

République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique
Direction Générale des Etudes Technologiques
ISET de Rades
Département Technologies de l'Informatique

bvDbÆEedvHc Jb' 'dF*d†Fc6Jb'
d†2bv1b' 'dF*c–Dd!E bvDc–'dFJ d†'dF(bÖ+ bvDc–DdVJ
bvDbV/bv1b' 'dF9bvEb' DdF/c 'c6'bø 'dF*d6Fd†Dd†,di)
bvDdV9dv/ bvDc–'dFJ dFDbö1bv3bv* bvDb!CdfHdFHbÆEJb' (c 'bö3
d&3d @dR *d6Fd†Dd†,di'bø 'dF%c–DbvEdi)

Adresse : Rue El Quods, BP 172 - 2098 - Radès
Téléphone : 71 460 100 Fax : 71 442 322
Site Web : www.isetr.rnu.tn

bvDc–Fd†'dc dfGbÂ 'dFBbö3`Â 5.b,Ó#s Ó " " Ö1bv/c2
bvDdv'b!A: 71 460 100 bvDd 'd63: 71 442 322
www.isetr.rnu.tn dVHd&9 bvDd†'bf

R
R
a
a
p
p
p
p
o
o
r
r
t
t

d
d
e
e

P
P
r
r
o
o
j
j
e
e
t
t

d
d
e
e

F
F
i
i
n
n

d
d
,
,
E
E
t
t
u
u
d
d

e
e
s
s

LICENCE APPLIQUEE EN TECHNOLOGIES DE L'INFORMATIQUE

PARCOURS: DEVELOPPEMENT DES SYSTEMES D'INFORMATIONS (DSI)

Entreprise : TANIT WEB

Conception et développement d'une application web
De revue maghrébine

Encadré par :
Encadrant(e) entreprise : Mr Hassen Ben
Hamouda et Mr Anis Harizi
Encadrant(e) ISET : M Eya Cheikh

Réalisé par :
Maaoui karima

Dédicaces

Je dédie ce modeste travail A à la mémoire de mon père disparu trop tôt.
J'espère que, du monde qui est sien maintenant, il apprécie cet humble geste comme
preuve de reconnaissance de la part d'une fille qui a toujours prié pour le salut de son
âme. Puisse Dieu, le tout-puissant, l'avoir en sa sainte miséricorde !

À ma mère Raoudha qui m'a fourni au quotidien un soutien et une confiance sans
failles.

A tous mes sœurs pour leur aide et leur soutien et qui n'ont pas épargnés leurs efforts
pour m'encourager.

A mes chers amis Ameni ,Wassila , Fatma et Arij qui m'ont beaucoup encouragée et
soutenu tout au long de ce e projet avec toute ma reconnaissance et gratitude.

Remerciements

Ce travail n'aurait jamais pu se concrétiser sans l'aide et le soutien de plusieurs personnes que je souhaite vivement remercier et à qui je dédie ce travail.

Monsieur Hassene Ben Hammouda, directeur associé de TANIT WEB pour l'accueil et la confiance qui m'a accordé au regard de ce projet.

Monsieur Anis Harizi, mon encadrant entreprise qui m'a formé et accompagné tout au long de cette expérience professionnelle avec beaucoup de patience et pédagogie.

Madame Eya Cheikh, mon encadrante, qui n'a pas cessé de me prodiguer ses conseils et qui n'a épargné aucun effort pour contribuer à la réussite de mon travail.

Le personnel de TANIT WEB pour l'expérience enrichissante et pleine d'intérêt qu'ils m'ont fait vivre durant la période du stage.

Tous mes professeurs et plus particulièrement les membres de jury qui ont accepté de juger mon travail.

Mon Institut qui m'a donnée l'occasion d'acquérir une formation académique de qualité.

Toutes personnes ayant contribué de près ou de loin à l'élaboration de ce modeste travail

Table des matières

Introduction	11
I. Présentation de l'organisme	
d'accueil :	11
Cadre du	
projet :	11
L'organisme de	
l'accueil :	11
II. Contexte du	
projet :	13
Problématique :	
.....	13
Analyse de	
l'existant :	13
2.1. Étude de	
l'existant :	13
2.2. Critique de l'existant :	
.....	13

2.3. Solution	
proposée :	13
III. Méthodologies de	
travail :	14
Définition de la	
méthode :	14
Les rôles dans	
SCRUM :	14
Conclusion	
.....	16
Introduction	
.....	18
Identification des acteurs du	
système :	18
Identification des	
besoins :	18
2.1 Besoins	
fonctionnels :	18
Besoins non fonctionnels :	
20	
Diagramme de cas d'utilisation globale:	20
Le Backlog produit :	21
Architecture :	23
2.5.1. Architecture	
physique :	23
2.5.2. Architecture	
logique :	24
Environnement de travail :	26
2.6.1. Environnement	
logiciel :	26
Planification du sprint:	29
Conclusion :	
.....	31

Table des figures

Liste des tableaux

Liste des abréviations

- API=Application Programming Interface
- CSS = Cascading Style Skee
- HTML = Hyper Text Markup Langage
- HTTP = Hyper Text Transfer Protocol
- MERN STACK = MongoDB , Epxress, js React js , Node
- JWT = JSON WEB TOKN
- REST =Respresentational State Transfer
- JS = JavaScript Language
- UML = Unified Lodeling Language

9

Introduction générale

Pendant les dernières années l'informatique s'est étendue dans le monde entier bien évidemment ainsi qu'en Tunisie. En effet ce domaine a subi une évolution importante dans notre pays, on entend du jour au lendemain la création de nouvelles applications qui peuvent être gratuites ou payantes accessibles sur le desktop localement ou à l'aide de l'internet ces derniers sont nommées des applications web qui fournissent des services numériques qui rendent la vie plus facile et confortable à mener. Dans ce contexte, l'agence « TANIT WEB » dans laquelle nous avons effectué notre stage de fin d'étude, voudrait mettre en place une application web : Mini site de revues maghrébines. Dès lors, notre projet consiste à concevoir et à réaliser une application web dont le but est de gérer les revues maghrébines qui sont déjà importé à travers l'API de manipuler ses données à travers notre application.

Pour réaliser ce rapport on va suivre la démarche suivante :

- Le premier chapitre intitulé « Cadre Général du projet » présente l'organisme d'accueil, décrit le contexte de notre projet ainsi que la méthodologie adoptée.
- Le deuxième chapitre « Capture de besoins » contient la méthodologie utilisée et la planification du projet explique notre démarche, soit, l'identification des futurs acteurs

de notre système et l'analyse des besoins. Ainsi que la description de l'architecture choisie pour réaliser notre solution.

• Les cinq derniers chapitres sont réservés à la présentation des sprints. Dans chaque sprint, nous décrivons le sprint backlog, l'analyse, la conception et la réalisation.

10

Etude préalable

Plan :

1. Présentation de l'organisme d'accueil
2. Contexte du projet
3. Méthodologies de travail

Chapitre 1. Etude préalable

11

Introduction

Ce chapitre est consacré à la présentation du cadre général du projet ainsi que l'organisme d'accueil « TANIT WEB ». Tout d'abord, nous présentons l'organisme d'accueil, le travail demandé. Ensuite, nous effectuons une étude de l'existant.

Enfin, nous expliquons notre méthode de travail, le chronogramme du projet et le langage de modélisation que nous avons adoptée au cours de notre travail.

I. Présentation de l'organisme d'accueil :

Cadre du projet :

Notre projet de fin d'étude intitulé « Conception et réalisation d'une application web de gestion de revue maghrébine. » a pour but d'obtenir un diplôme de la licence en développement système d'information L'Institut supérieur des études technologiques de Radès (ISET-Radès).

L'organisme de l'accueil :

Le projet a été réalisé au sein de l'entreprise TANIT WEB est une agence digitale globale créée en 2007, elle est basée à Tunis, Imm. Narimène, Rue Chatt Meriem 1073 Montplaisir, et à London 300 Vauxhall Bridge Road, SW1V 1AA.

