Remerciements

Je tiens à exprimer toute mes sincères remerciements avec un grand plaisir et un grand respect à mon encadrant Monsieur ZIED ELLEUCH, ses conseils, sa disponibilité et ses encouragements m'ont permis de réaliser ce travail dans les meilleures conditions.

J'adresse aussi mes reconnaissances à tous les enseignants et au corps administratif de notre école.

Je voudrais aussi exprimer ma gratitude envers tous ceux qui m'ont accordé leur soutien, tant par leur gentillesse que par leur dévouement, en particulier Rassas Achraf qui m'a souvent aidé à résoudre des difficultés techniques.

Je ne peux nommer ici toutes les personnes qui de près ou de loin m'ont aidé et encouragé mais je les en remercie vivement.

Enfin je tiens à dire combien le soutien quotidien de ma famille a été important tout au long de cette période, je leur dois beaucoup.

Merci à vous

Yosra Souissi

Sommaire

\mathbf{R}	emer	${f c}_{f c}$ iements	i
In	trod	uction générale	1
1	Ana	alyse et spécification des besoins	2
	Intr	$ oduction \ldots \ldots \ldots \ldots \ldots \ldots $	3
	1.1	Présentation de la sociète	3
	1.2	Etude de l'existant et Critique de l'existant	3
	1.3	Solution proposée	5
	1.4	Les besoins fonctionnels et non fonctionnels	5
		1.4.1 Les besoins fonctionnels	5
		1.4.2 Les besoins non fonctionnels	6
		1.4.3 Les acteurs	6
	1.5	Diagramme de cas d'utilisation :	7
		1.5.1 Raffinnement des cas d'utilisation	9
		1.5.2 Tableau des Scénario	10
	1.6	Diagrammes de séquences	11
	1.7	Diagrammes de séquences systèmes	14
	1.8	Méthodologie utilisée	16
	Con	clusion	17
2	Etu	ide conceptuelle	18
	Intr	oduction	19
	2.1	Diagramme de classe	19
	Con	clusion	21
3	Réa	alisation	22
	Intr	oduction	23
	3.1	Technologie et environnements	23

	3.1.1	Architecture du système	. 23
	3.1.2	Technologie utilisée	. 24
3.2	Envir	onnement de travail	. 25
	3.2.1	Environnement logiciel	. 25
3.3	Interf	aces de l'application	. 26
Cor			. 29
Concl	usion g	générale	30
Biblio	graphi	e	31

Table des figures

1.1	Page d'accueil du site books-stream	4
1.2	Une page instagram spécialisée à la vente des livres deuxième mains	4
1.3	Diagramme de cas d'utilisation général	8
1.4	Raffinnement du cas d'utilisation « demande de réservation d'un livre »	Ö
1.5	Raffinnement du cas d'utilisation « gérer les livres »	Ö
1.6	Diagramme de séquence «authentification d'un adhérant»	12
1.7	Diagramme de séquence «inscription d'un visiteur»	13
1.8	Diagramme de séquence «ajouter livre»	14
1.9	Diagramme de séquence système « inscription »	15
1.10	Diagramme de séquence système « ajouter livre »	15
1.11	Modèle de cycle en v	16
2.1	Diagramme de classe	20
3.1	Architecture MVC	23
3.2	Interface page d'accueil	27
3.3	Interface d'inscription	27
3.4	Formulaire d'ajout des livres pour la réservation	28
3.5	Espace d'authentification admin	29

Liste des tableaux

1.1	Tableau récapitulatif de la présentation de notre sociéte d'accueil	3
1.2	Tableau identifiant des rôles des acteurs	7
1.3	Description du cas d'utlilisation « S'inscrire »	10
1.4	Description du cas d'utilisation « passer une réservation »	10
1.5	Description de cas d'utilisation « gérer les réservations » $\dots \dots \dots$	11
2.1	Dictionnaire des données	21
3.1	Les avantages des de Ruby On Rails	24

Introduction générale

Jours après jours, fasciné par l'avancement et le progrès accéléré de l'informatique, le monde cherche à inclure cette technologie de tous les domaines en vis de répondre aux besoins et attentes des individus et tendre à améliorer le niveau de vie. Ces changements posent, naturellement, un grand challenge pour tous les domaines ce qui pousse l'homme à la concurrence. L'un des domaines visés par ce phénoméne est la lecture. Cet excellent moyen d'acquisition de savoir, de connaissance, a perdu son rôle d'accès à la culture et à la citoyenneté. Aujourd'hui, la majorité préfèrent de loin l'image au livre. Tout au plus, ils préfèrent la lecture numérique, c'est-a-dire lire directement sur l'écran d'ordinateur, de Smartphones, sur des tablettes qu'ils trouvent plus facile.

Donc afin d'inciter les gens nous allons élaborer un projet de fin d'année au sein de la société « Devagnos » afin de créer un site web de lecture en ligne qui présente un ensemble des livres à lires et à télécharger, ainsi qu'il garantit l'échange des oeuvres entre les lecteurs. L'objectif de cette application est de proposer une solution simple pour toutes personnes qui apprécit la lecture

De ce fait, les différentes phases sont incorporées dans les différentes activités, qui se présentent en quatre chapitres : Le premier chapitre nommé «Analyse et spécification des besoins» qui consiste à décrire une étude et critique de l'existant aboutissant ainsi à la proposition des solutions informatiques. Le second chapitre intitulé «Etude Conceptuelle» qui consiste à analyser les besoins décrits dans le chapitre précédent. Il s'intéresse sur l'aspect conceptuel de projet en présentant de différents scénarios par l'intermédiaire de langage de conception et de modélisation UML. Cette phase est importante car elle se présente comme une phase primordiale pour la réussite de programmation. Le troisième chapitre présentera «Le développement et la réalisation» évoquera l'architecture générale de l'environnement de développement dans lequel le projet a été réalisé, les étapes de la réalisation de l'application ainsi qu'un ensemble des interfaces de projet avec des simples descriptions.

