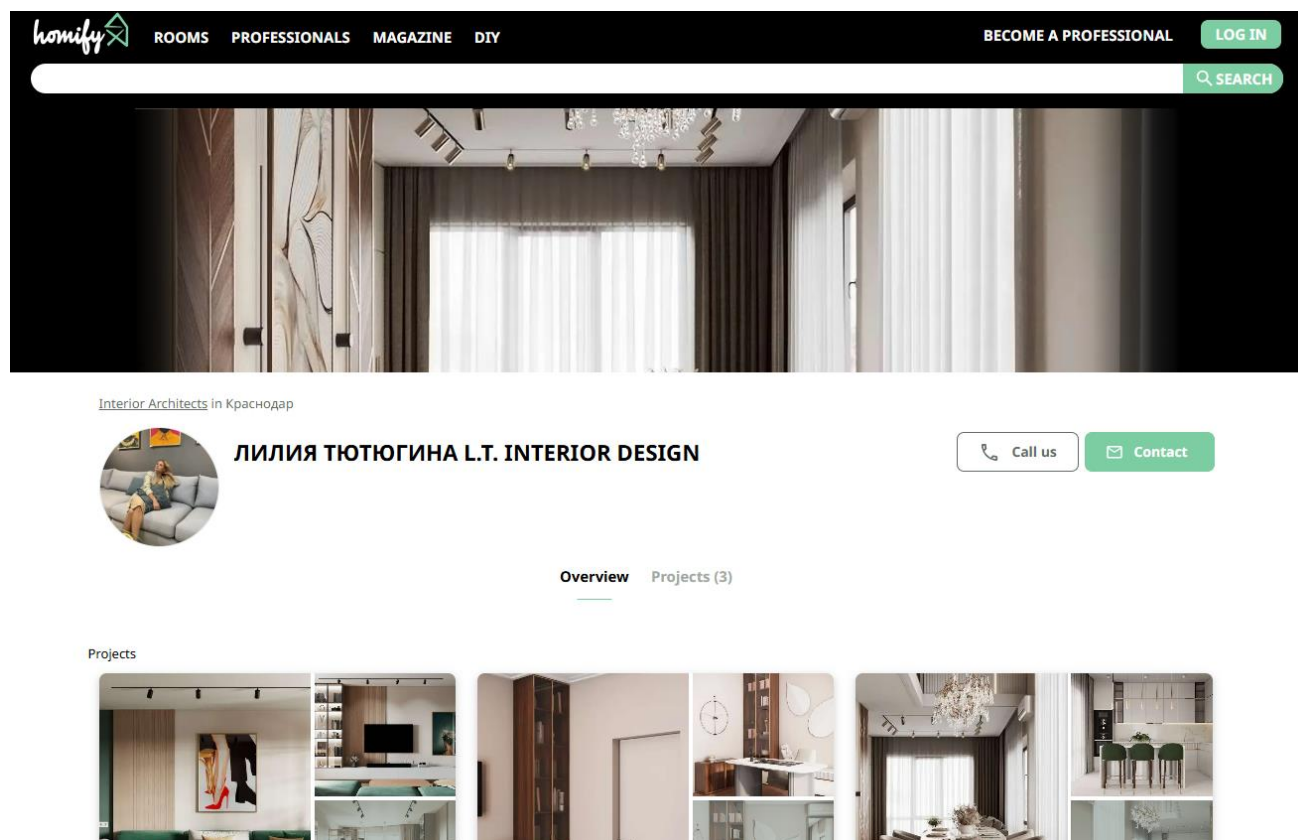
Figure 5 – Formulaire de description d'un projet sur Archidvisor

Avantages	Inconvénients
Spécialisation dans les projets architecturaux	Pas d'automatisation ou d'IA pour le matching
Architectes validés manuellement (garantie de qualité)	Pas d'espace collaboratif ou tableau de bord projet
Interface épurée	Suivi du projet souvent externalisé (emails, appels)

- **Tableau 3 : Synthèse des fonctionnalités d'Archidvisor**

4.1.3 Homify

Annuaire en ligne de professionnels (architectes, décorateurs, artisans) avec une interface centrée sur l'inspiration visuelle. Chaque professionnel y expose ses réalisations.



• Figure 7 – Fiche professionnelle d'un architecte sur Homify

Avantages	Inconvénients
Visibilité importante pour les professionnels	Aucune interaction structurée entre client et architecte
Présentation visuelle de projets réussis	Pas de devis en ligne ou système de messagerie intégré
Simple d'utilisation	Aucun accompagnement ou suivi projet intégré

• Tableau 4 : Synthèse des fonctionnalités de Homify

4.1.4 Plateformes Freelance (Upwork, Fiverr)

Marchés de services freelances où certains architectes proposent des prestations d'architecture ou de design d'intérieur. Très généralistes et peu spécialisés.

