



Rapport du Projet de Fin d'année

Elaboré par :

**Ayedi Emna
Gharbi Chaima**

Encadré par :

Mme. Imene LAHYENI

Jugé par :

Mme. Mouna Torjmen

Année scolaire :2019-2020

REMERCIEMENT :

Nos plus vifs remerciements vont à notre encadreur

*Mdm.Lahiani.Imene pour sa compréhension, sa disponibilité,
son aide et ses précieux conseils qui nous ont été très utile pour*

l'achèvement de ce projet. Un grand merci à

*Mdm.Tourjmene.Mouna qui a accepté d'évaluer et d'examiner
ce travail. Un profond merci à toute personne qui a contribué*

de près ou de loin à la réalisation de ce projet.

DEDICACE

*Nous tenons à dédier cet humble travail comme preuve
de respect, de gratitude et de reconnaissance A nos chers
parents, nos familles et amis A tous ceux que nous
aimons Pour leurs encouragements, soutiens, patiences
et prières.*

Table de matières

INTRODUCTION GENERALE	1
CHAPITRE 1 : ETUDE PRÉALABLE	3
1. Introduction	4
2. Cadre général du projet.....	4
2.1. Problématique et motivation	4
2.1. Solution proposée.....	4
3. Conclusion.....	5
CHAPITRE 2 : SPÉCIFICATION ET ANALYSE DES BESOINS	19
1. Introduction	19
2. Spécification des Besoins	19
2.1. Identification des acteurs	19
2.2. Besoins fonctionnels	19
2.3. Besoins non fonctionnels	20
3. Diagrammes de cas d'utilisation.....	21
3.1. Diagramme du cas d'utilisation général	21
3.2. Description textuelle du cas d'utilisation ‘S’authentifier’	21
3.3. Description textuelle du cas d'utilisation Valider un emprunt d’un livre”	21
3.4. Description textuelle du cas d'utilisation“’Laisser un avis concernant un document emprunté”	21
4. Conclusion.....	25
CHAPITRE 3 :CONCEPTION.....	49
1. Introduction	50
2. Langage de modélisation.....	50
2.1. Unified Modeling Language UML	50
2.2. Choix de l’outil de conception	51
3. Diagrammes de Séquence.....	51
3.1.Diagramme de Séquence du cas d’utilisation “Supervisionner le stockage des livres”	51
3.2.Diagramme de Séquence du cas d’utilisation “ ’ Valider un emprunt d’un livre ‘ ’.....	52
4. Diagramme de classes	57
5. Conclusion.....	57
CHAPITRE 4 : REALISATION	59
1. Introduction	60
2. Environnement de travail.....	Error! Bookmark not defined.
2.1.Environnement matériel.....	Error! Bookmark not defined.

2.2. Environnement logiciel	Error! Bookmark not defined.
2.2.1. Outils Logiciels	Error! Bookmark not defined.
2.2.2. Technologies utilisées.....	Error! Bookmark not defined.
2.2.3. Choix Technologique	Error! Bookmark not defined.
3. Architecture de l'application	60
3.1 Architecture physique	Error! Bookmark not defined.
3.2 Architecture Logique	Error! Bookmark not defined.
4. Interfaces de l'application réalisée	68
4.2. Interfaces de l'application web	68
4.3. Interfaces de l'application mobile.....	73
5. Conclusion	112
CONCLUSION ET PERSPECTIVES	113