Finalement nous clôturons le rapport par une conclusion générale qui présente le bilan de ce projet et les éventuelles perspectives.

Chapitre 1

Analyse et spécification des besoins

Sommaire

Intr	oduction
1.1	Présentation de la sociète
1.2	Etude de l'existant et Critique de l'existant
1.3	Solution proposée
1.4	Les besoins fonctionnels et non fonctionnels
	1.4.1 Les besoins fonctionnels
	1.4.2 Les besoins non fonctionnels 6
	1.4.3 Les acteurs
1.5	Diagramme de cas d'utilisation :
	1.5.1 Raffinnement des cas d'utilisation
	1.5.2 Tableau des Scénario
1.6	Diagrammes de séquences
1.7	Diagrammes de séquences systèmes
1.8	Méthodologie utilisée
Con	clusion

Introduction

Ce chapitre vise en première partie la présentation de l'entreprise d'accueil au sein de laquelle nous avons effectuer notre stage de fin d'années, ensuite la deuxième partie se concentre à étudier le fonctionnement du système existant. de ce produit en spécifiants les besoins fonctionnels et non fonctionnels. L'étude réalisé va souligner quelques problèmes concernant la gestion des emprunts, la lecture gratuite et la réservation des bouquints, pour cela nous allons proposer une solution totalement informatisée pour améliorer le contact entre les personnes doués de la lecture eux mêmes et l'administrateur.

1.1 Présentation de la sociète

Devagnos est une agence de communication numérique incontournable basée dans l'une des principales villes de Tunisie, Monastir, elle est aussi une société de développement et de conseil en logiciels alimentée par l'élite de développeurs talentueux dans la région, ce qui lui permet d'offrir des solutions complètes aux clients avec le meilleur des deux mondes. (Solutions spécifiques, Conception, Déploiement, CMS, E-Commerce, Intranet, ERP, CRM, Design, Gestion communautaire, Evénements, etc). Elle est spécialisé dans les solutions basées sur le web

Raison Sociale	DEVAGNOS	
Adresse	603 Immeuble Benzarti Monastir 5000	
Télephone	54 173 773	
Effectif moyen	environ 21 développeurs	
Régime de travail	$40 \mathrm{H/Semaine}$	
Domaine d'activité	société de développement et de conseil en logiciels	

TABLE 1.1 – Tableau récapitulatif de la présentation de notre sociéte d'accueil

1.2 Etude de l'existant et Critique de l'existant

Etude de l'existant :

La lecture est une source de divertissement, tout comme les spectacles, les jeux et le sport, elle nous procure un plaisir en nous détournant du réel où l'on vit, favorisant ainsi l'oubli des soucis et du stress du quotidien. Mais vu la pression du temps , il sera presque impossible pour l'homme d'aller au bibliothèque pour acheter un livre , alors comment résoudre le problème d'accessibilité à la bibliothèque?

- Description du processus de gestion des livres existant :

Pour un utilisateur doué pour la lecture, il n'est possible de pratiquer ce loisir que soit par l'achat des livres (qui est une solution bien couteuse à long terme), soit par l'échange entre camarades ou collègues ou membres de familles et soit en allant à la bilbliothéque pour emprunter un livre . Bien qu'il existe des actions d'échange non gratuit des livres sur les réseaux sociaux mais ça ne corrige pas vraiment les problèmes rencontrés.

- Exemple d'un site existant :

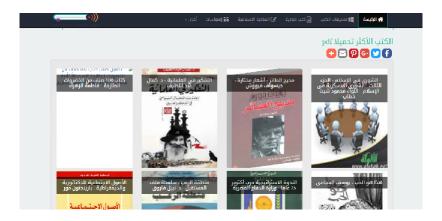


FIGURE 1.1 – Page d'accueil du site books-stream

Ce site est une vaste bilbliothèque qui permet au utlisateurs de lire en ligne et télécharger des livres. Le seul soucis de ce site qu'il est en arabe et qu'il ne contient pas des livres en français ou en anglais.

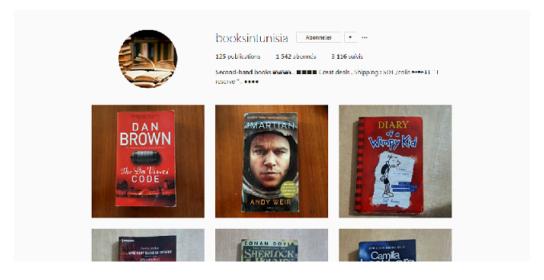


FIGURE 1.2 – Une page instagram spécialisée à la vente des livres deuxième mains

C'est une bonne idée pour s'investir des ancienns livres mais le premier inconvénient c'est

quelle est une solution payante de plus elle oblige l'utilisateur d'avoir un compte instagram ou facebook .

Critique de l'existant :

L'objectif de critique de l'existant est d'établir un diagnostic précis sur les lacunes existantes. Cela permet de trouver les solutions adéquates à ces problèmes. Parmi les problèmes rencontrés nous pouvons citer : Perte du temps lors des recherches des livres qu'on souhaitent lire, perte d'argent, la non-réutilistation des livres qu'on a déjà acheté, une mauvaise gestion des emprunts, et surtout l'absence d'espaces privés pour les gens passionnés de la lecture et l'écriture.

1.3 Solution proposée

L'objectif de notre projet est de satisfaire aux besoins des lecteurs et d'éviter les défaillances déjà évoquées. La solution que nous proposons sera la conception et le développement d'une plateforme de lecture et échange des bouquint en ligne, un site facile à utiliser pour l'administrateur ainsi que pour les utilisateurs et qui améliorera l'interaction entre eux.

L'apport de notre solution proposée est expliqué par :

Le gain du temps pour l'utlisateur : ce dernier n'est pas obligé de se déplacer pour acheter son livre.

