

Rapport de Projet Fédéré

Mention : Sciences de l'Informatique

Spécialité : Génie Logiciel et Systèmes
d'Information

Développement d'un site web culinaire

Par :

Yassmine Ben Ghozzi

Jobran Hamda

Youssef Goddi

Prof : Dr.Wafa Tebourski, Enseignante, ISTIC

Année Universitaire : 2023-2024

Introduction générale

Grâce aux avancées des technologies de l'information et de la communication (TIC), le monde de la cuisine a connu une transformation majeure. Les sites culinaires modernes ne se limitent plus à la simple transmission de recettes, mais offrent désormais un espace dynamique où les passionnés peuvent explorer bien plus que des plats. En plus de découvrir de nouvelles créations culinaires, les utilisateurs peuvent plonger dans un univers riche en actualités, en tendances et en partage d'avis.

Ces plateformes sont devenues des espaces de vie où les chefs, amateurs et professionnels, échangent des astuces, partagent des découvertes sur les dernières tendances gastronomiques et commentent leurs expériences culinaires. Que ce soit pour discuter des innovations dans le monde de la cuisine, partager des conseils pratiques ou donner leur avis sur une nouvelle recette, les membres de ces communautés gastronomiques peuvent s'inspirer mutuellement et enrichir leur passion pour la cuisine.

Dans ce cadre s'inscrit notre projet fédéré qui vise à développer une application web pour la gestion et le suivi des recettes en ligne.

Pour la modélisation de notre projet, nous avons suivi la méthodologie agile "SCRUM". Notre rapport est organisé en trois chapitres comme suit :

- Chapitre 1 intitulé "Spécification de besoins" qui présente l'identification des besoins fonctionnels, non fonctionnels, les acteurs et les cas d'utilisation.
- Chapitre 2 intitulé "Sprint 0" Nous présentons dans ce chapitre le raffinement, la conception et la réalisation des cas d'utilisations de priorité une.
- Chapitre 3 intitulé "Sprint 1" Nous présentons dans ce chapitre le raffinement, la conception et la réalisation du sprint 1.

Enfin, nous synthétiserons notre rapport par une conclusion générale et perspectives.

Chapitre 1

SPÉCIFICATION DES BESOINS

1.1 Introduction

Comme un premier pas de notre projet, il est nécessaire d'analyser les exigences du système qu'est l'objectif de ce chapitre. Dans une première partie, nous allons étudier les besoins et présenter les objectifs de notre projet en précisant les besoins fonctionnels et les besoins non fonctionnels. Dans une deuxième partie, nous allons nous focaliser sur la description de diagramme de cas d'utilisation et la présentation de notre backlog produit. La spécification des besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les besoins de notre application. Nous distinguons des besoins fonctionnels qui présentent les fonctionnalités attendues de notre application et les besoins non fonctionnels pour éviter le développement d'une application non satisfaisante ainsi de trouver un accord commun entre les spécialistes et les utilisateurs pour réussir le projet.

1.2 Identification des besoins fonctionnels

Un besoin fonctionnel est un besoin spécifiant une action qu'un système doit être capable d'effectuer, sans considérer aucune contrainte physique. [1] C'est un besoin du point de vue de l'utilisateur.

Notre projet consiste à réaliser un site web qui répond aux besoins suivants :

- Consulter les recettes.
- Rechercher les recettes.
- S'authentifier.
- S'inscrire.
- favoriser recettes.
- Gérer actualités.

1.3 Identification des cas d'utilisation

Après avoir extrait les besoins fonctionnels, nous avons besoin de les définir dans un ordre logique :

- Consulter les recettes : Notre application doit disposer d'une table des recettes avec des options de filtrage à travers le visiteur peut consulter une grande variété des recettes.
- Rechercher les recettes : Notre application doit disposer d'une barre de recherche sur les recettes enregistrés dans la base de données à travers le visiteur peut trouver précisément les recettes dont il a besoin.
- S'inscrire à la plateforme : Jusqu'à ce stade, le visiteur est toujours anonyme mais pour passer à un stade plus rigoureux, il faut qu'il s'inscrive comme un utilisateur ou comme
- S'authentifier à la plateforme : Après la phase d'inscription, l'utilisateur a le droit de s'authentifier et selon son privilège il sera guidé comme un utilisateur sinon il a un accès en tant qu'administrateur.