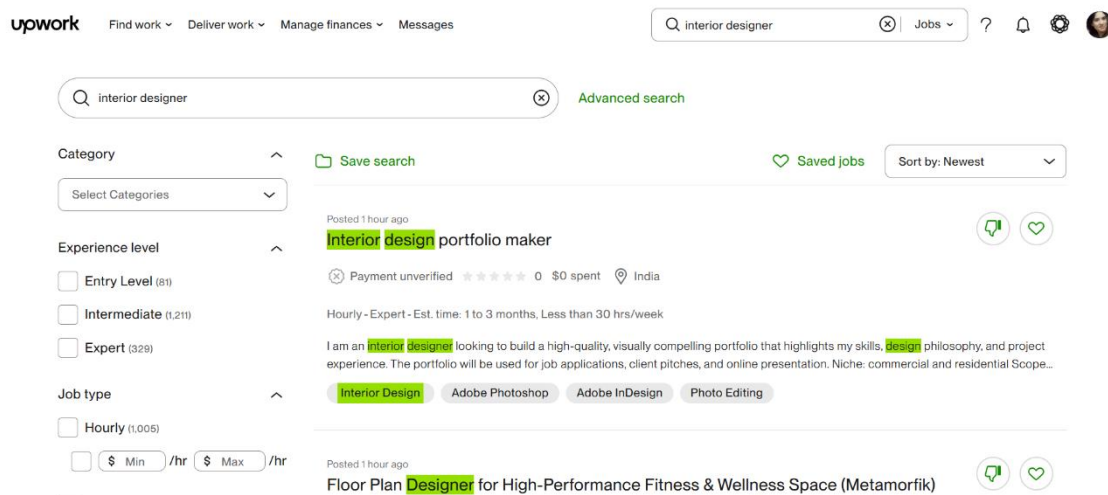


Figure 7– Résultat de recherche d'un architecte sur Upwork

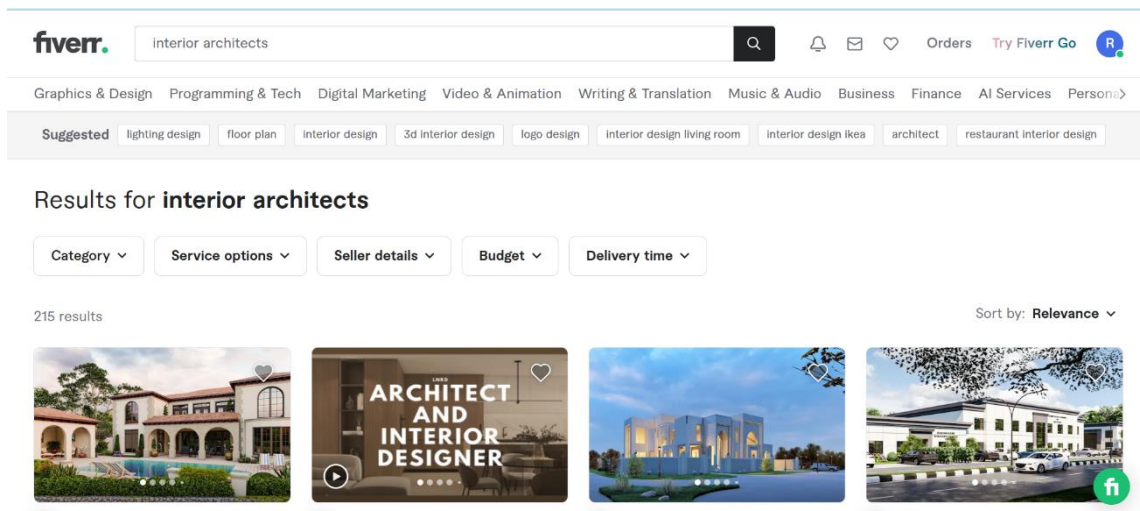


Figure 8 – Exemple de profil freelance en architecture sur Fiverr

Avantages	Inconvénients
Grande diversité de profils et de prix	Plateformes non spécialisées dans l'architecture d'intérieur
Système d'avis et de notation fiable	Pas de matching précis ni gestion collaborative
Processus de commande simplifié	Pas de filtrage métier ni validation des profils par spécialité

• Tableau 5 : Synthèse des plateformes freelance analysées

4.2 Critique de l'existant

L'analyse des solutions existantes telles que Houzz, Archidvisor, Homify ou la plateforme freelance montre qu'elles offrent une certaine visibilité aux professionnels et intègrent parfois des fonctionnalités utiles comme les portfolios ou les marketplaces. Toutefois, elles restent limitées par l'absence d'un système de matching intelligent basé sur les besoins spécifiques des clients, une gestion de projet souvent inexistante ou externe, un manque de transparence sur la qualité et les tarifs des prestations, ainsi qu'un déficit de collaboration en temps réel. De plus, la validation des profils professionnels est souvent superficielle. Ces limites justifient la nécessité de développer une solution plus complète, comme Visiona, qui répond à ces enjeux de manière centralisée et intelligente.

5. Objectifs du projet

L'objectif principal de ce projet est de concevoir et développer une plateforme web SaaS baptisée *Visiona*, dédiée à la mise en relation entre clients et architectes d'intérieur. Cette plateforme vise à répondre aux besoins réels du marché en proposant un environnement digital intelligent, sécurisé et collaboratif. Elle permettra aux clients de publier leurs besoins, de recevoir des devis personnalisés grâce à un système de recommandation intelligent basé sur l'intelligence artificielle, et de suivre l'évolution de leurs projets via un tableau de bord interactif. Du côté des architectes, elle offrira des outils de gestion de projets, de visibilité professionnelle (portfolio), ainsi qu'un espace marketplace pour promouvoir leurs produits ou services. L'administrateur disposera d'un module de supervision pour garantir la qualité, la sécurité et la fluidité de l'écosystème. En somme, ce projet a pour ambition de digitaliser et centraliser l'ensemble du processus de collaboration client–architecte de manière fluide, efficace et innovante.

6. Méthodologie de gestion de projet

6.1. Introduction à la méthodologie Agile

6.1.1. Définition et principes fondamentaux

La méthodologie Agile est une approche de gestion de projet qui privilégie la flexibilité, la collaboration et l'adaptation continue aux besoins changeants du client. Contrairement aux méthodes traditionnelles dites "en cascade", Agile repose sur des cycles courts de développement et une interaction constante entre les membres de l'équipe et les parties prenantes. Cette approche est formalisée dans le Manifeste Agile, qui met en avant quatre valeurs principales : les individus et leurs interactions, le logiciel fonctionnel, la collaboration avec le client et la réponse au changement.