Gain de temps et d'argent pour l'utilisateur.

Encourager les lecteurs pour essayer de partager ses propres publications.

Faciliter la lecture pour les gens intéressés.

1.4 Les besoins fonctionnels et non fonctionnels

1.4.1 Les besoins fonctionnels

Ces besoins fonctionnelles sont directement liés aux taches à réaliser ils doivent être transparentes le plus possible face aux utilisateurs. Ils doivent supporter les fonctionnalités techniques suivantes :

- Espace lecteur :

S'inscrire

Consulter le site (consulter les livres, les réservations et les publications des autres lecteurs)

Modifier son profil

Gérer les demandes de réservations proposés

Se communiquer avec les lecteurs et parfois avec l'administrateur

Espace administrateur : c'est lui qui est responsable au publications des livres à lire
 Gérer les livres à lire en ligne

Gérer les utilisateurs (parfois il sera obligé de bloquer quelques utilisateurs non sérieux)

Traiter les demandes de réservations envoyées

1.4.2 Les besoins non fonctionnels

Les besoins non fonctionnels sont les contraintes qui peuvent empêcher notre plateforme de fonctionner correctement et d'une manière performante et efficace. Ce sont donc des besoins de différents types liés à l'environnement (exigence en matière de performances, facilité de maintenance...)

Convivialité: Le système doit être facile à utiliser, en effet les interfaces utilisateurs doivent être conviviales c'est-à-dire simple et adaptée à l'utilisateur.

Performance : Le système doit être performant c'est-à-dire qu'il doit répondre aux fonctionnalités de base d'une façon optimale et dans un temps minimal.

Fiabilité: Un système avec un risque d'erreur et de panne minimal.

Sécurité : Accès personnalisés, connexions sécurisée.

Portabilité: Utilisable avec plusieurs systèmes d'exploitation.

L'ergonomie: Les interfaces doivent être simples, claires et professionnelles.

1.4.3 Les acteurs

UML n'emploie pas le terme d'utilisateur mais le terme d'acteur. Les acteurs d'un système sont les entités externes à ce système qui interagissent (saisie de données, réception d'information, ...) avec lui. Les acteurs sont donc à l'extérieur du système et dialogue avec lui. Ces acteurs permettant de cerner l'interface que le système offrir à son environnement. Un tableau récapitulant la liste des acteurs qui interagissent avec le système ainsi l'explication du rôle de chacun entre eux :

Nom acteurs	Rôle
Administrateur	Il chargée de :
	-La gestion des livres.
	-La gestion des demandes.
	-La gestion des réservations.
	-La gestion des comptes des
	${ m adh\'{e}rants}$.
	-La gestion des commentaires
	publiés.
Adhérant (utilisateur inscrit)	Il a le droit de :
	-gérer son profil.
	-modifier sa liste des livres à
	$ m \acute{e}chang\acute{e}.$
	-contacter l'administrateur
	contacter les adhérants inscrits à
	la plateforme.
Visiteur	Il peut consulter la listes des
	livres en lecture ainsi que la liste
	des auteurs

Table 1.2 – Tableau identifiant des rôles des acteurs

1.5 Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation sont des diagrammes Unified Modeling Language (UML) utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel.

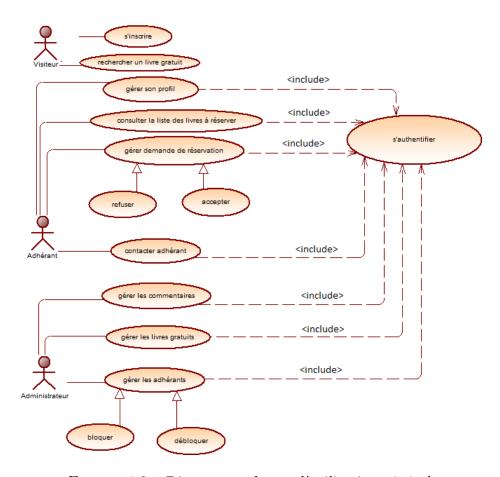


FIGURE 1.3 – Diagramme de cas d'utilisation général

1.5.1 Raffinnement des cas d'utilisation

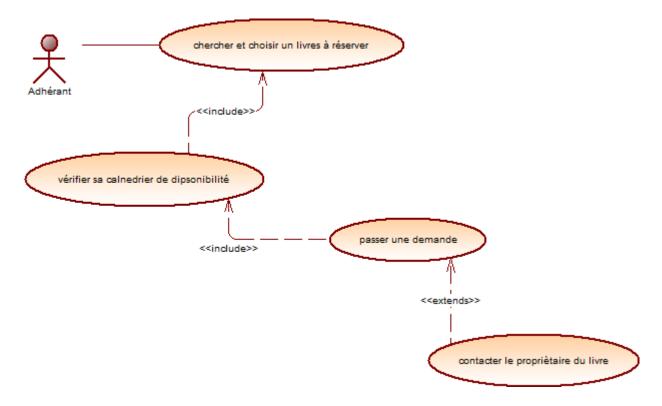


FIGURE 1.4 – Raffinnement du cas d'utilisation « demande de réservation d'un livre »

Pour réserver un livre l'adhérant doit vérifier la calendrier de sa disponibilité puis il va passer la demande avec une date précise, aprés l'envoi de la demande il peut contacter le propriètaire de ce livre pour avoir plus d'information.

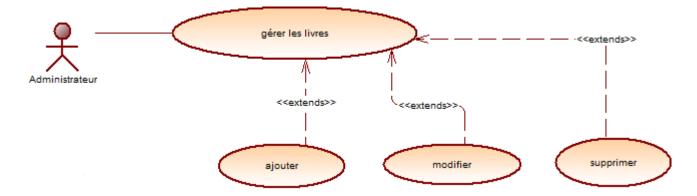


Figure 1.5 – Raffinnement du cas d'utilisation « gérer les livres »

Pour gérer les livres, l'administrateur peut ajouter des nouveaux livres comme il peut les modifier et les supprimer.