1.4 Identification des besoins non fonctionnels

Les besoins non fonctionnels sont des indicateurs de qualité de l'exécution des besoins fonctionnels [1], importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur, ce qui fait qu'ils ne doivent pas être négligés.

Pour cela, il faut répondre aux exigences suivantes :

- La fiabilité : L'application doit fonctionner de façon cohérente sans erreur et doit être satisfaisante.
- Les erreurs : Les ambiguïtés doivent être signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur
- L'ergonomie et la bonne Interface : L'application doit être adaptée à l'utilisateur sans qu'il ne fournisse aucun effort (utilisation claire et facile).
- La sécurité : Notre solution doit respecter surtout la confidentialité des données personnelles des clients qui reste l'une des contraintes les plus importantes dans les applications mobile.

1.5 Identification des acteurs

Un acteur est une personne physique ou morale prenant part ou affectée par l'action ou le projet en question. Il faut donc commencer par bien préciser par rapport à quelle action ou suite d'actions on cherche à déterminer qui sont et ce que sont les acteurs.[2] Notre site web contient trois acteurs qui agissent directement avec le système :

- Le visiteur : Toute personne qui accède la plateforme dont il a la possibilité de naviguer dans page d'accueil pour consulter des recettes et les rechercher.
- L'utilisateur : Toute personne qui accède la plateforme dont il a la possibilité de créer un compte et le gérer et recevoir des notifications .
- L'administrateur : C'est le superviseur qui possède toutes les permissions de contrôle pour gérer les articles.

Acteur	Rôle
Le visiteur	<ul style="list-style-type: none"> — Consulter Articles — Consulter Recettes — Rechercher Recettes
L'utilisateur	<ul style="list-style-type: none"> — Consulter Articles — Consulter Recettes — Rechercher Recettes — S'inscrire — S'authentifier — Gérer recettes — Favoriser recettes
L'administrateur	<ul style="list-style-type: none"> — S'authentifier — Consulter Recettes — Gérer actualités

TABLE 1.1 – Description détaillée des acteurs

1.6 Diagramme de cas d'utilisation

Diagramme de cas d'utilisation a comme objectif de déterminer ce que chaque utilisateur attend du système. Notre extraction de besoins est basée sur la représentation de l'interaction entre les utilisateurs et les futures fonctions du système.