Depuis sa création, TANIT WEB poursuit son développement et sa stratégie de croissance en

diversifiant ses services et renforçant sa présence internationale.

Son succès repose sur une réelle culture entrepreneuriale, une gestion rigoureuse et une large

autonomie donnée à ses filiales.

Figure 1 : Logo de l'agence

C

Chapitre 1. Etude préalable

12

Figure 2 : L'organigramme de la société

Expertise :

TANIT WEB rassemble sous le même toit Design, Développement, Infrastructure et Webmarketing pour la mise en place d'un dispositif digital performant et rentable.

Proposant

la fois le conseil, le service, le déploiement, la maintenance et le suivi, TANIT WEB se distingue par sa capacité à développer les solutions adéquates, à mettre en place les infrastructures et déployer les stratégies e-marketing nécessaires à la promotion de ces projets.

Les services offerts par TANIT WEB sont les suivants :

• Marketing Digital (Référencement naturel (SEO), Référencement payant (SEA), Stratégie

Web Marketing ...)

• Réseaux sociaux (Gestion des chaînes YouTube, E-Réputation, Marketing d'influence...)

• Web design & développement (Création de site web, Applications mobiles, Objets connectés IoT, Solutions E-commerce...)

• Hébergement web & Cloud (Serveurs managés, Live Streaming, Sécurité et performance...)

Parmi les techniques utilisées :

CRM, Design UX & UI, SEO, Développement web & mobile, Portail média...

Une liste non exhaustive des solutions développées est la suivante :

• BookSpace.tn

• Mosaïque FM

• El Hiwar Ettounsi
• L'Université Centrale

Chapitre 1. Etude préalable

13

• Diva Deep

II. Contexte du projet :

Dans cette partie, nous allons présenter l'étude de l'existant, la problématique, les objectifs ainsi les différentes étapes de déroulement du projet.

Problématique :

La revue papier s'impose comme étant le moyen le plus commun pour se mettre au courant des dernières actualités du monde. La revue en papier se fait oublier de jour en jour.

Et avec la propagation de covid, la palpation des magazines en papier augmente le risque de contamination c'est pour cela les utilisateurs ou les animateurs les passionnés par les journaux évitent vers les journaux et les revues en papier.

Analyse de l'existant :

L'analyse de l'existant permet de détecter les anomalies et les défauts des solutions présentes sur le marché afin de proposer une solution meilleure.

2.1. Étude de l'existant :

Il existe plusieurs solutions permettant la gestion des revues maghrébines. Après l'investigation que nous avons faite au milieu des experts et d'après les recherches, nous

avons noté que peu sont les solutions qui font ça et les plus populaires sont :

- Revue maghrébienne de documentation et d'information.
- Open Journal System

2.2. Critique de l'existant :

Les applications de nos jours, bien qu'elles soient performantes ne satisfont pas la plupart des clients. En effet certaines fonctionnalités ne se font pas de manière automatique, simple et facile. Elles limitent les libertés des utilisateurs.

2.3. Solution proposée :

C'est pour cela on a développé lors de ce projet un site dédié destinée à la promotion des revues maghrébines. La revue numérique offre la possibilité d'accéder à différents articles portant sur le même sujet permettant au lecteur la liberté de choix. Son efficacité réside aussi dans sa mobilité, à l'aide d'un smartphone, d'une tablette et d'un

ordinateur, vous avez accès à votre recherche en un temps-record. Notre application offre

Chapitre 1. Etude préalable

14

également un environnement pour la gestion de contrôle des utilisateurs, la gestion des favoris et des réactions sur les articles. De plus, elle permet d'ajouter un contact à l'application et de notifier les auteurs afin de faire suivi de leurs articles qui contiennent toutes les informations nécessaires.

III. Méthodologies de travail :

Un projet informatique, quelle que soit sa taille et la portée de ses objectifs, nécessite la mise en place d'un planning organisationnel tout au long de son cycle de vie. C'est ainsi qu'est apparue la notion de méthode.

Définition de la méthode :

La méthode de gestion de projet adoptée est l'approche agile Scrum.

Cette méthode privilégie les objectifs à court terme et lieu à des interactions régulières avec

les membres afin de discuter de l'avancement sur le projet.

Scrum est une méthode ou un cadre de gestion. Elle est construite autour d'une équipe Scrum,

d'artefacts et de cérémonies.

C'est un cadre méthodologique incrémentale, organisée en sprint pendant l'équipe scrum et

produire un incrément potentiellement livrable du

L'objectif est d'apporter une valeur ajoutée au produit final.

Cette méthode a pour objectif d'améliorer la productivité des équipes et de favoriser le dialogue entre le client et le prestataire, afin d'optimiser la réussite des projets. Le principe de

Scrum est de développer un logiciel de manière incrémentale en maintenant une liste totalement transparente des fonctionnalités à développer, des demandes d'évolutions ou de

corrections à implémenter (backlog). Avec des livraisons très fréquentes, le client reçoit à

chaque fois un logiciel avec des fonctionnalités nouvelles, et en parfait état de fonctionnement. Pour cela, la méthode s'appuie sur des développements itératifs à un rythme

constant d'une durée de 1 à 4 semaines.

Les rôles dans SCRUM :

Scrum est une méthodologie agile qui consiste à avoir une équipe soudée orientant le projet au fil de son avancement afin d'atteindre un but faisant intervenir trois rôles principaux

qui sont :

• Le gestionnaire de produit (Product Owner) Il a la responsabilité de décider le travail à réaliser.

• Le maître SCRUM (ScrumMaster) Il agit en tant que leader au service de l'équipe en aidant l'équipe et l'organisation à faire le meilleur usage de Scrum.

Chapitre 1. Etude préalable

15

• L'équipe de développement (Development Team) • Il est composé de professionnels qui travaillent pour fabriquer le produit de manière incrémentale avec une série de courtes

périodes de temps appelées Sprints La figure suivante montre le cycle de vie de la méthode

SCRUM et les relations entre le client et le Product Owner.

Le tableau suivant présente l'équipe et les rôles Scrum de ce projet :

Rôles Scrum Personnes

Product Owner Hassene Ben Hammouda , Anis Harizi

Product Master Eya Cheikh

Development Team Karima Maaoui

La figure suivante montre le cycle de vie de la méthode SCRUM et les relations entre le client

et le Product Owner.

Figure 3 : Cycle de vie de la méthode SCRUM

Backlog produit (ou catalogue des besoins)

— Besoins priorisés par le Product Owner.

— Besoins évalués par l'équipe.

Sprint (itération)

— Développement des fonctionnalités du Backlog de sprint.

— Aucune modification du Backlog de sprint possible.

— Mêlée quotidienne (Rencontre quotidienne)

Chapitre 1. Etude préalable

16

— Point de contrôle quotidien de l'équipe.

— Intervention régulières – 2 min par personne.

Sprint Backlog

— Extrait du Backlog produit. Produit livrable livré au Product Owner à la fin du sprint.

Conclusion

Dans ce chapitre, nous avons donné un aperçu du projet en décrivant l'organisme d'accueil, Tanit Web, et le contexte du projet. Nous avons présenté aussi notre méthodologie

de travail pour réaliser notre projet, comme la méthode Scrum. Le reste du rapport sera organisé selon cette méthode. Le chapitre suivant sera consacré au lancement du projet et la spécification de ses besoins.

Chapitre 2. Sprint 0 :Capture des besoins

17

Sprint 0 :Capture des besoins

Plan :

1. Identification des acteurs du système
2. Identification des besoins
3. Diagramme des cas d'utilisation globale
4. Le Backlog Produit
5. Architecture
6. Environnement de travail
7. Planification du sprint

Chapitre 2. Sprint 0 :Capture des besoins

18

Introduction

e chapitre est consacré à la capture des besoins fonctionnels et non fonctionnels de notre système, au pilotage de projet avec scrum, par élaboration du Backlog produit avec une planification des sprints ainsi qu'à la description de l'environnement de travail.