1.5.2 Tableau des Scénario

- Le tableau ci dessous montre le raffinement du cas d'utilisation « s'inscrire » pour le visiteur en traitant tous les cas possibles.

Table 1.3 – Description du cas d'utilisation « S'inscrire »

Cas d'utilisation	S'inscrire
Acteur	Un simple visiteur
Pré-condition(s)	Le visiteur souhaite s'inscrire
Post-condition(s)	Remplir tous les champs correctement
Scénario principal	Pour réussir l'inscription, le visiteur doit remplir
	correctement les informations demandées depuis le
	formulaire
Exception	En cas de problème, un message d'erreur s'affichera

- Ce tableau montre une briève description du cas d'utilisation « passer une réservation »

Table 1.4 – Description du cas d'utilisation « passer une réservation »

Cas d'utilisation	passer une réservation	
Acteur	Adhérant (un utilisateur connecté)	
Pré-condition(s)	Le visiteur souhaite passer une réservation d'échange	
	pour un livre qui lui a plait	
Post-condition(s)	Il vérifie les dates dans lesquelles ce livre sera	
	disponible puis il envoie une demande au propriétaire	
	du livre	
Scénario principal	L'adhérant consulte la liste des livres disponible à	
	l'échange, il choisit une date précise puis il envoie la	
	${\it demande}.$	
Exception	En cas de problème, un message apparaitra dans la	
	messagerie de l'adhérant	

- Description de cas d'utilisation « gérer les réservations »

Table 1.5 – Description de cas d'utilisation « gérer les réservations »

Cas d'utilisation	gérer les réservations	
Acteur	$\operatorname{Administrateur}$	
Pré-condition(s)	L'administrateur gére tous, il est donc responsable au	
	réservations des adhérants par acceptation ou refus.	
Post-condition(s)	Il vérifie les dates dans lesquelles ce livre sera	
	disponible puis il envoie cette demande au propriétaire	
	du livre.	
Scénario principal	L'administrateur consulte la liste des demandes de	
	réservations et les traitent par ordre de priorité puis	
	décide celles qui vont étre envoyés à la personne	
	concernée et les autres qui sont réfusés.	

1.6 Diagrammes de séquences

Définition : Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages.

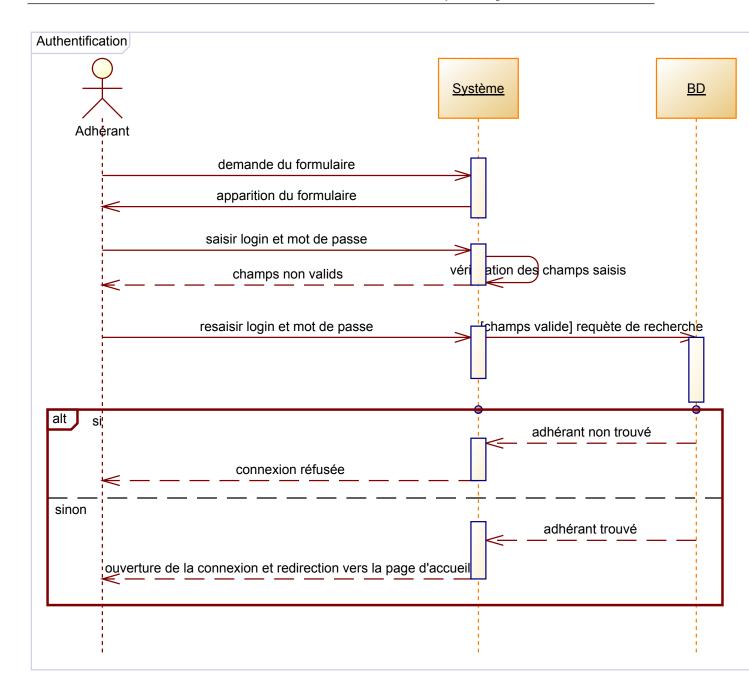


FIGURE 1.6 – Diagramme de séquence «authentification d'un adhérant»

Ce diagramme de séquence montre l'authentification de l'adhérant par étapes. L'adhérant saisie son login et son mot de passe Le système donc va vérifier les champs saisis : si un champ est vide un message d'erreur va être affiché Le système va vérifier la validité des données (login et mot de passe valide) : si les données ne sont pas conformes avec ceux qui sont stockés dans la base de données alors un message d'erreur va être affiché sinon un message de validation s'affiche.

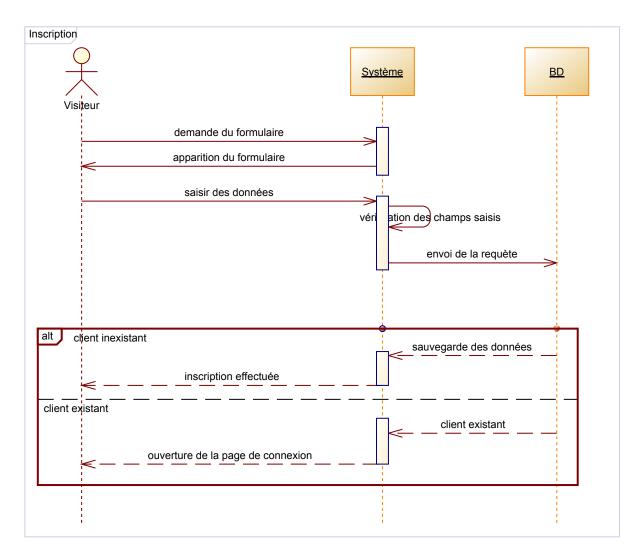


FIGURE 1.7 – Diagramme de séquence «inscription d'un visiteur»

Le visiteur remplit le formulaire d'inscription Le système donc va vérifier les champs saisis, si tous les champs sont remplis il va donc envoyer la requête. Le système va vérifier la validité des données : si le visiteur n'existe pas alors un message de validation va être affiché « votre inscription a été bien effectué » sinon un message d'erreur s'affiche « vous êtes déjà inscrit, connectez-vous »