6.1.2. Avantages de l'approche Agile

L'approche Agile permet de réduire les risques liés aux projets complexes en favorisant une livraison progressive et régulière de fonctionnalités. Elle améliore la satisfaction client grâce à une meilleure prise en compte des retours et facilite la communication au sein des équipes. De plus, Agile encourage l'amélioration continue, ce qui permet d'optimiser les processus au fil du projet.

6.2. Présentation des principaux frameworks Agile

6.2.1. Scrum : cadre de travail itératif

Scrum est un cadre Agile structuré qui organise le travail en sprints, cycles courts généralement de 2 à 4 semaines, au terme desquels une version fonctionnelle du produit est livrée. Il définit des rôles spécifiques (Product Owner, Scrum Master, équipe de développement) et des événements réguliers

(réunions quotidiennes, revues de sprint, rétrospectives) pour assurer la transparence, l'inspection et l'adaptation.

6.2.2. Kanban : gestion visuelle du flux

Kanban est une méthode de gestion du travail basée sur la visualisation des tâches à travers un tableau (physique ou numérique). Elle permet de limiter le nombre de tâches en cours pour fluidifier le flux de travail et identifier rapidement les blocages. Kanban est particulièrement adaptée aux environnements où les priorités changent fréquemment.

6.2.3. Autres frameworks (Lean, XP, Crystal, SAgile, etc.)

- Lean : vise à éliminer les gaspillages et optimiser la valeur délivrée.
- Extreme Programming (XP) : met l'accent sur la qualité technique et les bonnes pratiques de développement.
- Crystal : famille de méthodes adaptées selon la taille et la criticité du projet.
- SAgile (Scaled Agile Framework) : permet de déployer Agile à grande échelle dans les grandes organisations.

6.2.4. Comparaison synthétique des frameworks

Framework	Points forts	Utilisation typique
Scrum	Structure claire, sprints, rôles définis	Projets complexes nécessitant flexibilité et collaboration
Kanban	Visualisation, flux continu, flexibilité	Environnements à flux de travail variable
SAgile	Coordination multi-équipes, échelle	Grandes organisations avec plusieurs équipes Agile

Tableau 6 : Comparaison des frameworks Agile

6.3. Focus sur Scrum

6.3.1. Description détaillée de Scrum

Scrum est conçu pour gérer des projets complexes en favorisant un développement itératif et incrémental. Le travail est organisé en sprints, chacun aboutissant à un produit potentiellement livrable. Scrum s'appuie sur trois piliers : la transparence, l'inspection et l'adaptation, garantissant ainsi une amélioration continue.

6.3.2. Rôles et responsabilités dans Scrum

Product Owner : Responsable de la priorisation du backlog et de la vision du projet.

Scrum Master : Garant de l'application de la méthodologie Scrum et facilitateur de l'équipe.

Development Team : Équipe de développement qui conçoit, développe et teste les fonctionnalités.

6.3.3. Événements et artefacts Scrum

- Sprint Planning : réunion de planification du sprint.
- Daily Scrum : réunion quotidienne de synchronisation.
- Sprint Review : présentation des résultats du sprint aux parties prenantes.
- Sprint Retrospective : réunion d'amélioration continue.
- Backlog produit : liste ordonnée des besoins.
- Backlog sprint : sélection des tâches pour le sprint en cours.

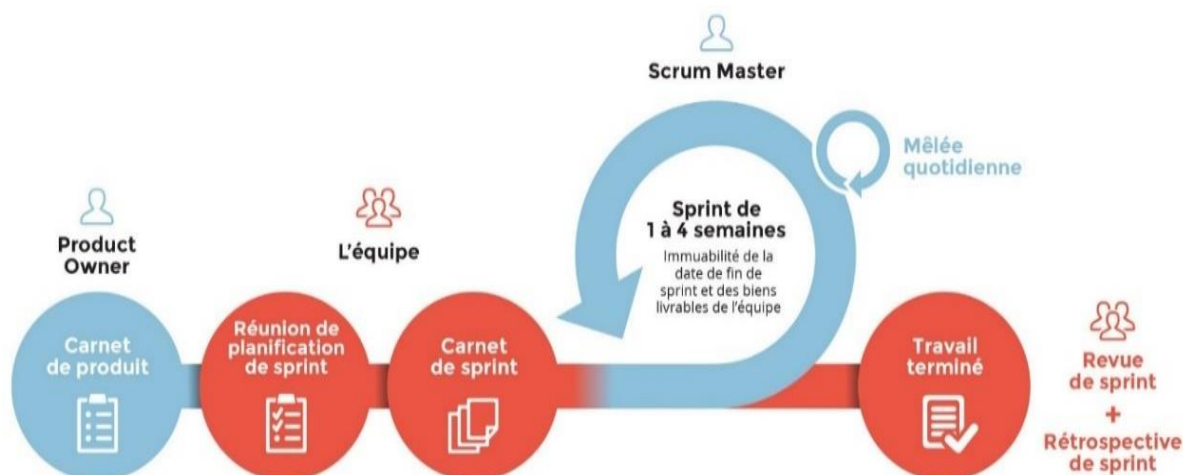


Figure 1 : Cycle de vie Scrum

6.4. Justification du choix de Scrum

6.4.1. Adaptabilité et flexibilité

Scrum permet de s'adapter rapidement aux changements grâce à ses cycles courts et ses points de contrôle réguliers. Cela est particulièrement utile dans un contexte où les besoins évoluent fréquemment.

6.4.2. Collaboration et communication renforcées

La méthode favorise la communication quotidienne et la collaboration entre les membres de l'équipe ainsi qu'avec le client, ce qui améliore la qualité du produit final.

6.4.3. Livraison rapide et amélioration continue

Les sprints garantissent des livraisons régulières, permettant d'obtenir rapidement des retours et d'ajuster le produit. Les rétrospectives encouragent une amélioration constante des processus.