Identification des acteurs du système :

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Pour notre application, nous avons identifié quatre acteurs :

un lecteur , l'auteur, l'administrateur et l'éditeur de notre application.

Figure 4 : Diagramme des acteurs

— L'utilisateur simple de l'application est le lecteur. Une interface sera disponible pour consulter les articles publiés et peut aussi gérer son propre profil, téléchargé, ajouté à la liste

des favoris ...

— L'auteur peut gérer des articles.

— L'éditeur peut consulter des articles et les corriger.

— L'administrateur de notre application a tous les privilèges. Il peut accéder à toutes les

interfaces de l'application, ajouter des utilisateurs, ajouter des types d'articles ...

Identification des besoins :

2.1 Besoins fonctionnels :

Les besoins fonctionnels représentent les actions qu'un système doit être capable d'exécuter et qu'il ne sera opérationnel et qu'il sera opérationnel qu'une fois celles-ci sont

effectués.

C

Chapitre 2. Sprint 0 : Capture des besoins

19

Notre solution doit fournir un ensemble de fonctionnalités répondant aux besoins de nos différents acteurs.

L'application doit permettre à l'éditeur de :

- Gérer son profil.
- Gérer les utilisateurs.

Ajouter et supprimer un compte.

- Envoyer/ recevoir des notifications en temps réel :

Lors d'une ajout d'un article, l'éditeur reçoit une notification synchrone.

Lors de réponse à une demande, l'éditeur envoie un email synchrone à l'auteur.

- Gérer les articles :

Corriger un/des article(s).

Supprimer un/des article(s).

- Gérer les règles :

Ajouter des règles.

Supprimer des règles.

Modifier des règles.

- Gérer les statistiques :

Afficher les statistiques.

- Gérer les demandes de devenir un auteur

- Contacter les autres utilisateurs :

Envoyer des messages.

L'application doit permettre à l'utilisateur de :

- Gérer son compte :

S'inscrire.

Se connecter

- Consulter les articles :

Télécharger un/des article(s).

Ajouter à la liste des favoris.

Recherche d'un article par thème

- Demander de devenir un auteur.
- Contacter les autres utilisateurs :

Envoyer des messages.

L'application doit permettre à l'auteur de :

- Gérer son compte :

S'inscrire.

Se connecter

Chapitre 2. Sprint 0 : Capture des besoins

20

- Consulter les articles.

Voir la liste des j'aime.

Voir la liste des commentaires.

Voir la liste des articles publiées.

- Envoyer/ recevoir des notifications en temps réel :

Envoi des notifications en temps réel à l'éditeur lors de l'envoi d'un article.

Lors de réponse à une demande, l'éditeur envoie un email synchrone à l'auteur.

- Gérer les commentaires :

Ajouter des commentaires aux articles.

- Contacter les autres utilisateurs :

Envoyer des messages.

2.1. Besoins non fonctionnels :

Un besoin non fonctionnel concerne une contrainte à prendre en considération pour mettre en place une solution adéquate aux attentes des concepteurs des architectures dynamiques.

—Sécurité :

Accès sécurisé et personnalisé. Le système doit être sécurisé, Notre plateforme assure

pour chaque login un mot de passe bien sécurisé. Mise en place d'un système pour tracer les

accès dont la gestion se base sur une politique de gestion de rôle. Par ailleurs, l'accès aux

interfaces est fait suite à une analyse des rôles.

— L'ergonomie :

Notre application assure des interfaces bien présentées et bien guidées.

— Performance :

Le temps d'exécution du système doit être faible.

2.2. Diagramme de cas d'utilisation globale:

Chapitre 2. Sprint 0 :Capture des besoins

21

Figure 5 : Diagramme de cas d'utilisation globale

2.3. Le Backlog produit :

Nous allons présenter les différentes tâches des acteurs dans le backlog produit :

ID Fonctionnalité User

Story

Tache

Chapitre 2. Sprint 0 :Capture des besoins

22

1

Authentification

1.1

En tant qu'utilisateur, je dois saisir un username et un mot de passe pour accéder à l'application.

1.2 En tant qu'utilisateur, je reçois un email de vérification de compte.

2

Gestion de profil

2.1

En tant qu'utilisateur, je peux consulter/modifier mon profil.

3

Gestion des articles 3.1

En tant qu'auteur, je peux publier un article.

3.2

En tant qu'admin, je peux consulter la liste des articles et exporter en CSV.

3.3

En tant qu'utilisateur, je peux consulter la liste des articles publiées et telecharger et ajouter aux favoris ,

mettre des réactions (commentaire, j'aime) .

3.4

En tant qu'auteur, je peux ajouter/modifier un article.

3.5

En tant qu'auteur, je peux consulter les réactions sur ses articles.

3.6

En tant qu'editeur, je peux corriger les articles.

4

Gestion des utilisateurs

4.1 En tant qu'admin, je peux ajouter des utilisateurs et consulter la liste des utilisateurs et exporter en CSV.

5

Demande d'etre un auteur

5.2 En tant qu'utilisateurs, je peux enovyer une demande d'etre un auteur.

5.3 En tant qu'admin, je peux accepter/ refuser la demande d'ajout des auteurs.

6

Gestion des règles 6.1 En tant qu'admin, je peux consulter la liste des règles et exporter en CSV , ajouter/modifier/supprimer des règles

Chapitre 2. Sprint 0 :Capture des besoins

23

7

Gestion des types 7.1 En tant qu'admin, je peux consulter la liste des types et exporter en CSV , ajouter/modifier/supprimer des types

8

Gestion des themes 8.1 En tant qu'admin, je peux consulter la liste des themes et exporter en CSV , ajouter/modifier/supprimer des themes

9

Envoit des messages 9.1 En tant qu'utilisateur, je peux contacter les autres utilisateurs en message privé

10

Envoi des emails 10.1 En tant qu'utilisateur, je peux contacter l'administrateur avec email

11

Gestion Feedback 11.1 En tant qu'editeur/auteur/lecteur, je peux ajouter un Feedback

11.2 En tant qu'admin, je peux supprimer un Feedback

12

Envoi des emails 12.1 En tant qu'utilisateur, je peux contacter l'administrateur avec email

13

Gestion des favoris 13.1 En tant que lecteur, je peux ajouter un article aux liste des favoris

Architecture :

2.5.1. Architecture physique :

L'architecture physique de notre application (où également nommée l'architecture technique d'une application), décrit l'ensemble des composants matériels constituant l'application web. Dans ce contexte, notre application est constituée principalement d'un

client, serveur de base de données MONGODB et un serveur web. Nous allons détailler

l'architecture physique de notre application web dans la figure suivante :

Figure 6 : Architecture physique

Chapitre 2. Sprint 0 :Capture des besoins

24

L'architecture est composée :

— D'un niveau client gérant la couche présentation : (FrontEnd) et qui permettant à son utilisateur d'accéder à l'application via internet..

— D'un niveau serveur: qui contient l'application Back-End celle qui accède aux données à la demande de l'application Front-End afin de retourner un contenu compréhensible et représentable dans l'application frontale.

— D'un noeud physique contenant les serveurs de la base de données utilisé pour stocker la partie persistance des données de notre application.(MongoDB)

2.5.3. Architecture technique :

A. Le modele REST API(RESTFUL API) (Backend) :

L'API RESTful est un style architectural d'interface de programmation d'application (API)

qui utilise des requêtes HTTP pour accéder aux données et les utiliser. Ces données peuvent

être utilisées pour les types de données GET, PUT, POST et DELETE. Ces types de données

font référence aux opérations de lecture, de mise à jour, de création et de suppression liées

aux ressources. Un système reposant se compose d'un :

• Le client qui a demandé la ressource.

• Serveur qui a les ressources.