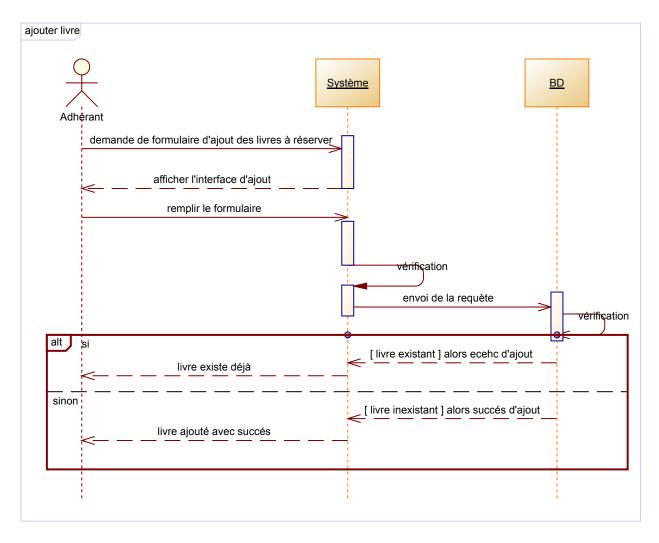


FIGURE 1.8 – Diagramme de séquence «ajouter livre»

Après avoir ouvrir l'interface ajouter, un adhérant doit remplir le formulaire Le système donc va vérifier les champs saisis, si tous les champs sont remplis il va donc envoyer la requête. Le système va vérifier la validité des données : si le livre existe alors un message d'erreur va être affiché « livreicle existe déjà » sinon un message de validation s'affiche « livre ajouté avec succès »

1.7 Diagrammes de séquences systèmes

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

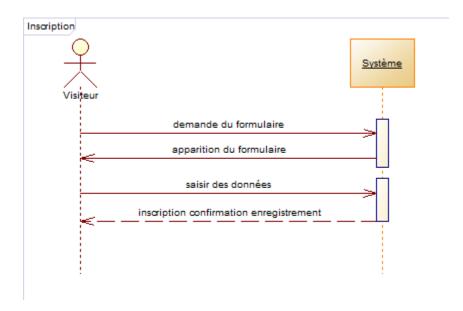


Figure 1.9 – Diagramme de séquence système « inscription »

Ce diagramme montre l'intéraction de l'acteur visiteur avec le système pour effectuer l'opération de l'authentification .

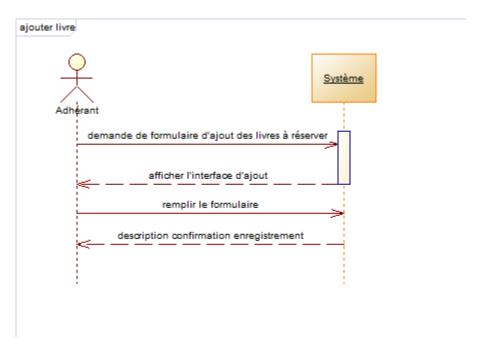


FIGURE 1.10 – Diagramme de séquence système « ajouter livre »

Ce diagramme montre l'intéraction de l'acteur adhérant avec le système pour effectuer l'opération de l'ajout d'un livre .

1.8 Méthodologie utilisée

1.8.0.1 Modéle de cycle de vie

Le développement d'un projet se fait suivant un cycle appelé le cycle de vie du logiciel. Le cycle de vie d'un logiciel est une modélisation conventionnelle de la succession d'étape qui aboutit à la mise en œuvre d'un logiciel. On a choisi le modèle de cycle de vie incrémental vu qu'il diminue les risques d'échec en divisant le projet en incréments. A chaque incrément ou itération il décrit une succession d'étapes qui sont représentées par six étapes fondamentales (Spécifications, Conception générale, Conception détaillée, Codage, Intégration, Mise en production et Maintenance) qui se terminent à des dates précises par la production de documents ou logiciels. Les résultats de l'étape sont examinés attentivement avant de passer à l'étape suivante.

1. Modéle de cycle en V : C'est un modèle conceptuel de gestion de projet imaginé suite au problème de réactivité du modèle en cascade. Il permet, en cas d'anomalie, de limiter un retour aux étapes précédentes. Les phases de la partie montante doivent renvoyer de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés, afin d'améliorer le logiciel.

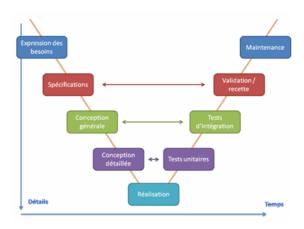


FIGURE 1.11 – Modèle de cycle en v

2. Avantages et Inconvénients

Avantages:

 Validations intermédiaires
 Limitent les risques du cycle en cascade par la préparation de validation à chaque étape. L'obligation de concevoir les jeux de tests et leurs résultats oblige à une réflexion et à des retours sur la description en cours.

- Modèle très utilisé pour les grands projets.

Inconvénients

- Un modèle toujours séquentiel (linéaire)
 Prédominance de la documentation : validation tardive du système par le client .
 Pas de résultats intermédiaires dont on peut discuter avec le client.
- limité aux problèmes sans zones d'ombre Idéal quand les besoins sont bien connus et quand l'analyse et la conception sont claires.

Conclusion

Dans ce chapitre, nous avons présenté notre solution informatique afin de critiquer l'existant. Ainsi nous aurons terminé notre étude préalable pour entamer à l'étude conceptuelle dans le chapitre suivant.