6.4.4. Disponibilité des ressources et adoption dans l'industrie

Scrum est largement adopté dans le monde professionnel, ce qui facilite l'accès à des ressources, des outils et des formations. Cette popularité assure également une meilleure intégration avec d'autres pratiques et équipes.

6.5. Conclusion sur la méthodologie adoptée

En résumé, la méthodologie Agile, et plus précisément le framework Scrum, offre un cadre structuré et flexible adapté aux projets complexes et évolutifs. Son approche itérative, centrée sur la collaboration et l'adaptation, permet de maximiser la valeur délivrée tout en minimisant les risques. C'est pourquoi Scrum a été retenu comme méthodologie principale pour la gestion de ce projet.

7. Conclusion

Ce premier chapitre a permis de présenter le contexte général du projet, l'entreprise d'accueil et l'analyse de l'existant. Il a mis en évidence les besoins non couverts par les solutions actuelles et a justifié la pertinence du projet Visiona. Cette base solide permet d'aborder, dans le chapitre suivant, l'analyse détaillée des besoins fonctionnels et techniques, étape essentielle pour garantir la réussite du développement de la plateforme.

Chapitre 2 : Analyse et spécification des besoins

1. Introduction

Avant de passer à la phase de développement, la première étape de notre projet consiste à comprendre précisément les attentes des utilisateurs et les contraintes du système. Cette phase est cruciale, car elle permet à l'analyse des besoins, tant fonctionnels que non fonctionnels, en identifiant les acteurs du système, en définissant les fonctionnalités attendues, et en choisissant les outils techniques et organisationnels les plus adaptés. Elle nous aide également à établir le backlog initial (la liste des tâches) et à planifier les sprints (périodes de travail). Cette étape est essentielle pour établir une base solide sur laquelle reposera toute la conception de la plateforme.

2. Identification des besoins

2.1. Besoins Fonctionnels

2.1.1. Pour le Client

- Créer un compte, s'authentifier et gérer son profil : Accéder à un espace personnel sécurisé pour la gestion de ses informations.
- Déposer une fiche projet : Exprimer ses besoins et attentes de manière détaillée via un formulaire intuitif.
- Consulter les profils des architectes : Accéder à des fiches complètes présentant les compétences, réalisations et avis des professionnels.
- Recevoir et comparer plusieurs devis : Bénéficier d'un système transparent permettant de comparer les offres et de choisir la plus adaptée à son projet.
- Acheter des produits ou services via la marketplace : Accéder à une sélection de solutions complémentaires pour enrichir son projet.
- Laisser des avis et notes : Partager son retour d'expérience afin de contribuer à la qualité du réseau.
- Recevoir des notifications en temps réel : Être informé instantanément des propositions, messages ou changements relatifs à ses projets.

2.1.2. Pour l'Architecte

- S'inscrire et compléter son profil professionnel : Présenter ses compétences, expériences, réalisations et spécialités à travers un portfolio détaillé.
- Validation par un administrateur : Garantir la qualité et la fiabilité des profils via un processus de validation préalable.
- Accéder aux projets compatibles : Visualiser et filtrer les demandes clients correspondant à son expertise.
- Envoyer des devis personnalisés : Proposer des offres sur mesure en réponse aux besoins identifiés.
- Gérer ses projets : Suivre l'évolution des projets acceptés, planifier les tâches, gérer le calendrier et déposer les livrables.
- Mettre à jour son tableau de bord : Suivre ses activités (projets, produits, commandes) et gérer son offre sur la marketplace.
- Ajouter des produits à la marketplace : Valoriser son expertise à travers la mise en ligne de produits ou services complémentaires.
- Consulter ses statistiques personnelles : Analyser ses performances (nombre de vues, commandes, avis reçus) pour optimiser sa présence.
- Recevoir des notifications en temps réel : Être informé immédiatement des nouvelles opportunités ou des interactions clients.

2.1.3. Pour l'Administrateur

- Gérer les inscriptions des architectes : Valider ou rejeter les demandes d'inscription afin de garantir la qualité du réseau.
- Gérer les comptes utilisateurs : Suspendre ou supprimer des comptes en cas de non-respect des conditions d'utilisation.
- Superviser les projets en cours : Suivre l'avancement des projets pour anticiper d'éventuels dysfonctionnements.
- Gérer le contenu de la marketplace : Valider ou modérer les produits proposés et traiter les signalements.
- Gérer les paiements : Superviser les transactions financières pour assurer la transparence et la sécurité des échanges.
- Modérer les avis : Contrôler les retours utilisateurs afin de préserver la fiabilité de la plateforme.
- Accéder aux statistiques d'usage : Analyser les données d'utilisation pour piloter l'évolution de la plateforme et prendre des décisions stratégiques.

2.2 Besoins Non Fonctionnels

Les exigences non fonctionnelles de la plateforme sont essentielles pour garantir la qualité, la pérennité et l'adoption du service par ses utilisateurs. Elles couvrent plusieurs axes clés qui vont au-delà des simples fonctionnalités, afin d'assurer une expérience fiable, sécurisée et évolutive.

Les besoins non fonctionnels de mon projet comprennent :

- Portabilité : L'application web doit être compatible avec une large gamme de navigateurs.
- Sécurité : Authentification JWT, validation des profils.
- Accessibilité : Application responsive.
- Scalabilité : Supporter l'évolution du trafic et des utilisateurs.
- Ergonomie : Il est essentiel que l'application fournisse une interface conviviale, intuitive et moderne, garantissant une utilisation facile et offrant une expérience utilisateur agréable.
- Performance optimale : L'application doit être optimisée pour garantir des temps de réponse rapides et des temps de chargement minimaux

3. Planification du projet avec Scrum

Dans la continuité de la méthodologie Scrum présentée au chapitre précédent, cette section détaille la planification opérationnelle du projet, en mettant l'accent sur la gestion du backlog produit et l'organisation des sprints.