Figure 7 : Image correspond rest API pour multi-platform

• Couches ou composants de l'API REST Node.js :

En règle générale, le service RESTful accomplira la tâche de servir la demande dans l'approche en couches. Dans le monde Node.js, différents composants sont disponibles pour

implémenter la fonctionnalité API. Il s'agit de la route, du controller, middleware ,et du modèle.

Routes :

Aide à créer une liste de tous vos routes d'applications et à les associer à un fichier de contrôleur contenant le code d'implémentation.

Via Routes Express, la requête HTTP qui correspond à un itinéraire sera vérifiée par CORS

Middleware avant d'arriver à la couche de sécurité.

La couche de sécurité comprend :

Chapitre 2. Sprint 0 :Capture des besoins

25

Middleware d'authentification JWT : vérifier l'inscription, vérifier le jeton

Middleware d'autorisation : vérifier les rôles de l'utilisateur avec enregistrement dans la base

de données.

Si ces Middleware génèrent une erreur, un message sera envoyé en tant que réponse HTTP.

Controlleur :

Les contrôleurs interagissent avec MONGODB Database envoient une réponse HTTP (jeton, informations utilisateur, données basées sur les rôles...) au client.

Model :

Ce composant est la couche d'accès aux données pour récupérer et enregistrer les documents. La couche de service invoquera les modèles pour effectuer des actions sur le

document dans la base de données via le modèle. Un modèle représente le document qui peut

être créé, mis à jour, supprimé et extrait de la base de données.

B. Architecture du Frontend :

– Components : ce sont des classes ou des fonctions javascript dans lesquelles nous définissons la logique de l'interface utilisateur, elles acceptent d'autres propriétés (props) à

venir d'autres composants et renvoient la façon dont notre interface utilisateur prendra forme.

Il y a un composant racine qui enveloppe tous les autres composants (App.js) comme des

enfants, c'est pourquoi nous nous assimilons à une structure arborescente.

Le composant App est un conteneur avec React Router. En se basant sur l'état, la barre de

navigation peut afficher ses éléments.

– Les composants Login & Register ont un formulaire pour la soumission de données Ils

appellent des méthodes à partir des fonctions pour faire une demande de connexion / enregistrement.

– les fonctions utilisent axios pour effectuer des requêtes HTTP. Il stocke ou obtient également JWT à partir du stockage local du navigateur à l'intérieur de ces méthodes.

– Les composants BoardEditor, BoardLecteur, BoardAuthord, BoardAdmin seront affichés

par état user.roles.

Chapitre 2. Sprint 0 : Capture des besoins

26

Figure 8 : Architecture du Frontend avec axios

— Redux store : il s'agit d'un conteneur de gestion d'état global pour notre application React.

Tout composant connecté au store Redux peut envoyer un appel d'action, après les appels

d'actions transmettent des propriétés aux réducteurs(reducers), chaque réducteur (reducer) a

une action définie et un état donné, le rôle des réducteurs (reducers) est de fournir le nouvel

état aux composants.

Figure 9 : Architecture du Frontend avec Redux

Environnement de travail :

L'architecture matérielle utilisée pour la réalisation de l'application est la suivante:

Marque MSI

Processus 11th Gen Intel(R) Core(TM) i7-11800H @

2.30GHz 2.30 GHz

Mémoire 32GO

Système d'exploitation Windows 10

2.6.1. Environnement logiciel :

Pour le développement front-end, nous avons utilisé un nouvel éditeur développé récemment par Microsoft : c'est Visual code studio. Cet éditeur a gagné une très bonne satisfaction au sein de développeur front-end, c'est un outil open source qui facilite le développement moyennant les langages JavaScript et TypeScript.

Chapitre 2. Sprint 0 : Capture des besoins

27

Figure 10 : Logo Visual Studio Code

— ReactJS : C'est une bibliothèque JavaScript libre développée par Facebook depuis 2013.

Conçue principalement pour faciliter la création d'application web (Single page application), en utilisant les composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état. Dans notre application, on utilise la bibliothèque ReactJS pour le développement frontal.

Figure 11 : Logo ReactJs

ReactJS : C'est une Redux gestionnera l'état pour les applications JavaScript. Vous pouvez utiliser Redux principalement avec React ou autres framework et bibliothèque. Il possède un large écosystème d'add-ons utilisable.

Figure 12 : Logo Redux

— JavaScript : C'est un langage de programmation qui est inclus dans le code HTML. Il permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes.

Figure 13 : Logo Javascript

— HTML5 et CSS3 : « Hyper Text Markup Language 5 » HTML5 considéré comme l'une des nouvelles Technologies les plus importantes, qui étaient émergées en 2010, et qui sont encore un travail ne sont pas totalement finalisées par le W3C. Elle est un langage de

balisage conçu pour la création des pages Web. Quant au CSS3 « Cascading Style Sheets

» permet d'arrondir les images, faire des ombres sur les divs, des ombres sur du texte,

des polices de caractères plus fun, des bordures d'images, 20 Chapitre 2. Sprint 0

Chapitre 2. Sprint 0 : Capture des besoins

28

: Capture des besoins etc. Et surtout l'ajout de l'animation. Il utilisait pour la description des styles et la disposition des éléments des pages HTML.

Figure 14 : Logo Html/Css

— Postman : Postman est un logiciel qui se focalise sur les tests des API en interrogeant les

micro Services, offrant ainsi de nombreuses fonctionnalités, une prise en main rapide et une interface graphique agréable .

Figure 15 : Logo Postman

— Visual Paradigm : Visual Paradigm (VP) est un logiciel de création de diagrammes dans le cadre d'une programmation. Il possède plusieurs fonctionnalités qui facilite énormément la modélisation en ULM.

Figure 16 : Logo Visual Paradigm

— Github:

GitHub est une société à but lucratif qui offre un service d'hébergement de référentiel Git basé sur le cloud. Essentiellement, il est beaucoup plus facile pour les individus et les équipes d'utiliser Git pour le contrôle de version et la collaboration.

Figure 17 : Logo Github

— MongoDB:

MongoDB est un programme de base de données multiplateforme orienté document et disponible en source. Classé comme un programme de base de données NoSQL, MongoDB utilise des documents de type JSON avec des schémas facultatifs. MongoDB est développé par MongoDB inc. et sous licence Server Side Public License (SSPL).

Chapitre 2. Sprint 0 : Capture des besoins

29

Figure 18 : Logo mongoDB

— MongoDB Atlas:

MongoDB Atlas est une base de données cloud entièrement gérée développée par les mêmes personnes qui construisent MongoDB. Atlas gère toute la complexité du déploiement, de la gestion et de la réparation de vos déploiements sur le fournisseur de services cloud de votre choix (AWS, Azure et GCP).

Figure 19 : Logo mongoDB Atlas

—Node js:

Node.js est un environnement d'exécution JavaScript back-end open source, multiplateforme

qui s'exécute sur le moteur V8 et exécute le code JavaScript en dehors d'un navigateur Web.

Figure 20 : Logo NodeJS

— Express js:

Express.js, ou simplement Express, est un framework d'application Web back-end pour Node.js, publié en tant que logiciel gratuit et open source sous la licence MIT. Il est conçu pour créer des applications Web et des API. Il a été appelé le framework de serveur standard de facto pour Node.js.

Figure 21 : Logo ExpressJs

Arborescence du projet:

La figure ci-dessous illustre l'arborescence ou la structure de notre projet :

Chapitre 2. Sprint 0 :Capture des besoins

30

Figure 22 : Arborescence de projet front et back

Planification des sprints:

Sprint Fonctionnalités Durée (en Jours)

Sprint 1 Authentification

Gestion des utilisateurs

Gestion de profil

Sprint 2 Gestion des types

Gestion des thèmes

Envoi/réponse des demandes
d'être auteur

Sprint 3

Gestion des articles

Gestion des règles

Correction des articles

Gestion des notifications
pour le correcteur

Chapitre 2. Sprint 0 :Capture des besoins

31

Sprint 4 Gestion des réactions

Recherche des articles avec
filtrage par mots clés
Recherche des articles avec
filtrage par thème
Recherche des articles avec
filtrage par type

Sprint 5 Gestion des feedbacks
Envoi des messages et des
emails
Migration vers l'application
responsive

Conclusion :

Dans ce chapitre nous allons détailler les fonctionnalités offertes par notre système ainsi que leurs associations avec les acteurs décrits auparavant à l'aide du diagramme de cas d'utilisation globale de notre application, par la suite nous avons présenté le Backlog de notre système. Ainsi nous avons détaillé la phase de planification des sprints. Enfin nous avons choisi l'architecture de notre projet ainsi que notre environnement de travail.