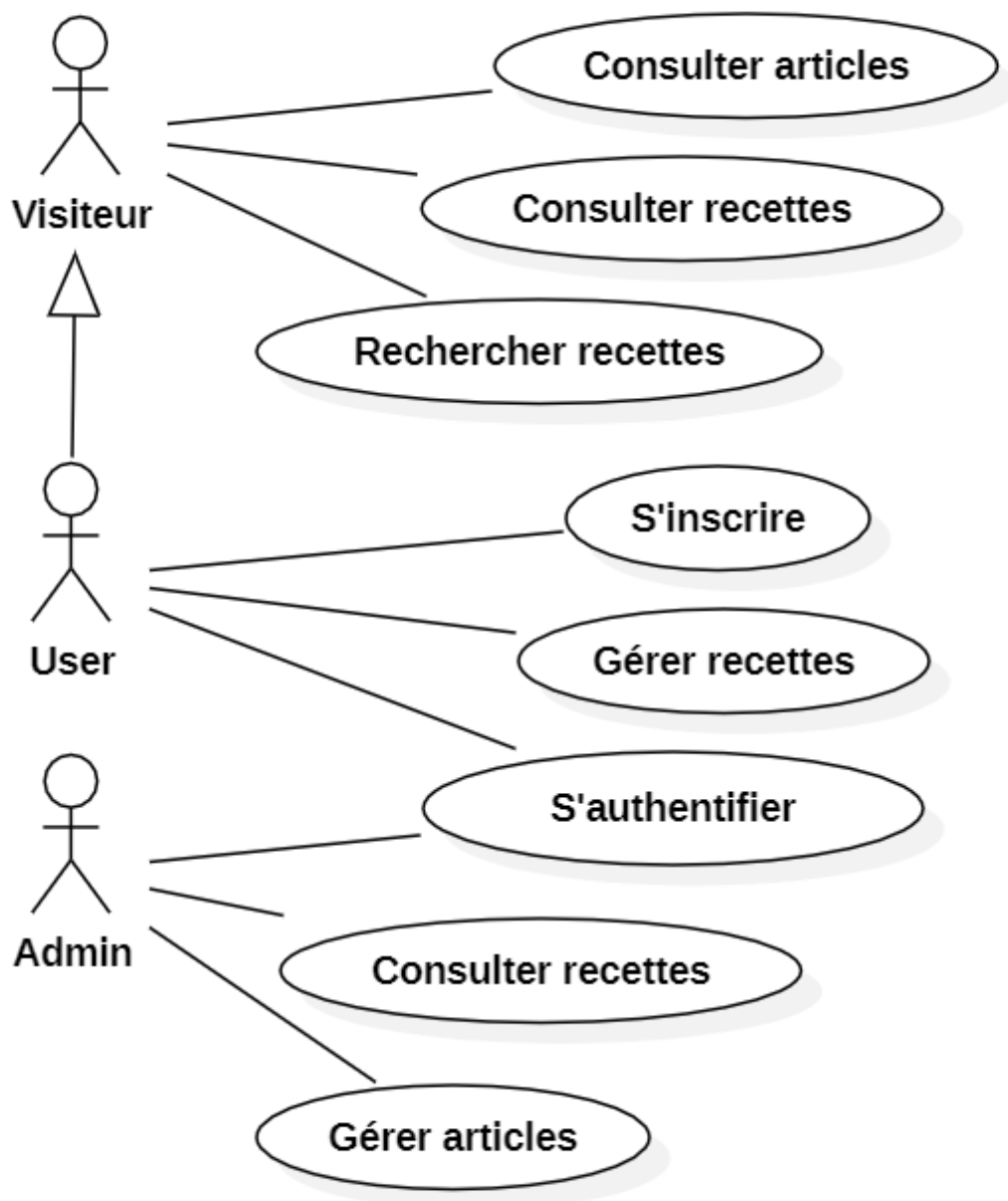


FIGURE 1.1 – Diagramme de cas d'utilisation structuré

1.7 Backlog de produit

Après avoir identifié les exigences fonctionnelles de notre système, nous présentons dans cette section le Backlog de Produit.

Backlog de produit	Priorité	Estimation	Planification
En tant qu'utilisateur, je peux m'inscrire.	1	faible	Sprint 0
En tant qu'utilisateur, je peux m'authentifier.	1	faible	Sprint 0
En tant qu'administrateur, je peux m'authentifier.	1	faible	Sprint 0
En tant qu'administrateur, je peux gérer les articles.	1	fort	Sprint 0
En tant qu'administrateur, je peux gérer les recettes.	2	fort	Sprint 1
En tant qu'utilisateur, je peux favoriser des recettes.	2	moyen	Sprint 1
En tant que visiteur, je peux consulter des recettes.	2	moyen	Sprint 1
En tant qu'utilisateur, je peux consulter des articles.	2	moyen	Sprint 1
En tant qu'administrateur, je peux gérer les recettes.	2	fort	Sprint 1

TABLE 1.2 – Backlog de produit

1.8 Environnement de travail

1.8.1 Méthodologie Agile

La méthodologie Agile s'oppose généralement aux méthodologies traditionnelles de gestion de projet. Cette méthode place les besoins du client au centre des priorités et privilégie le dialogue entre toutes les parties prenantes du projet.