3.1. Equipe et rôles

Dans cette phase, je vais présenter les membres clés de l'équipe Scrum :

Le Product Owner : Monsieur **Amir Bouker** est désigné comme le représentant des clients. Sa responsabilité principale est de définir les priorités et les objectifs du produit.

Le Scrum Master : Madame **Mouna Karaja** prend en charge le rôle de Scrum Master. Elle nous guidera dans l'application dans le cadre méthodologique Scrum, et nous aidera à surmonter les obstacles rencontrés lors des différentes réunions.

Développeur : **Roua Mejdoub** en charge du développement itératif, j'ajoute de nouvelles fonctionnalités à chaque sprint.

3.2. Product backlog

Le backlog produit constitue la liste priorisée des fonctionnalités et tâches à réaliser, exprimées sous forme d'user stories. Chaque user story décrit un besoin fonctionnel ou technique du point de vue de l'utilisateur final, accompagnée d'une estimation de complexité (story points) et d'une priorité.

ID	Titre	User Story	Priorité	Complexité
US01	Authentification multi-rôle	En tant qu'utilisateur (client, architecte, admin), je veux m'authentifier pour accéder à mon espace sécurisé.	Haute	Moyenne
US02	Validation inscription architecte	En tant qu'administrateur, je peux valider ou rejeter l'inscription d'un architecte avant activation de son compte.	Haute	Faible
US03	Gestion des rôles	En tant qu'administrateur, je veux attribuer et gérer les rôles des utilisateurs.	Haute	Moyenne
US04	Création de compte client	En tant que client, je veux pouvoir créer un compte pour accéder aux services de la plateforme.	Haute	Moyenne
US05	Création de compte architecte	En tant qu'architecte, je veux pouvoir créer un compte et soumettre mon profil pour validation.	Haute	Moyenne
US06	Consultation profils architectes	En tant que client, je veux consulter les profils détaillés des architectes.	Haute	Moyenne
US07	Dépôt de fiche projet	En tant que client, je veux déposer une fiche projet pour décrire mes besoins.	Haute	Moyenne
US08a	Envoi de devis par architectes	En tant qu'architecte, je veux pouvoir envoyer des devis personnalisés aux clients.	Haute	Moyenne
US08b	Consultation et comparaison des devis	En tant que client, je veux consulter et comparer plusieurs devis reçus pour mon projet.	Haute	Moyenne
US09	Gestion des abonnements architectes	En tant qu'architecte, je souhaite choisir un abonnement (Gratuit, Premium, VIP) avec fonctionnalités spécifiques selon le plan.	Haute	Moyenne
US10	Gestion des projets - création	En tant qu'architecte, je veux créer et organiser mes projets dans la plateforme.	Haute	Moyenne
US11	Gestion des projets - suivi via Kanban	En tant qu'architecte, je veux suivre l'avancement de mes projets à travers un tableau Kanban.	Moyenne	Moyenne
US12	Gestion des projets - collaboration	En tant qu'architecte et client, je veux échanger et collaborer via la plateforme sur un projet.	Moyenne	Moyenne
US13	Marketplace produits	En tant qu'architecte, je veux vendre des produits via la marketplace intégrée.	Moyenne	Moyenne

US14	Intégration Stripe	En tant qu’architecte, je veux pouvoir souscrire à un abonnement et payer via Stripe.	Haute	Moyenne
US15	Notifications en temps réel	En tant qu’utilisateur, je veux recevoir des notifications instantanées sur les interactions importantes.	Moyenne	Moyenne
US16	Gestion des avis et notes	En tant que client, je veux laisser des avis et notes sur les architectes et produits.	Moyenne	Faible
US17	Modération des contenus	En tant qu’administrateur, je veux modérer les avis, produits et profils pour garantir la qualité.	Haute	Moyenne
US18	Moteur de matching IA	En tant que client, je veux que la plateforme me recommande des architectes adaptés à mes besoins via un moteur IA.	Moyenne	Élevée

Tableau 7 : Product backlog général

Ce backlog est régulièrement révisé et affiné avec le Product Owner pour intégrer les retours et ajuster les priorités en fonction de la valeur métier.

3.3. Planification des sprints

Le projet est découpé en cycles de développement courts, appelés **sprints**, d’une durée de 2 à 3 semaines. Chaque sprint vise à livrer un incrément fonctionnel prêt à être testé et validé.

La planification des sprints s’appuie sur :

- La sélection des user stories les plus prioritaires et réalisables dans le temps imparti.
- La répartition des tâches au sein de l’équipe de développement.
- La définition d’objectifs clairs pour chaque sprint, favorisant la livraison continue de valeur.

Sprint	Objectifs du sprint	User Stories / Tâches incluses	Durée	Livrables clés
Sprint 1	Authentification & inscription des utilisateurs	US01, US02, US03, US04, US05	2 semaines	Système d’authentification multi-rôle, inscription validée par admin
Sprint 2	Gestion des projets et devis entre clients et architectes	US07, US08, US09a, US09b	3 semaines	Interface projet + gestion devis fonctionnelle
Sprint 3	Abonnements architectes & outils de gestion	US10, US11, US12, US13	3 semaines	Système d’abonnement opérationnel, Tableau de bord architecte
Sprint 4	Fonctionnalités avancées & intelligence artificielle	US14, US15, US16, US17, US18, US19	2 semaines	Marketplace, notifications, avis, modération, moteur IA

Tableau 8 : planification des sprints

4. Architecture générale de l'application

4.1 Architecture logique : Modèle MVC (Modèle-Vue-Contrôleur)

L'application suit une architecture MVC (Modèle-Vue-Contrôleur), un modèle largement adopté pour séparer les responsabilités et faciliter la maintenance [1]

Modèle (Modèle de données)

- **Rôle** : Gère les données et la logique métier.
- **Implémentation** :
 - Schémas MongoDB (utilisateurs, projets, devis, produits) avec validation stricte des données.
 - Couche d'accès aux données (DAO) pour les opérations CRUD.