32

Sprint 1 :

Plan :

1. Présentation de l'organisme Objectif du Sprint
2. Contexte du Analyse
3. Méthodologies de Analyse spécifique
4. Conception
5. Réalisation

Chapitre 3. Sprint 1

Introduction :

Dans ce chapitre nous allons concevoir et réaliser les fonctionnalités du sprint 1 de notre projet. Nous allons en premier lieu présenter le sprint backlog, où nous allons détailler

les fonctionnalités demandées, l'étape suivante consiste à faire l'analyse et la conception des fonctionnalités spécifiées. En fin, nous montrons l'output de ce sprint.

Phase de preparation :

2.1. But :

L'objectif du sprint est une phrase claire et compréhensible de tous qui permet de définir

l'objectif à atteindre sur le sprint en cours. Celui-ci sera fixe tout au long du sprint qui se déroule.

Le but de notre premier sprint étant en premier lieu d'autoriser la personne à accéder à

certaines ressources sécurisées et en second lieu de gérer les différents utilisateurs de notre application et gérer le profil.

2.2. Analyse :

Cette partie sera consacrée à l'analyse du premier sprint de notre projet, nous allons

présenter les backlog produit de sprint 1, les diagrammes de cas d'utilisation et leurs

descriptions textuelles ainsi que le diagramme séquence système.

2.2.5. Sprint Backlog:

Le backlog sprint est une liste des tâches identifiées par l'équipe Scrum à réaliser durant du premier sprint. Lors de la réunion de planification du sprint, l'équipe sélectionne un

certain nombre de produits des éléments de backlog, généralement sous la forme de user

stories, et identifie les tâches nécessaires pour user stories. La plupart des équipes estiment

également combien d'heures chaque tâche nécessitera achevée. Dans la figure suivante, nous

avons l'arriéré du sprint 1, l'estimation est en jours.

ID Themes User Story Estimation
(en jours)

1

Authentication

En tant qu'utilisateur je veux
m'authentifier
10 jours
En tant qu'utilisateur, je souhaite
vérifier le compte via un lien envoyé

Chapitre 3. Sprint 1

34
par le système par e-mail.
En tant qu'utilisateur je peut oublié
mot de passe et le système va
m'envoyer un nouveau mot de passe.

2

Gestion des utilisateurs

En tant qu'admin je veux afficher les
utilisateurs .

2 jours

En tant qu'admin je veux modifier un
utilisateur.

En tant qu'admin je veux supprimer un
utilisateur.

En tant qu'admin je veux exporter les
utilisateurs en Csv.

3

Gestion de profil

En tant qu'utilisateur je veux consulter
mon profil

3 jours

En tant qu'utilisateur je veux modifier
mon profil

Phase de preparation :

2.1. Modelisation fonctionnelle:

Les besoins à réaliser dans ce premier sprint, ont été spécifiés. Nous passons maintenant à la présentation des diagrammes de cas d'utilisation qui ont pour but de donner une vue globale sur l'ensemble des fonctionnalités fournies par l'application, ainsi que les descriptions textuelles qui décrivent les scénarios de chaque cas.

2.1.1. Diagramme de cas d'utilisation global du Sprint 1:

Figure 23 : Diagramme de cas d'utilisation global du sprint 1

Chapitre 3. Sprint 1

35

2.1.2. Raffinement de cas d'utilisation « Mot de passe oublié »:

La figure ci-dessous décrit le diagramme détaillé de cas d'utilisation « mot de passe oublié » :

2.1.3. Raffinement de cas d'utilisation « gérer les utilisateurs »:

La figure ci-dessous décrit le diagramme détaillé de cas d'utilisation « gérer les utilisateurs » :

Figure 24 : Diagramme de cas d'utilisation Gérer les utilisateurs

2.1.1. Raffinement de cas d'utilisation « gérer profil »:

La figure ci-dessous décrit le diagramme détaillé de cas d'utilisation « gérer les utilisateurs » :

Chapitre 3. Sprint 1

36

Figure 25 : Diagramme de cas d'utilisation Gérer les utilisateurs

a. Description textuelle de diagramme de cas d'utilisation « s'authentifier »

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas d'utilisation « s'authentifier » :

Sommaire

Titre :

Authentification

Auteur :

Administrateur, Lecteur, Editeur, Auteur

Description d'enchaînement

Pré-conditions :

Post-Conditions :

Authentification effectuée avec succès

Scénario nominal

1. L'utilisateur accède à l'interface d'authentification
2. L'utilisateur remplit le formulaire et l'envoie
3. L'application front fait appel à l'application back pour vérifier les données
4. Redirection de l'utilisateur vers son espace
5. FIN.

Chapitre 3. Sprint 1

37

Scénario d'exceptions

1. Username ou password incorrect

Ø L'application affiche une alerte indiquant l'erreur.

b. Description textuelle de diagramme de cas d'utilisation « mot de passe oublié »

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas d'utilisation « s'authentifier » :

Sommaire

Titre :

Redéfinir un nouveau mot de passe

Auteur :

Administrateur, Lecteur, Editeur, Auteur

Description d'enchaînement

Pré-conditions :

Post-Conditions :

Changer de mot de passe

Scénario nominal

1. L'utilisateur accède à la page de connexion

2. L'utilisateur appuie sur le bouton « Mot de passe oublié ? »

3. L'utilisateur saisi son adresse email

4. L'utilisateur reçoit un email sur l'adresse entrée

5. L'utilisateur appuie sur le lien envoyé dans l'email

6. L'utilisateur est redirigé vers une page de ré entrer l'adresse email ou il entre

l'adresse email et l'utilisateur reçoit un email sur l'adresse entrée ou il a trouvé un nouveau mot de passe.

Chapitre 3. Sprint 1

38

7. FIN.

Scénario d'exceptions

1. Email incorrect

Ø L'application affiche une alerte indiquant l'erreur.

c. Description textuelle de diagramme de cas d'utilisation « Gestion des utilisateurs »

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas d'utilisation « gestion des utilisateurs » :

Sommaire

Titre :

Gérer un utilisateur

Auteur :

Administrateur

Description d'enchaînement

Pré-conditions :

L'administrateur est connecté

Post-Conditions :

Gestion des utilisateurs avec succès

Scénario nominal

1. L'administrateur consulte l'interface de la liste des utilisateurs.
2. L'administrateur demande l'ajout, modifie ou supprime un utilisateur.
3. Le système affiche le formulaire ou la liste des utilisateurs.
4. L'administrateur remplit le formulaire et valide
5. L'administrateur choisit l'utilisateur, modifier ou supprimer et valide

Chapitre 3. Sprint 1

39

6. Le système affiche une alerte de succès

FIN.

Scénario alternative

1. Email ou username déjà existant :
Ø L'application affiche une alerte indiquant l'erreur.

d. Description textuelle de diagramme de cas d'utilisation « Gestion de profil »

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas d'utilisation « gestion de profil » :

Sommaire

Titre :

Gérer un profil

Auteur :

Utilisateur

Description d'enchaînement

Pré-conditions :

L'utilisateur est connecté

Post-Conditions :

Gérer profil ajouté avec succès

Scénario nominal

1. L'utilisateur consulte l'interface de profil.
2. L'utilisateur clique sur le bouton modifier.
3. L'utilisateur remplit le formulaire et clique sur enregistrer.
4. Le système affiche une alerte de succès
5. FIN.