À l'origine, cette approche a été créée pour les projets de développement web et informatique. Aujourd'hui, la méthode Agile est de plus en

plus répandue car elle est adaptable à de nombreux types de projets, tous secteurs confondus.[3]

1.8.2 Scrum

Scrum est la méthodologie de travail pour le développement et la maintenance de produits complexes permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière créative et productive des produits de la plus

grande valeur possible. Scrum utilise une approche incrémentale à fin optimiser la prédictibilité et pour contrôler le risque.[4]

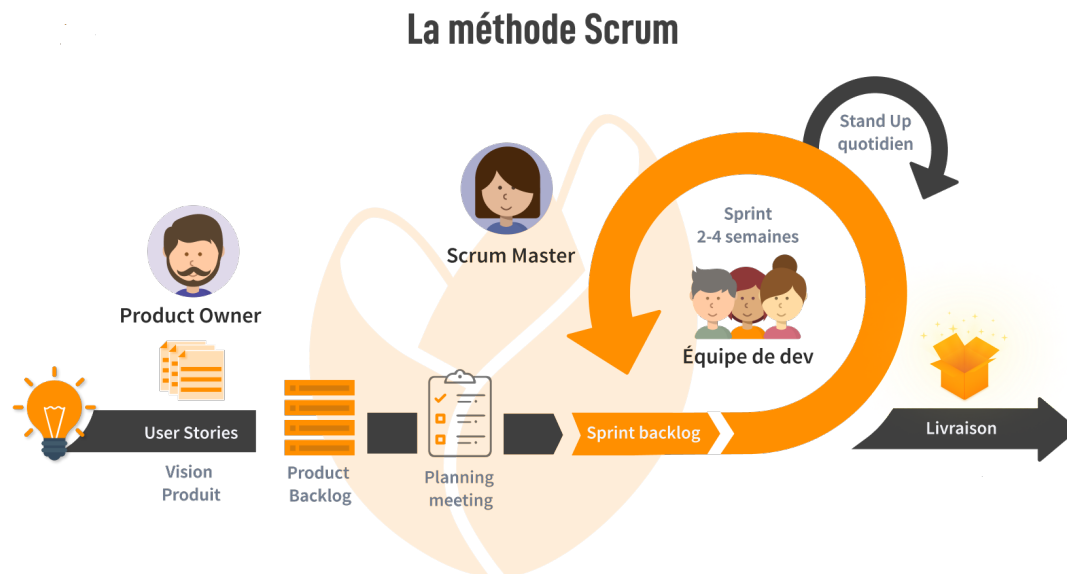


FIGURE 1.2 – Le processus de Scrum [5]

Sprint

La planification du sprint comprend l'identification des points prioritaires que l'équipe pense pouvoir réaliser au cours du sprint. La revue du sprint a lieu à la fin de chaque sprint où l'équipe de développement présente les fonctionnalités terminées. L'incrément produit est éventuellement livrable et la mise en place du prochain sprint peut être anticipée. La rétrospective du sprint permet d'améliorer le processus de développement des sprints suivants en faisant un point sur le sprint réalisé.

1.9 Outils de développement

1.9.1 Frontend

Angular

Angular est un framework JavaScript open-source développé par Google. Il est utilisé pour construire des applications web robustes et dynamiques.



FIGURE 1.3 – Logo d'Angular [6]

1.9.2 Backend

PHP

Angular est un framework JavaScript open-source développé par Google. Il est utilisé pour construire des applications web robustes et dynamiques.