Chapitre 2

Etude conceptuelle

Sommaire

Introduction	19
2.1 Diagramme de classe	19
Conclusion	21

Introduction

Nous allons entamer dans ce chapitre la proposition conceptuelle qui répond aux besoins analysés dans le chapitre précédent. L'analyse conceptuelle est la phase la plus importante qui précède le développement de l'application. Dans ce chapitre nous présentons une conception détaillée de notre application. En premier lieu nous commençons par la présentation de l'architecture générale du projet. En deuxième lieu, nous référons sur une vue statique à travers le diagramme de classe, les diagrammes d'intéractions et les diagrammes de package.

2.1 Diagramme de classe

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. C'est un diagramme entités-associations décrivant les différentes classes, leur structure et les associations statiques les unissant.

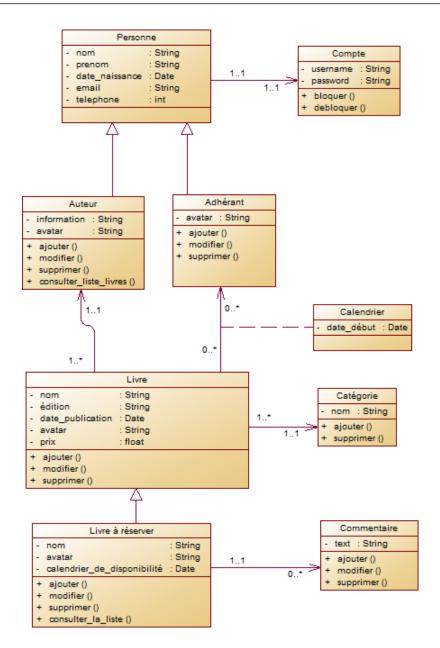


FIGURE 2.1 – Diagramme de classe

Table 2.1 – Dictionnaire des données

Code champs	Type	Description
nom_p	Chaine de caractère	nom du personne
prenom	Chaine de caractère	prenom du personne
${ m date_naissance}$	date	date de naissance du personne
email	Chaine de caractère	email du personne
telephone	Chaine de caractère	telephone du personne
username	Chaine de caractère	login
password	Chaine de caractère	mot de passe
information	text	information sur l'ateur
avatar_auteur	Chaine de caractère	image de l'auteur (s'il existe)
avatar_adhérant	Chaine de caractère	image de l'adhérant (s'il existe)
${ m date_d\'ebut}$	date	date de la réservation
nom_livre	Chaine de caractère	nom du livre
édition	Chaine de caractère	édition du livre
$date_publication$	date	nom du personne
avatar_livre	Chaine de caractère	image du livre
prix	double	prix du livre
nom_catégorie	Chaine de caractère	nom de la catégorie
nom_livre_réserver	Chaine de caractère	nom du livre à réserver
avatar_livre_réservé	Chaine de caractère	image du livre à réserver
calendrier de disponibilité	date	calendrier de disponiblité d'un livre à réserver

Conclusion

Dans ce chapitre, nous avons réalisé l'étude conceptuelle pour faciliter la compréhension de l'environnement de notre travail. Ce chapitre est vu comme le plus important puisqu'il nous rapproche de la réalisation. On décrit dans le chapitre suivant, l'environnement de développement de notre application.

Chapitre 3

Réalisation

Sommaire

Intr	Introduction					
3.1	Technologie et environnements	23				
	3.1.1 Architecture du système	23				
	3.1.2 Technologie utilisée	24				
3.2	Environnement de travail	25				
	3.2.1 Environnement logiciel	25				
3.3	Interfaces de l'application	26				
Con	clusion	2 9				

Introduction

Après avoir achevé la phase conception du système, nous entamons la phase de développement du site web. Au cours de cette phase, nous présentons tout d'abord l'architecture du système puis par la suite nous allons parler de l'environnement du travail ensuite l'architecture technique de l'application et nous clôturons cette phase par des captures d'interfaces de l'application.

3.1 Technologie et environnements

3.1.1 Architecture du système

L'architecture de système est une discipline multiforme et transversale qui traite de "comment concevoir et construire un système". Son objectif est de maîtriser l'évolution du système d'information et la structure des systèmes informatiques à travers l'anticipation des problèmes de frontières, de faisabilité et de performance tout au long du processus logiciel.

- Architecture MVC (Modèle/Vue/Contrôleur)

L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface.

Dans l'architecture MVC, les rôles des trois entités sont les suivants :

modèle : données (accès et mise à jour)

vue : interface utilisateur (entrées et sorties)

contrôleur : gestion des événements et synchronisation

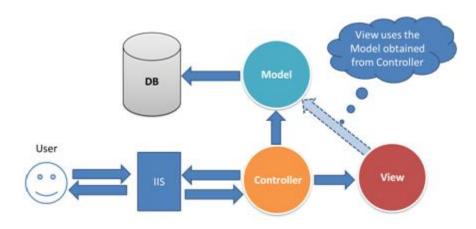


FIGURE 3.1 – Architecture MVC

3.1.2 Technologie utilisée

3.1.2.1 Ruby On Rails

Ruby on Rails (RoR)¹ est un framework d'applications Web écrit en Ruby. Ce framework a été écrit par David Heinemeier Hansson.s. RoR permet à un programmeur de développer des applications Web beaucoup plus rapidement en minimisant les étapes de développement en gagnant beaucoup du temps eten assurant la performance du programmation.

Le langage RUBY

Ruby², comme Perl et Python, est un langage de «script» à typage dynamique :

Ruby est un langage open-source dynamique qui met l'accent sur la simplicité et la productivité. Sa syntaxe élégante en facilite la lecture et l'écriture.

Il se distingue pas sa facilité d'implémentation

Sa philosophie est DRY:

- Don't Repeat Yourself
- Convention over configuration
- REST: Representational State Transfer
- CRUD

Le Framework Rails

Rails est un framework pour les applications Web, écrit avec et grâce à Ruby et il se caractérise par sa souplesse d'intégration et son dynamisme. Rails s'adapte également avec rapidité aux développements les plus récents de la technologie internet.

- Pourquoi Ruby On Rails?