Chapitre 3. Sprint 1

40

Scénario alternative

Conception :

3.1. Diagrammes de séquence détaillée :

3.1.1. Diagramme de séquence de « s'authentifier et mot de passe oublié »

Figure 26 : Diagramme de séquence "s'authentifier et mot de passe oublié"

Chapitre 3. Sprint 1

41

3.1.2. Diagramme de séquence de « gérer les utilisateurs »

Figure 27 : Diagramme de séquence de « gérer les utilisateurs »

3.1.3. Diagramme de séquence de « gérer profil»

Réalisation :

4.1 Page d'accueil :

C'est la première interface de notre site

Chapitre 3. Sprint 1

42

4.2 S'authentifier

L'accès à la page principale du site s'effectue à l'aide de l'authentification. Pour se connecter, l'utilisateur saisit son login et mot de passe ou bien il peut créer un nouveau

compte.

Chapitre 3. Sprint 1

43

En cas d'incorrectes identifiants le système affiche de message.

4.3 Mot de passe oublié :

Chapitre 3. Sprint 1

44

4.4 Gestion des utilisateurs :

Chapitre 3. Sprint 1

45

Chapitre 3. Sprint 1

46

4.5 Gestion de profil :

Conclusion

Dans ce chapitre, nous avons créé l'artefact du projet avec un modèle principal, l'artefact les services sont sécurisés à l'aide de JWT. L'artefact comprend des services d'authentification, page de connexion, services pour gestion et l'importation des utilisateurs et services de gestion de profil. Dans le chapitre qui suit, nous allons produire un nouveau sprint couvrant les fonctionnalités suivantes : Gestion des demandes d'être auteur ,gestion des thèmes , gestion les types d'article.

47

Sprint 2 :

Plan :

1. Présentation de l'organisme Objectif du Sprint
2. Contexte du Analyse
3. Méthodologies d'Analyse spécifique
4. Conception
5. Réalisation

Chapitre 4. Sprint 2

48

Introduction :

Après avoir fait le sprint d'authentification, la gestion des utilisateurs , la gestion de profil on va passer au deuxième sprint qui consiste à se concentrer sur la gestion des types , la gestion des thèmes , la gestion des demandes d'être auteur et on va exposer les différents diagrammes d'analyse et de conception et ensuite on va présenter la phase de réalisation.

Phase de préparation :

5.1 But :

Le but de ce deuxième sprint est de gérer les types d'article, gérer des thèmes , la gestion des demandes d'être auteur.

5.2 Analyse :

Cette partie sera consacrée à l'analyse du premier sprint de notre projet, nous allons présenter les backlog produit de sprint 1, les diagrammes de cas d'utilisation et leurs descriptions textuelles ainsi que le diagramme séquence système.

5.2.2 Sprint Backlog

ID Thèmes User Story Estimation
(en jours)

1

Gestion des thèmes

En tant qu'administrateur je veux afficher les thèmes.

2 jours

En tant qu'administrateur je veux modifier un thème.

En tant qu'administrateur je veux supprimer un thème.

En tant qu'administrateur je veux exporter les thèmes en Csv.

2

Gérer les types d'article

En tant qu'administrateur je veux consulter la liste des types d'article.

4 jours

En tant qu'administrateur je veux modifier un type d'article.

En tant qu'administrateur je veux supprimer un type d'article.

En tant qu'administrateur je veux exporter les types d'article en Csv.

3

Gérer les demandes d'être

En tant qu'administrateur je veux 15jours

Chapitre 4. Sprint 2

49

auteur

accepter/refuser la demande

En tant que lecteur je veux envoyer une demande d'être auteur

Phase de préparation :

2.1 Modélisation fonctionnelle:

Cette partie est consacrée à la modélisation des diagrammes de cas d'utilisation ainsi que les descriptions textuelles des interactions entre le système et l'utilisateur.

2.1.2. Diagramme de cas d'utilisation global du Sprint 2:

2.1.3. Raffinement de cas d'utilisation « Gérer les types d'article »:

2.1.4. Raffinement de cas d'utilisation « Gérer les thèmes»:

Chapitre 4. Sprint 2

50

2.1.5. Raffinement de cas d'utilisation « Gérer les demandes »:

a. Description textuelle de diagramme de cas d'utilisation « Gestion des thèmes»

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas d'utilisation « gestion des thèmes» :

Sommaire

Titre :

Gérer des thèmes

Auteur :

Administrateur

Description d'enchaînement

Chapitre 4. Sprint 2

51

Pré-conditions :

L'administrateur est connecté

Post-Conditions :

Gestion thèmes avec succès

Scénario nominal

1. L'administrateur consulte l'interface de la liste des thèmes.

2. L'administrateur demande l'ajout, modifie ou supprime un thème.
3. Le système affiche le formulaire ou la liste des thèmes.
4. L'administrateur remplit le formulaire et valide
5. L'administrateur choisit le type, modifier ou supprimer et valide
6. Le système affiche une alerte de succès
7. FIN.

Scénario alternative

1. Libellé déjà existant :
 Ø L'application affiche une alerte indiquant l'erreur.

a. Description textuelle de diagramme de cas d'utilisation « Gestion de demande d'être auteur »

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas d'utilisation « gestion de demande d'être auteurs » :

Sommaire

Titre :

Demande d'être Auteur

Chapitre 4. Sprint 2

52

Auteur :

Lecteur, Administrateur

Description d'enchaînement

Pré-conditions :

Le Lecteur est connecté

L'administrateur est connecté

Post-Conditions :

Envoyer / répondre à une demande

Scénario nominal

1. Le Lecteur peut envoyer une demande
 2. Le lecteur remplit le formulaire et clique sur envoyer.
 3. L'administrateur répond aux demandes (accepté, refusé)
 4. Le système envoie un email au cas d'accepter la demande
- FIN.

Conception :

3.1. Diagrammes de séquence détaillée :

3.1.1. Diagramme de séquence de « Gérer des thèmes » :

Chapitre 4. Sprint 2

53

3.1.2. Diagramme de séquence de « Gérer des types d'article » :

3.1.1. Diagramme de séquence de « Gérer des demandes » :

Chapitre 4. Sprint 2

54

3.1.2. Diagramme de séquence de « Corriger les articles » :

Chapitre 4. Sprint 2

55

3.1.3. Diagramme de séquence de « Consulter Articles » :

Réalisation :

4.1 Gestion des thèmes :

Chapitre 4. Sprint 2

56

Chapitre 4. Sprint 2

57

4.2 Gestion des types d'article :

Chapitre 4. Sprint 2

58

4.3 Gestion des demandes d'être auteur :

Conclusion

Chapitre 4. Sprint 2

59

Dans ce chapitre, nous avons terminé la gestion et l'importation des thèmes et des types d'article et services de gestion des demandes d'être auteur. Dans le chapitre qui suit, nous allons produire un nouveau sprint couvrant les fonctionnalités suivantes : Gestion des règles, Gestion des articles, la correction des articles.

60

Sprint 3

Plan :

1. Présentation de l'organisme Objectif du Sprint
2. Contexte du Analyse
3. Méthodologies d'Analyse spécifique
4. Conception
5. Réalisation

Chapitre 5. Sprint 3

61

Introduction :

Après avoir fait le sprint d'authentification, la gestion des thèmes , la gestion des types d'article et gestion des demandes d'être auteur on va passer au troisième sprint qui consiste à se concentrer sur la gestion des règles , la gestion des thèmes , la gestion des demandes d'être auteur et on va exposer les différents diagrammes d'analyse et de conception et ensuite on va présenter la phase de réalisation.

Phase de préparation :

1.1 But :

Le but de ce troisième sprint est de gérer les types d'article, gérer des thèmes, la gestion des demandes d'être auteur.

1.2 Analyse :

Cette partie sera consacrée à l'analyse du premier sprint de notre projet, nous allons présenter les backlog produit de sprint 1, les diagrammes de cas d'utilisation et leurs descriptions textuelles ainsi que le diagramme séquence système.