Vue (Couche de présentation)

- **Rôle** : Affiche les données et capture les interactions utilisateur.
- **Implémentation** :
 - Frontend React.js/Next.js avec composants modulaires.
 - Tableau de bord
 - Interactif (charts, formulaires dynamiques).
 - Communication via API REST (Axios) et WebSocket (Socket.io).
 - Optimisation SEO grâce au rendu côté serveur (SSR) de Next.js.

Contrôleur (Logique applicative)

- **Rôle** : Orchestre les interactions entre le modèle et la vue.
- **Implémentation** :
 - Contrôleurs Express.js pour gérer les routes (authentification, projets, devis).
 - Middlewares de validation et d'authentification (JWT).

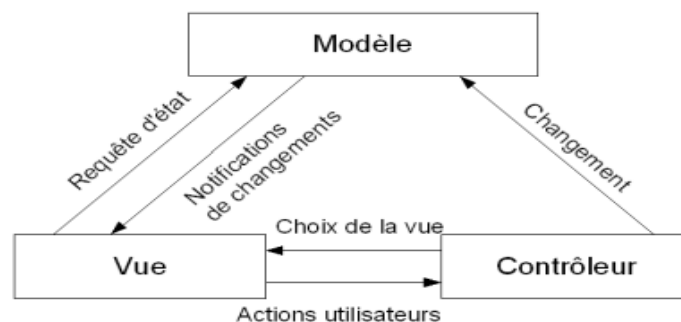


Figure 10 : Architecture MVC

4.2. Architecture physique (Déploiement et infrastructure)

L'architecture physique suit une approche cloud-native pour la scalabilité et la résilience

Composants clés

1. Couche Client :

- Navigateurs web (Chrome, Firefox, Safari).
- Applications mobiles (via API responsive).

2. Couche Serveur :

- Frontend : Déployé sur Vercel, avec cache CDN et pré-rendu statique.
- Backend :
 - Cluster AWS EC2 auto-scalable (2-8 instances).
 - Gestion des sessions via Redis pour les WebSockets.
 - Conteneurisation Docker pour l'isolation des services.

3. **Base de données :**

- MongoDB Atlas (M0 Cluster) avec réplication multi-région.
- Indexation avancée pour les requêtes fréquentes (ex : recherche d'architectes).

4.3. Justification des choix techniques

4.3.1. Pourquoi MVC ?

-Une meilleure organisation du code, facilitant la maintenance et l'évolution.

-Une indépendance des équipes frontend et backend, qui peuvent travailler sur la vue et le modèle respectivement sans conflit.

-Une testabilité accrue, car chaque composant peut être testé isolément, notamment avec des frameworks Jest pour les tests unitaires.

4.3.2. Pourquoi MongoDB ?

Flexibilité : Schémas dynamiques adaptés aux besoins changeants des clients.

Performance : Requêtes agrégées optimisées pour les Tableaux de bord en temps réel.

5.Méthode et langage de Modélisation utilisé

5.1. Définition d'UML et son utilité



Figure 11 : Logo d'UML

Le **langage de modélisation unifié (UML, Unified Modeling Language)** est un standard international défini par l'OMG (Object Management Group) qui permet de spécifier, visualiser, construire et documenter les systèmes logiciels et les processus métiers. UML offre un langage graphique commun, facilitant la compréhension et la communication entre les différents acteurs d'un projet, qu'ils soient développeurs, architectes ou clients.[2]

5.1.1. Pourquoi UML ?

L'objectif principal d'UML est de fournir un cadre formel et standardisé pour représenter la structure, le comportement et l'architecture d'un système complexe, en particulier dans le contexte du développement orienté objet. Il permet ainsi de décrire clairement les exigences, la conception et les interactions au sein du système, ce qui favorise la qualité, la maintenabilité et la collaboration tout au long du cycle de vie du projet.

5.1.2 Types de diagrammes UML

Il existe deux sous-catégories de diagrammes UML : les diagrammes structurels et les diagrammes comportementaux.

Les **diagrammes structurels** visualisent les éléments qui composent un système et les relations entre eux, montrant ainsi les aspects statiques d'un système.

Les **diagrammes comportementaux**, quant à eux, représentent ce qui se passe à l'intérieur d'un système, y compris la manière dont les composants interagissent entre eux et avec d'autres systèmes ou utilisateurs.[3]

5.2. Les diagrammes UML utilisés dans le projet

5.2.1. Diagramme de cas d'utilisation

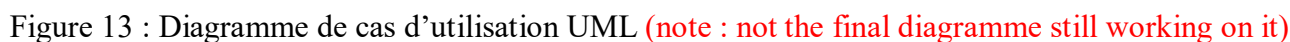
En langage UML, les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système.

Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne. [4]



Figure 12 : Diagramme de classes UML global du projet

Dans le langage UML, les *diagrammes de classes* appartiennent à l'un des six types de diagramme structurel. Les diagrammes de classes sont fondamentaux pour le processus de modélisation des objets et modélisent la structure statique d'un système [5]. Le diagramme de classes représente la structure statique du système en détaillant les principales classes métier, leurs attributs, méthodes et les relations qui les unissent (associations, héritages, agrégations). Ce diagramme est essentiel pour la conception de la base de données et la mise en œuvre de la logique applicative.