Table 3.1 – Les avantages des de Ruby On Rails

Développement rapide	l'implémentation de nouvelles fonctionnalités est
	facilitée, votre projet peut évoluer en continu
Evolutivité	fonctionnalités sur mesure, interfaces riches,
	interactivité
Créativité	l'architecture même du framework "contraint" les
	développeurs dans leur manière de coder.
Performance	les fichiers Javascript et CSS sont minifiés et
	compressés, ce qui permet à votre site d'être plus
	rapide

^{1.} http://rubyonrails.org/

^{2.} https://www.ruby-lang.org

3.2 Environnement de travail

3.2.1 Environnement logiciel

3.2.1.1 Outils de dévèloppement

- Visual Code : est un éditeur de code source pour Windows, OSX et Linux, qui est gratuit et qui permet d'éditer et de débugger votre code, sans avoir besoin de la version payante de Visual Studio. Il intègre un gestionnaire de paquets et de dépôts et peut communiquer avec Git et des outils de diff. De plus si ça vous dit, il est même possible



de développer des plugins pour VS Code.

3.2.1.2 Outils de conception

- StarUML est un logiciel de modélisation UML (Unified Modeling Language) open



3.2.1.3 Outils de traitement de texte

LyX³ est un logiciel libre sous licence GNU GPL pour la création de documents LaTeX. Ce logiciel a été conçu pour que l'utilisateur n'ait pas à sa charge la mise en page, et qu'il puisse se concentrer sur le contenu du texte et sur la structure du document. Les concepteurs de LyX ont développé le logiciel afin qu'il obéisse à la règle WYSIWYM selon laquelle ce que



vous voyez (à l'écran) est ce que vous voulez dire.

3.2.1.4 Langage de dévèloppement

- HTML5 (HyperText Markup Language 5) est la dernière révision majeure d'HTML. Cette version est en développement en 2013. HTML5 spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM: HTML5 et XHTML5. Le language comprend également une couche application avec de nombreuses API, ainsi qu'un algorithme afin de pouvoir traiter les documents à la syntaxe non conforme. Le travail a été repris par

^{3.} http://www.lyx.org/

le W3C en mars 2007 après avoir été lancé par le WHATWG. Les deux organisations travaillent en parallèle sur le même document afin de maintenir une version unique de



la technologie.

 JavaScript : c'est un langage de script incorporé dans un document HTML. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage



HTML.

— SQLite : c'est un système de base de données qui a la particularité de fonctionner sans serveur, on dit aussi "standalone" ou "base de données embarquée". On peut l'utiliser avec beaucoup de langages : PHP, Python, C# (.NET), Java, C/C++, Delphi, Ruby



– Bootstrap : c'est un framework CSS qui embarque également des composants HTML et JavaScript. Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style pour les boutons, les formulaires, la navigation. Il permet ainsi de concevoir un site web rapidement et avec peu de lignes



de code ajoutées.

3.3 Interfaces de l'application

- Espace adhérant

Cette partie met l'accent sur la présentation des interfaces des adhérant tout en commentant ces derniers. Notre page d'accueil offre aux visiteurs du plateforme un ensemble de services : Consulter les catégories, les livres, faire l'inscription, effectuer l'authentification, consulter les livres ainsi contacter l'administrateur.

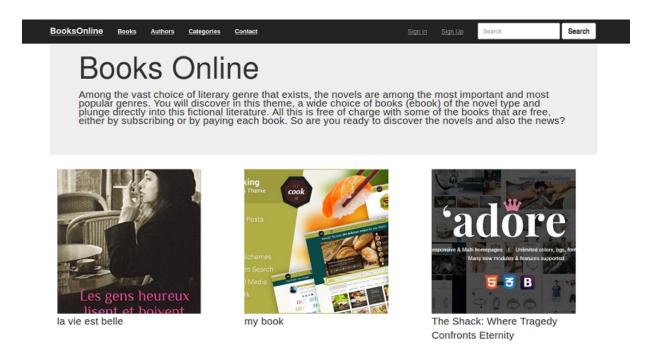


FIGURE 3.2 – Interface page d'accueil

Si le visiteur de la plateforme n'est pas encore inscrit, il ne peut pas accéder aux services offerts. Alors il doit s'inscrire avant de pouvoir bénéficier des fonctionnalités à l'aide de formulaire d'inscription.

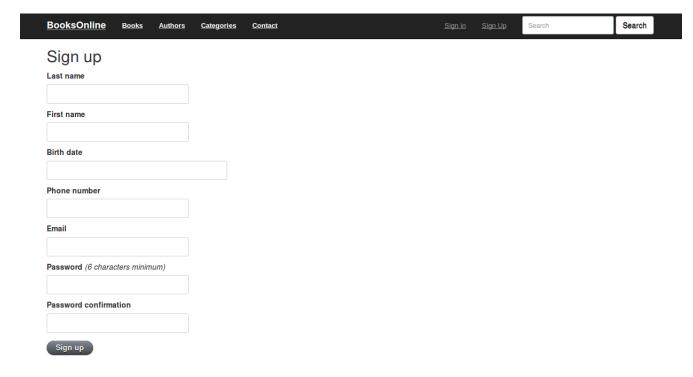


FIGURE 3.3 – Interface d'inscription

Apres l'inscription chaque adhérant peut ajouter les livres qu'il posséde et qu'il voudrait les mettre à la réservation. Il doit alors remplir soigneusement ce formulaire.