1.2.2 Sprint Backlog

ID Thèmes User Story

1

Gestion des règles

En tant qu'administrateur je veux afficher les règles.

En tant qu'administrateur je veux
modifier une règle.

En tant qu'administrateur je veux
supprimer une règle.

En tant qu'administrateur je veux
exporter les règles en Csv.

2

Gérer des articles

En tant qu'auteur je veux écrire un
article.

En tant qu'auteur je veux modifier un
article qu'il a écrit.

3

Gestion des notifications
pour le correcteur

En tant que correcteur je veux reçoit
une notification lorsque un nouveau
article écrite par l'auteur.

4

Correction des articles

En tant que correcteur je veux

Chapitre 5. Sprint 3

62

consulter la liste des articles.

En tant que correcteur je veux corriger
un article.

En tant que correcteur, je veux mettre
un commentaire s'il y a un champ dans
un article incorrect.

5

Consulter les articles publiés

En tant que lecteur je veux consulter
les articles publiés

Phase de préparation :

2.1 Modélisation fonctionnelle:

Cette partie est consacrée à la modélisation des diagrammes de cas d'utilisation ainsi que les descriptions textuelles des interactions entre le système et l'utilisateur.

2.1.6. Diagramme de cas d'utilisation global du Sprint 3:

2.1.7. Raffinement de cas d'utilisation « Gérer des règles »:

Chapitre 5. Sprint 3

63

2.1.8. Raffinement de cas d'utilisation « Gérer des articles »:

2.1.9. Raffinement de cas d'utilisation « Consulter les notifications»:

Chapitre 5. Sprint 3

64

2.1.10. Raffinement de cas d'utilisation « Corriger les articles »:

2.1.11. Raffinement de cas d'utilisation « Consulter les articles publiés »:

a. Description textuelle de diagramme de cas d'utilisation « Consulter les notifications»

Chapitre 5. Sprint 3

65

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas

d'utilisation «consulter les notifications» :

Sommaire

Titre :

Consulter les notifications

Auteur :

Correcteur
Description d'enchaînement
Pré-conditions :

Le correcteur est connecté

Post-Conditions :

Consulter les notifications

Scénario nominal

1. Le correcteur peut consulter la liste des notifications.
2. FIN.

b. Description textuelle de diagramme de cas d'utilisation « Corriger les articles»

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas d'utilisation « corriger les articles» :

Sommaire

Titre :

Corriger les articles

Auteur :

Correcteur

Description d'enchaînement

Pré-conditions :

Le correcteur est connecté

Post-Conditions :

corriger les articles

Chapitre 5. Sprint 3

66

Scénario nominal

1. Le correcteur peut consulter et lire un article
2. Le correcteur peut ajouter un commentaire au cas de trouver une erreur à un champ de l'article
3. Le correcteur peut publier/rejeter un article
4. Le système envoyer un email à l'auteur
5. FIN.

Conception :

3.2. Diagrammes de séquence détaillée :

3.2.1. Diagramme de séquence de «Corriger les articles» :

Réalisation :

4.1 Gestion des règles :

Chapitre 5. Sprint 3

67

Chapitre 5. Sprint 3

68

4.2 Consulter les notifications :

4.3 Corriger les articles :

Chapitre 5. Sprint 3

69

4.4 Gérer les articles :

Chapitre 5. Sprint 3

70

4.1 Consulter les articles publiés:

Conclusion

Dans ce chapitre, nous avons terminé la gestion et l'importation des règles, services de gestion des articles et services de correction des articles. Dans le chapitre qui suit, nous allons

produire un nouveau sprint couvrant les fonctionnalités suivantes : Gestion des réactions,

Consulter article : recherche des articles avec filtrage par mots clés, recherche des articles

avec filtrage par thème, recherche des articles avec filtrage par type, et consultation des articles publiés.

71

Sprint 4

Plan :

1. Présentation de l'organisme Objectif du Sprint
2. Contexte du Analyse
3. Méthodologies d'Analyse spécifique
4. Conception
5. Réalisation

Chapitre 6. Sprint 4

72

Introduction :

Après avoir fait le sprint de la gestion des règles, la correction des articles et consultation

des notifications on va passer au quatrième sprint qui consiste à se concentrer sur la gestion

des réactions sur les articles (j'aime, commentaire, vue), recherche des articles avec filtrage

par mots clés, par thème, par type et on va exposer les différents diagrammes d'analyse et de

conception et ensuite on va présenter la phase de réalisation.

Phase de préparation :

1.1 But :

Le but de ce quatrième sprint est de gérer les réactions sur les articles (j'aime, commentaire, vue), rechercher des articles avec filtrage par mots clés et rechercher des articles avec filtrage par thème

1.2 Analyse :

Cette partie sera consacrée à l'analyse du premier sprint de notre projet, nous allons présenter les backlog produit de sprint 1, les diagrammes de cas d'utilisation et leurs descriptions textuelles ainsi que le diagramme séquence système.

1.2.2 Sprint Backlog

ID Thèmes User Story Estimation
(en jours)

1

Gestion des réactions sur
l'article

En tant que consultant je veux mettre
une j'aime sur un article.

1jours

En tant que consultant je veux mettre
un commentaire sur un article.

2

Recherche des articles avec
filtrage par mots clés

En tant que consultant je veux
rechercher un article par un mot clés
données.

3

Recherche des articles avec
filtrage par thème

En tant que consultant je veux
rechercher un article par un thème
sélectionné.

4

Recherche des articles avec
filtrage par type

En tant que consultant je veux
rechercher un article par un type
sélectionné.

5

Recherche des articles avec
En tant que consultant je veux

Chapitre 6. Sprint 4

73

filtrage par le plus lus
rechercher un article par le plus grand
nombre de vues.

Phase de préparation :

2.1 Modélisation fonctionnelle:

Cette partie est consacrée à la modélisation des diagrammes de cas d'utilisation
ainsi

que les descriptions textuelles des interactions entre le système et l'utilisateur.

2.1.12. Diagramme de cas d'utilisation global du Sprint 4:

a. Description textuelle de diagramme de cas d'utilisation « Recherche les articles»

La description textuelle représentée dans le tableau ci-dessous montre le scénario du
cas

d'utilisation «recherche des articles» :

Sommaire

Titre :

Rechercher les articles

Auteur :

Lecteur

Description d'enchaînement

Pré-conditions :

Le lecteur est connecté

Chapitre 6. Sprint 4

74

Post-Conditions :

Rechercher les articles

Scénario nominal

1. Le lecteur peut afficher et lire un article
2. Le lecteur peut trouver des articles par thème
3. Le lecteur peut trouver des articles par mot clés
4. Le lecteur peut trouver des articles par type

5. Le lecteur peut trouver des articles par nombre de vues

6. FIN.

b. Description textuelle de diagramme de cas d'utilisation « Gestion des réactions sur les articles»

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas

d'utilisation «gestion des réactions sur mes articles» :

Sommaire

Titre :

réactions sur les articles

Auteur :

Lecteur

Description d'enchaînement

Pré-conditions :

Le lecteur est connecté

Post-Conditions :

réactions sur les articles

Scénario nominal

1. Le lecteur peut afficher et lire un article

2. Le lecteur peut mettre une j'aime sur un article

Chapitre 6. Sprint 4

75

3. Le lecteur peut mettre un commentaire sur un article

4. Le lecteur peut consulter la liste des j'aime/commentaire(s) sur un article

5. FIN.

Conception :

3.3. Diagrammes de séquence détaillée :

3.3.1. Diagramme de séquence de «Rechercher des articles par thème»

3.3.2. Diagramme de séquence de «Rechercher des articles par mot clés / par type»:

3.3.3. Diagramme de séquence de «Gestion des réactions» :

Chapitre 6. Sprint 4

76

Réalisation :

4.1 Gestion des réactions sur les articles:

4.1.1. Gestion des commentaires sur les articles :

Chapitre 6. Sprint 4

77

4.1.2. Gestion des j'aime sur les articles :

4.2 Recherche sur les articles:

4.2.1. Recherche des articles par mot clés :

Chapitre 6. Sprint 4

78

4.2.2. Recherche des articles par type :

4.2.3. Recherche des articles par thème :

Chapitre 6. Sprint 4

79

4.2.4. Recherche des articles par le plus lus (le plus grand nombre de vues) :

Conclusion

Dans ce chapitre, nous avons terminé la gestion des réactions, services de recherche des articles par mots clés, type, thème. Dans le chapitre qui suit, nous allons produire un nouveau sprint couvrant les fonctionnalités suivantes : L'envoi des messages entre les différents utilisateurs, L'envoi d'un email avant de connecter à notre plateforme, l'ajout d'un feedback, Et la gestion des feedbacks.