Ce chapitre a permis de poser les bases solides du projet en identifiant précisément les besoins des différents acteurs, en détaillant les exigences fonctionnelles et non fonctionnelles, et en définissant les choix techniques et méthodologiques adaptés. Grâce à cette analyse approfondie, nous avons pu structurer le backlog, planifier les sprints selon la méthode Scrum, et concevoir une architecture robuste et évolutive. La modélisation UML a facilité la formalisation des interactions et de la structure du système, garantissant ainsi une vision partagée et cohérente du projet. Cette étape d'analyse et de spécification constitue le socle indispensable à la réussite du développement, en minimisant les risques et en assurant une meilleure adaptation aux attentes des utilisateurs. La suite du rapport détaillera la conception technique et la mise en œuvre concrète de la plateforme.

Chapitre 3 : Etude Et Réalisation Du Sprint 1

1.Introduction

Le Sprint 1 constitue la première étape cruciale dans le développement de la plateforme. Son objectif principal est de mettre en place un système d'authentification sécurisé multi-rôle, permettant aux utilisateurs (clients, architectes, administrateurs) de créer et gérer leurs comptes. Cette phase inclut également la validation manuelle des inscriptions des architectes par un administrateur, garantissant la qualité et la fiabilité des profils présents sur la plateforme.

Ce sprint pose les bases techniques et fonctionnelles indispensables pour la suite du projet, en assurant la sécurité, la gestion des accès et la structuration des rôles utilisateurs.

2.Backlog de Sprint 1

Le backlog du Sprint 1 regroupe les user stories essentielles suivantes :

ID	Titre	User Story	Priorité	Complexité
US01	Authentification multi-rôle	En tant qu'utilisateur (client, architecte, admin), je veux m'authentifier pour accéder à mon espace sécurisé.	Haute	Moyenne
US02	Validation inscription architecte	En tant qu'administrateur, je peux valider ou rejeter l'inscription d'un architecte.	Haute	Faible
US03	Gestion des rôles	En tant qu'administrateur, je veux attribuer et gérer les rôles des utilisateurs.	Haute	Moyenne
US04	Création de compte client	En tant que client, je veux créer un compte pour accéder aux services de la plateforme.	Haute	Moyenne
US05	Création de compte architecte	En tant qu'architecte, je veux créer un compte et soumettre mon profil pour validation.	Haute	Moyenne

Tableau 9 : backlog du Sprint 1

3. Spécification fonctionnelle :

3.1 Diagramme de cas d'utilisation sprint 1

Le diagramme présente les interactions entre les acteurs (client, architecte, administrateur) et les fonctionnalités clés : création de compte, authentification, validation des architectes, gestion des rôles.

3.2. Analyse de cas d'utilisation « Créer compte »

3.2.1. Diagramme cas d'utilisation « Créer compte »

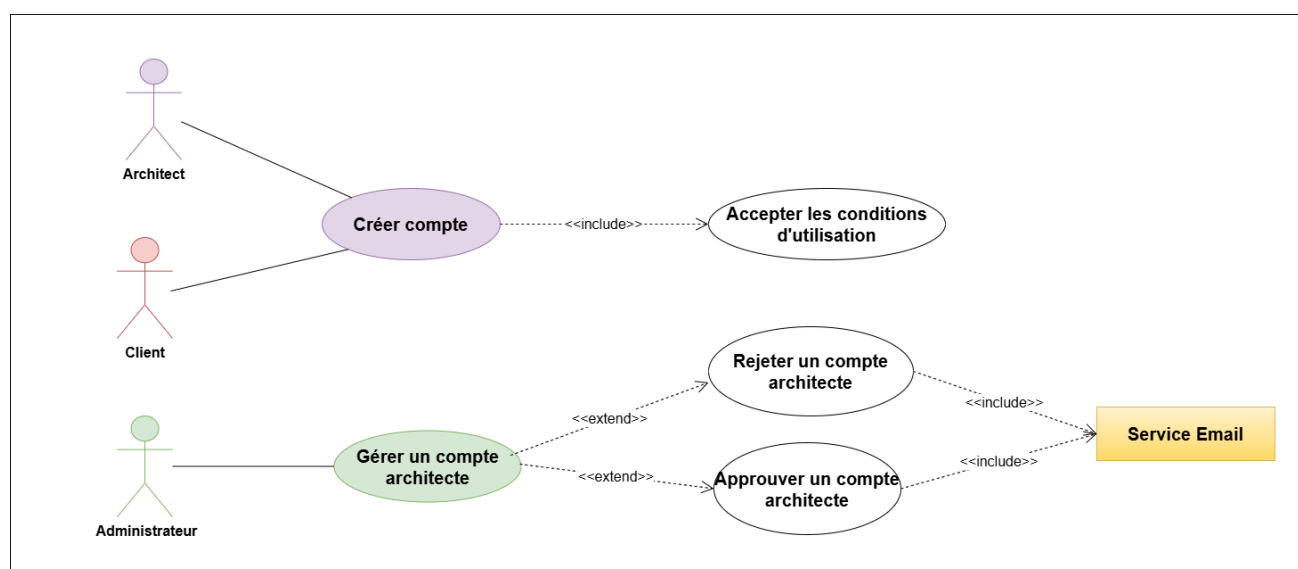


Figure 14 : Diagramme de classes « Créer compte »

3.2.2. Description textuelle de cas d'utilisation « Créer compte »

Titre:	Créer compte.
Description brève du CU:	L'acteur doit remplir les champs de formulaire pour créer son propre compte.
Acteur:	Client, Architecte
Préconditions:	L'acteur n'est pas encore inscrit.
Post-condition:	Un nouveau compte est créé dans la base (client ou architecte selon le rôle).
Scénario de base:	<ol style="list-style-type: none">1. L'acteur accède à la page d'inscription.2. L'acteur sélectionne son type de compte (client ou architecte).3. Le système affiche un formulaire adapté.4. L'acteur remplit les champs nécessaires.