FIGURE 1.4 – Logo de PHP [7]

MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR) open-source qui permet aux utilisateurs d'interagir avec les bases de données soit directement en utilisant SQL, soit plus souvent avec d'autres programmes pour mettre en œuvre des applications qui nécessitent des capacités de base de données relationnelles.



FIGURE 1.5 – Logo de MySQL [8]

1.9.3 Logiciels

Visual Studio Code

Visual Studio Code est un éditeur de code source gratuit et open-source développé par Microsoft. Il est léger, rapide et extensible, offrant une interface utilisateur intuitive et des fonctionnalités puissantes pour le développement de logiciels.

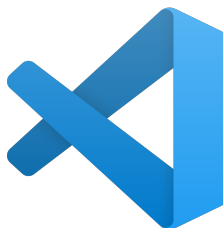


FIGURE 1.6 – Logo de Visual Studio Code [9]

Postman

Postman est une plateforme API permettant aux développeurs de créer et d'envoyer différentes requêtes HTTP et de recevoir leurs réponses. Par conséquent, il est principalement utilisé comme outil pour tester les fonctionnalités des API RESTful pendant le processus de développement.



FIGURE 1.7 – Logo de Postman [10]

StarUML

StarUML est un outil UML développé par MKLab. Il prend en charge presque tous les types de diagrammes répertoriés dans UML 2.0.



FIGURE 1.8 – Logo de StarUML[11]

XAMPP

XAMPP est un package logiciel serveur web gratuit et open-source, multiplateforme, qui contient le serveur HTTP Apache, la base de données MariaDB, ainsi que des interprètes pour les scripts PHP et Perl. Il facilite le processus de transition d'un serveur local pour les tests à un serveur en ligne.



FIGURE 1.9 – Logo de XAMPP [12]

Eclipse Java

Eclipse Java est un environnement de développement intégré (IDE) utilisé pour programmer en Java. Il propose une édition de code avancée, une gestion de projet robuste, des outils de débogage efficaces et une large gamme d'extensions et de plugins. C'est un outil polyvalent et extensible qui facilite le développement d'applications Java de toutes tailles et complexités.



FIGURE 1.10 – Logo d'Eclipse Java [13]

JEE

JEE (Java Platform, Enterprise Edition) est une plateforme de développement logiciel pour les applications d'entreprise en Java. Elle fournit des spécifications et des composants pour développer des applications distribuées et sécurisées.



FIGURE 1.11 – Logo de JEE [14]

1.9.4 Les outils de collaboration

GitHub

Git est un système de contrôle de version distribué, hébergé par GitHub, une plateforme d'hébergement de code. Il s'agit d'un logiciel développé pour suivre les modifications apportées aux fichiers au fil du temps, ce qui permet aux utilisateurs d'avoir toujours la dernière version de chaque fichier dans un référentiel, de comparer ces modifications et de les revenir à un état spécifique si nécessaire ultérieurement.

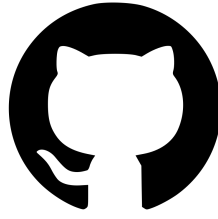


FIGURE 1.12 – Logo de GitHub [15]

1.10 Conclusion

Dans ce chapitre, nous avons présenté le diagramme de cas d'utilisation qui nous a permis le découpage fonctionnel de notre système. Dans le chapitre suivant, nous élaborons le premier sprint tout en exposant la conception et la réalisation.

Netographie

- [1] : besoin non fonctionnel
<https://savoir.plus/besoins-fonctionnels-non-fonctionnels/>
- [2] : Identification des acteurs
<http://www.nzdl.org/cgi-bin/library?e=d-00000-00—off-0unesec>
- [3] : Scrum
<https://www.planzone.fr/blog/quest-ce-que-la-methodologie-scrum>
- [4] : Le processus de Scrum
<https://www.tuleap.org/fr/agile/comprendre-methode-agile-scrum-10-minutes>
- [5] : Scrum
<https://www.planzone.fr/blog/quest-ce-que-la-methodologie-scrum>