80

Sprint 5

Plan :

1. Présentation de l'organisme Objectif du Sprint
2. Contexte du Analyse
3. Méthodologies d'Analyse spécifique
4. Conception
5. Réalisation

Chapitre 7. Sprint 5

81

Introduction :

Après avoir fait le sprint de la gestion des réactions sur les articles, et la recherche des articles avec filtrage on va passer au cinquième sprint qui consiste à se concentrer sur la gestion des feedbacks publiés par les utilisateurs de notre plateforme, gestion d'ajouter des articles aux liste des favoris, l'envoi d'un email à l'administrateur avant de connecter à notre plateforme, et l'envoi des messages entre les différents utilisateurs et on va exposer les différents diagrammes d'analyse et de conception et ensuite on va présenter la phase de

réalisation.

Phase de préparation :

1.1 But :

Le but de ce cinquième sprint est de gérer les feedbacks, Ajouter d'un feedback, de gérer la liste des favoris et envoyer des emails et des messages.

1.2 Analyse :

Cette partie sera consacrée à l'analyse du cinquième sprint de notre projet, nous allons

présenter les backlog produit de sprint 5, les diagrammes de cas d'utilisation et leurs

descriptions textuelles ainsi que le diagramme séquence système.

1.2.2 Sprint Backlog :

ID Thèmes User Story

1

Gestion des feedbacks
publiés par les utilisateurs
de notre plateforme

En tant qu'utilisateur je veux mettre un
feedback sur le fonctionnement de la
plateforme.

2

Ajouter un Feedback
En tant que consultant je veux mettre
un feedback.

3

Gestion de la liste des
favoris
En tant que consultant je veux ajouter
un article à la liste des favoris.
En tant que consultant je veux
supprimer un article à la liste des
favoris.

4

Envoyer un email
En tant qu'utilisateur je veux envoyer
un email à l'administrateur si il y'a un
problème de connexion.

5

Envoyer un email
En tant qu'utilisateur je veux envoyer
un message privé à un autre utilisateur.

Chapitre 7. Sprint 5

Phase de préparation :

2.1 Modélisation fonctionnelle:

Cette partie est consacrée à la modélisation des diagrammes de cas d'utilisation ainsi que les descriptions textuelles des interactions entre le système et l'utilisateur.

2.1.1. Diagramme de cas d'utilisation global du Sprint 5:

2.1.2. Raffinement de cas d'utilisation « Gérer des feedbacks »:

2.1.3. Raffinement de cas d'utilisation « Ajouter un feedback » :

Chapitre 7. Sprint 5

83

2.1.4. Raffinement de cas d'utilisation « Gérer la liste des favoris » :

2.1.5. Raffinement de cas d'utilisation « Envoyer un message » :

2.1.6. Raffinement de cas d'utilisation « Envoyer un email » :

Chapitre 7. Sprint 5

84

a. Description textuelle de diagramme de cas d'utilisation « Gérer les feedbacks » :

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas

d'utilisation «gérer les feedbacks» :

Sommaire

Titre :

gérer les feedbacks

Auteur :

Administrateur

Description d'enchaînement

Pré-conditions :

L'administrateur est connecté

Post-Conditions :

gérer les feedbacks

Scénario nominal

1. L'administrateur peut afficher la liste des feedbacks.
2. L'administrateur peut supprimer un feedback
3. FIN.

b. Description textuelle de diagramme de cas d'utilisation « L'ajout d'un feedback » :

Chapitre 7. Sprint 5

85

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas d'utilisation «ajouter un feedback» :

Sommaire

Titre :

Ajouter un feedback

Auteur :

Lecteur

Description d'enchaînement

Pré-conditions :

Le Lecteur est connecté

Post-Conditions :

Ajouter un feedback

Scénario nominal

1. Le Lecteur peut envoyer un feedback
2. Le lecteur remplit le formulaire et clique sur envoyer
3. FIN.

c. Description textuelle de diagramme de cas d'utilisation « Gérer la liste des favoris » :

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas

d'utilisation «gérer la liste des favoris» :

Sommaire

Titre :

Gérer la liste des favoris

Auteur :

Lecteur

Description d'enchaînement

Pré-conditions :

Le Lecteur est connecté

Chapitre 7. Sprint 5

86

Post-Conditions :

Gérer la liste des favoris

Scénario nominal

1. Le lecteur consulte l'interface de la liste des articles.
2. Le lecteur choisit l'article qui souhaite à ajouter aux favoris et valider
3. Le système affiche une alerte de succès
4. Le lecteur consulte l'interface de la liste des favoris.
5. Le Lecteur peut supprimer un favori.
8. FIN.

Scénario alternative

1. Article déjà existant :

Ø L'application affiche une alerte indiquant l'erreur.

d. Description textuelle de diagramme de cas d'utilisation « L'envoi d'un message » :

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas

d'utilisation «envoyer un message» :

Sommaire

Titre :

envoyer un message

Auteur :

Lecteur

Description d'enchaînement

Pré-conditions :

L'utilisateur est connecté

Post-Conditions :

envoyer un message

Chapitre 7. Sprint 5

87

Scénario nominal

1. L'utilisateur consulter la boîte de messagerie.
2. L'utilisateur choisit la personne qui souhaite à contacter et saisir son message et valider

e. Description textuelle de diagramme de cas d'utilisation « L'envoi d'un email »

La description textuelle représentée dans le tableau ci-dessous montre le scénario du cas

d'utilisation «envoyer un email» :

Sommaire

Titre :

envoyer un email

Auteur :

Lecteur

Description d'enchaînement

Pré-conditions :

L'utilisateur n'est pas connecté

Post-Conditions :

envoyer un email

Scénario nominal

1. L'utilisateur consulter l'espace d'envoi d'un email.

2. L'utilisateur remplir et envoyer

Conception :

3.4. Diagrammes de séquence détaillée :

3.4.1. Diagramme de séquence de «Gérer la liste des feedbacks» :

Chapitre 7. Sprint 5

88

3.4.2. Diagramme de séquence d «Ajouter un feedback» :

3.4.3. Diagramme de séquence de «Gérer la liste des favoris et ajouter un article à la liste » :

Chapitre 7. Sprint 5

89

3.4.4. Diagramme de séquence de «Envoyer un message» :

Réalisation :

4.1. Gestion des feedbacks:

Chapitre 7. Sprint 5

90

4.2. Ajout d'un feedbacks :

4.3. Gestion de la liste des favoris :

Chapitre 7. Sprint 5

91

4.4. L'ajout d'un article aux favoris :

Chapitre 7. Sprint 5

92

4.5. L'envoi d'un message :

4.6. L'envoi d'un email :

Chapitre 7. Sprint 5

93

94

95

Agence digitale, conception web, solutions d'hébergement et cloud (tanitweb.com)
Quelle presse demain ? – La Tribune de Montaigne (home.blog)
https://www.memoireonline.com/10/19/11057/m_Mise-en-place-d-un-portail-de-gestion-des-recrutements-evaluations-et-suivi-des-cursus-des-etudia6.html
AGILITE EN PRATIQUE : SCRUM (utc.fr)