	5. L'acteur accepte les conditions générales. 6. L'acteur clique sur « S'inscrire ». 7. Le système valide les champs saisis. 8. Le système vérifie si le compte existe déjà. 9. Le mot de passe est crypté. 10. Le compte est enregistré en base (status = en attente si architecte). 11. Un email de confirmation est envoyé. 12. Redirection vers la page de connexion.
--	--

Tableau 10 : Description textuelle du CU « créer compte »

Scénarios alternatifs :

6.a. L'acteur ne respecte pas les contrôles de saisie :

→ Le système affiche un message d'erreur et retourne à l'étape 4.

7.a. Le compte existe déjà dans la base :

→ Le système affiche un message d'erreur et retourne à l'étape 4.

10.a. En cas d'erreur serveur ou base :

→ Le système affiche un message d'échec de création et invite à réessayer plus tard.

3.3. Analyse de cas d'utilisation « S'authentifier » → TO DO

3.3.1. Diagramme de cas d'utilisation « S'authentifier » → TO DO

3.3.2. Description textuelle de cas d'utilisation « S'authentifier » → TO DO

3.5. Analyse de cas d'utilisation « validation des architectes » → TO DO

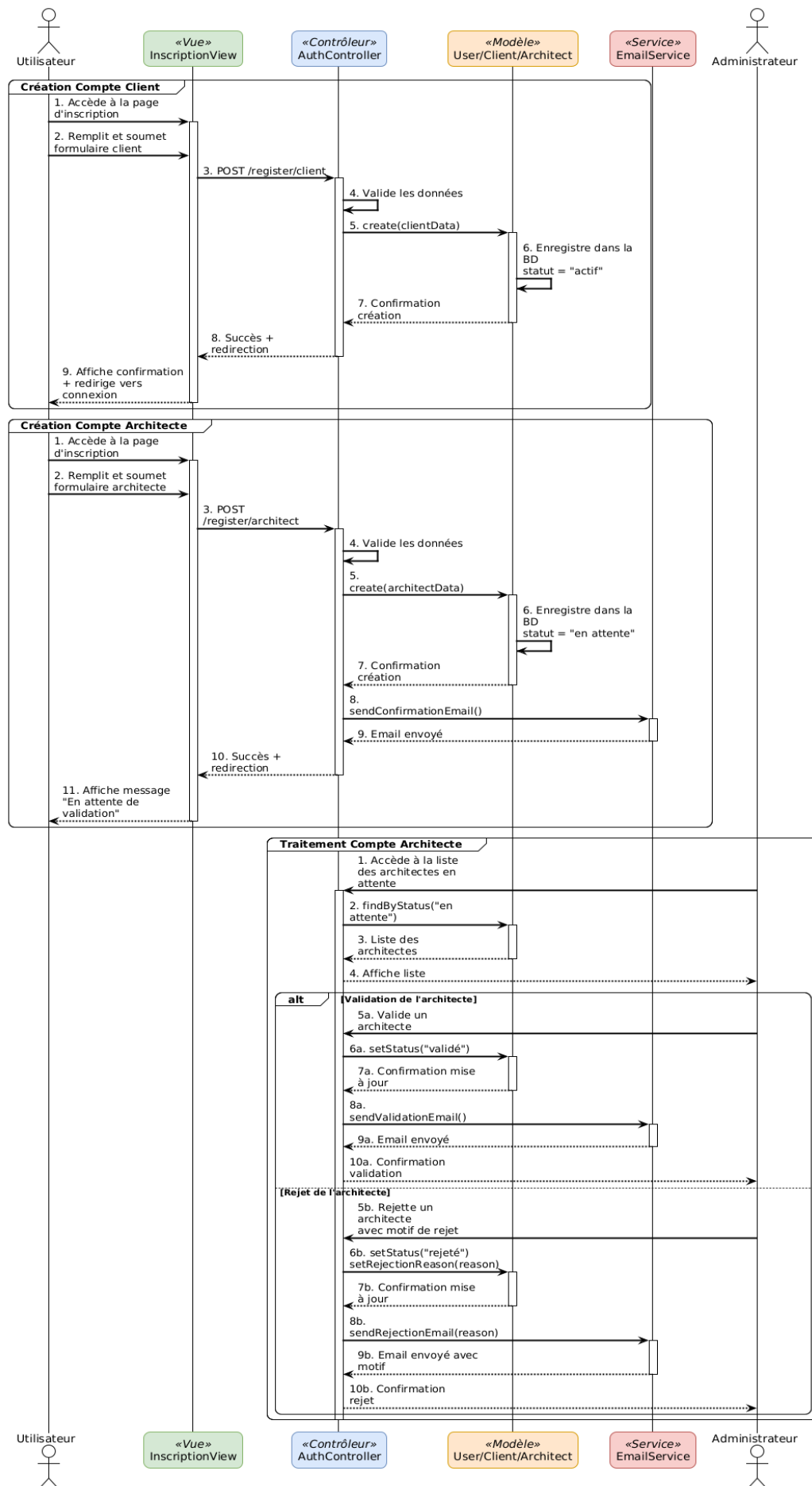
3.5.1. Diagramme de cas d'utilisation « validation des architectes » → TO DO

3.5.2. Description textuelle de cas d'utilisation « validation des architectes » → TO DO

4. Conception → TO DO

4.1. Diagramme de séquence de cas d'utilisation « Créer compte » → TO DO

Diagramme de Séquence « Créer compte »



4.2. Diagramme de séquence de cas d'utilisation « S'authentifier » → TO DO

4.6. Diagramme de classe sprint 1 → TO DO

5. Maquettes des interfaces→ TO DO

5.1. Prototypage page register→ TO DO

5.2. Prototypage page login→ TO DO

6. Réalisation : → TO DO

6.1. Interface « Créer compte » → TO DO

6.2. Interface « S'authentifier » → TO DO

7. Test

8. Outils de suivi du sprint 1

8.1. Scrum Board du sprint 1

8.2. Burn down chart du sprint 1