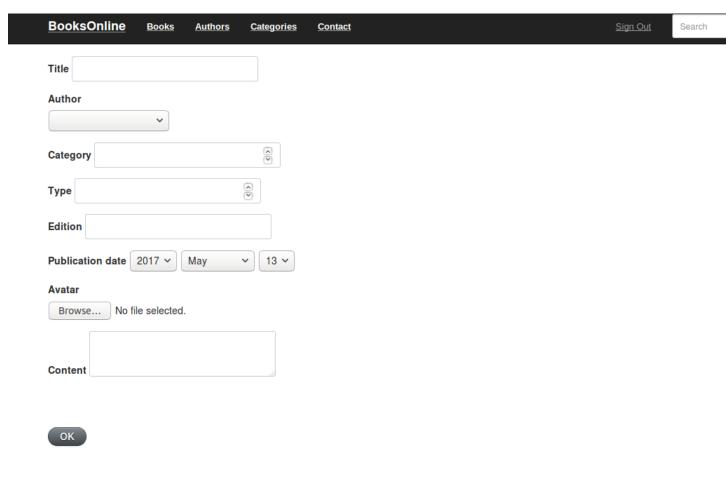


FIGURE 3.4 – Formulaire d'ajout des livres pour la réservation

- Espace administrateur

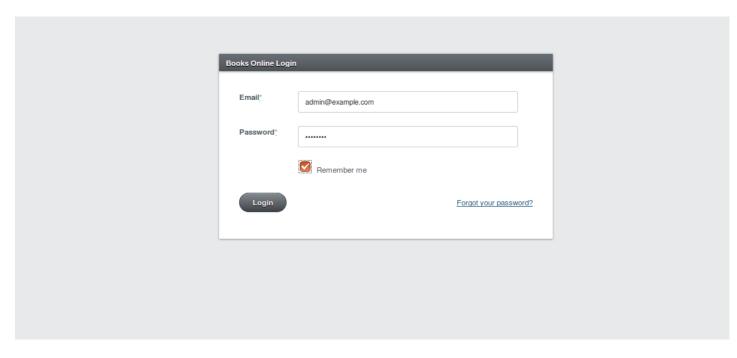


FIGURE 3.5 – Espace d'authentification admin

Il s'agit d'une interface permettent à l'administrateur de l'application de gérer ces fonctionnalités. A partir de la page connexion, il peut s'identifier en tant qu'administrateur.

Conclusion

Dans ce chapitre, nous avons dans un premier temps présenté notre environnement de travail puis nous avons exposé quelques les interfaces du site.

Conclusion générale

En guise de conclusion, on peut mentionner que ce projet de fin d'année nous a permis de maîtriser des outils de conception et de réalisation à savoir des logiciels que nous n'avons pas auparavant eu la chance de les connaître. Notre projet consiste à analyser, concevoir et développer une plateforme d'ouvrage facilitant la lecteurs pour ceux et celles qui sont passionnés par la litérrature ainsi qu'elle est un espace virtuel ou on peut avoir de bonnes et nouvelles amitiés. On a également, tout au long de ce rapport, exposée les différentes étapes du cycle de vie de développement de notre site : nous avons exposée l'étude de l'existant qui nous a mené à l'analyse et la spécification des besoins, Nous avons aussi présenté la conception de notre projet qui a été couronnée par la mise en œuvre de notre plateforme. Nous souhaitons que notre travail ait accompli ses objectifs, mais, comme tout autre travail il ne peut prétendre la perfection. Finalement, nous tenons à signaler que cette application est évolutive et extensible et peut être toujours enrichi et peut être développé. Dans ce terme, nous avons fixé comme prespectives l'ajout d'une application mobile associée avec cette plateforme qui peut faciliter encore et encore la communication entres les lecteurs.

Bibliographie

- [1] A. Bakharia. Ruby on Rails Power!: The Comprehensive Guide. Course Technology, 2006.
- [2] F. Benhamou and A. Colmerauer, editors. Constraint Logic programming, Selected Research. MIT Press, 1993.
- [3] A. Brillant. Ruby: Les fondamentaux du langage Mise en oeuvre avec Ruby on Rails. Ressources Informatiques. Informatique technique. Editions ENI, 2008.
- [4] M. Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language. Object Technology Series. Addison-Wesley, 2004.
- [5] M. Hartl. Ruby on Rails Tutorial: Learn Web Development with Rails. Addison-Wesley Professional Ruby Series. Pearson Education, 2015.
- [6] T. Muntean. Puces très performantes. Terres du futur, Les Editions UNESCO. Hatier, Paris, 1993.
- [7] P. Roques. UML 2 par la pratique : études de cas et exercices corrigés. Eyrolles, 2004.

Table des matières

\mathbf{R}_{0}	Remerciements Introduction générale								
In									
1	Analyse et spécification des besoins Introduction								
	1.1	Présentation de la sociète	3						
	1.2	Etude de l'existant et Critique de l'existant	3						
	1.3	Solution proposée	5						
	1.4	Les besoins fonctionnels et non fonctionnels	5						
		1.4.1 Les besoins fonctionnels	5						
		1.4.2 Les besoins non fonctionnels	6						
		1.4.3 Les acteurs	6						
	1.5	Diagramme de cas d'utilisation :	7						
		1.5.1 Raffinnement des cas d'utilisation	8						
		1.5.2 Tableau des Scénario	10						
	1.6	Diagrammes de séquences	11						
	1.7	Diagrammes de séquences systèmes							
	1.8								
		1.8.0.1 Modéle de cycle de vie	16						
	Con	nclusion	17						
2	Etude conceptuelle								
	Introduction								
	2.1 Diagramme de classe								
	Conclusion								
3	Réalisation								
	Introduction								

	3.1	Techn	ologie et	environnements		23				
		3.1.1	Architec	cture du système		23				
		3.1.2	ogie utilisée		24					
			3.1.2.1	Ruby On Rails		24				
	3.2	Enviro	$\mathbf{onnement}$	de travail		25				
		3.2.1	Environ	nement logiciel		25				
			3.2.1.1	Outils de dévèloppement		25				
			3.2.1.2	Outils de conception		25				
			3.2.1.3	Outils de traitement de texte		25				
			3.2.1.4	Langage de dévèloppement		25				
	3.3	Interfa	aces de l'a	application		26				
	Con	clusion				29				
Co	nclu	sion g	énérale			30				
Bibliographie										