Liste des figures

FIGURE 1:DIAGRAMME DE CAS D'UTILISATION GENERAL	21
FIGURE 2: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION "SUPERVISIONNER LE STOCKAGE DES LIVRES"	51
FIGURE 3: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION " VALIDER UN EMPRUNT D'UN LIVRE "	ERROR! BOOKMARK NOT DEFINED.2
7FIGURE 4:DIAGRAMME DE CLASSES	57
FIGURE 5:LOGO GIT	ERROR! BOOKMARK NOT DEFINED.
FIGURE 6:LOGO ANDROID	ERROR! BOOKMARK NOT DEFINED.
FIGURE 7:LOGO SUBLIMETEXT	ERROR! BOOKMARK NOT DEFINED.
FIGURE 8:LOGO IDLE	ERROR! BOOKMARK NOT DEFINED.
FIGURE 9:LOGO STARUML	ERROR! BOOKMARK NOT DEFINED.
FIGURE 10:LOGO GOOGLE CHROME	ERROR! BOOKMARK NOT DEFINED.
FIGURE 11:LOGO HTML5	ERROR! BOOKMARK NOT DEFINED.
FIGURE 12:LOGO CSS3	ERROR! BOOKMARK NOT DEFINED.
FIGURE 13:LOGO BOOTSTRAP	ERROR! BOOKMARK NOT DEFINED.
FIGURE 14:LOGO JAVASCRIPT	ERROR! BOOKMARK NOT DEFINED.
FIGURE 15:LOGO PYTHON	ERROR! BOOKMARK NOT DEFINED.
FIGURE 16:LOGO ANDROID	ERROR! BOOKMARK NOT DEFINED.
FIGURE 17:LOGO FIREBASE	ERROR! BOOKMARK NOT DEFINED.
FIGURE 18:ARCHITECTURE GLOBALE DE L'APPLICATION	ERROR! BOOKMARK NOT DEFINED.
FIGURE 19:PAGE D'AUTHENTIFICATION	69
FIGURE 20:TABLEAU DE BORD DU ADMIN	70
FIGURE 21: INTERFACE LISTE DES « LIVRES »	ERROR! BOOKMARK NOT DEFINED.
FIGURE 22: INTERFACE DE VISUALISATION DES EMPRUNTS	ERROR! BOOKMARK NOT DEFINED.
FIGURE 23:INTERFACE D'AJOUT D'UN NOUVEAU LIVRE	ERROR! BOOKMARK NOT DEFINED.
FIGURE 24:INTERFACE D'AJOUT D'UN NOUVEL EMPRUNT	ERROR! BOOKMARK NOT DEFINED.
FIGURE 25: INTERFACE D'AUTHENTIFICATION MOBILE	ERROR! BOOKMARK NOT DEFINED.
FIGURE 26:INTERFACE LE MENU DE L'APPLICATION MOBILE	ERROR! BOOKMARK NOT DEFINED.
FIGURE 27: L'INTERFACE DES ACTUALITES DE LA SALLE DE LECTURE	ERROR! BOOKMARK NOT DEFINED.
FIGURE 28: LA FENETRE DE DIALOGUE AU CAS DE SATURATION	ERROR! BOOKMARK NOT DEFINED.
FIGURE 29: L'INTERFACE DE CONSULTATION DES LIVRES DISPONIBLES	ERROR! BOOKMARK NOT DEFINED.
FIGURE 30: LA FENETRE DE DETAILS D'UN LIVRE	ERROR! BOOKMARK NOT DEFINED.
FIGURE 31:INTERFACE DE VISUALISATION DES EMPRUNTS	72
FIGURE 32: LA FENETRE DE DETAILS D'UNE EMPRUNTE	72
FIGURE 33: L'INTERFACE DE CONSULTATION DES AVIS PERSONNELS	ERROR! BOOKMARK NOT DEFINED.
FIGURE 34: LA FENETRE DE VALIDER LE DEGRE DE SATISFACTION	ERROR! BOOKMARK NOT DEFINED.
FIGURE 35: L'INTERFACE D'ENVOI D'UNE RECLAMATION	ERROR! BOOKMARK NOT DEFINED.

Liste des Tableaux

TABLEAU 1:REPRESENTATION GENERALE DES ACTEURS ET LEURS ROLES.....	11
TABLEAU 2:DESCRIPTION TEXTUELLE DU CAS D'UTILISATION "S'AUTHTENTIFIER"	2323
TABLEAU 3:DESCRIPTION TEXTUELLE DU CAS D'UTILISATION "VALIDER UN EMPRUNT D'UN LIVRE"	24
TABLEAU 4:DESCRIPTION TEXTUELLE DU CAS D'UTILISATION "LAISSER UN AVIS CONCERNANT UN DOCUMENTEMPRUNTE".....	25

Introduction générale

Le besoin des étudiants et des chercheurs à l'acquisition de supports d'informations a considérablement accru. Ce qui a engendré une surcharge au niveau des bibliothèques et une diversité de leurs contenus. De ce fait, la gestion manuelle de cette masse, est devenu de plus en plus une tâche très lourde et peu efficace pour le bibliothécaire. D'autre part, elle a provoqué également des difficultés aux adhérents à trouver facilement un livre spécifique dans les étagères afin d'extraire une information utile.

Plusieurs travaux dans la littérature ont tenté de répondre à cette problématique. Cependant, deux éléments principaux et non complémentaires caractérisent ces travaux : (1) la gestion classique des bibliothèques et (2) la recherche des documents.

Dans ce contexte se situe notre contribution par le biais de ce projet de fin d'études de développer une solution "intelligente" basée sur la nouvelle technologie de l'internet des objets permettant de trouver des nouvelles orientations à la gestion d'une bibliothèque. Cette solution s'appuie principalement sur l'expérience utilisateur pour permettre à la fois d'automatiser la gestion administrative d'une bibliothèque et d'orienter l'adhérent à trouver facilement les documents les plus pertinents et les plus adéquats à son besoin dans des délais minimums.

Le présent rapport s'articule autour de quatre chapitres principaux.

Dans le premier chapitre nous commençons par présenter le cadre général du projet à travers une petite présentation de l'organisme d'accueil ainsi qu'une présentation du travail demandé. Ensuite, nous continuons par présenter le champ d'étude ainsi qu'analyser l'existant et nous dégagons par la suite les anomalies des solutions implémentées afin de mettre en évidence les objectifs que nous visons dans notre solution.

Le chapitre suivant sera consacré à la description des spécifications des besoins fonctionnels et non fonctionnels et à la présentation des diagrammes de cas d'utilisations.

Le troisième chapitre contient une caution de l'architecture de l'application et sa conception tout en précisant les différents modules qui sont en interaction.

Dans le dernier chapitre, nous expliquons les différents aspects de notre travail, ensuite nous exposons une description de l'environnement de travail utilisé, ainsi qu'une description de la phase d'implémentation de notre application.

Enfin, nous clôturons notre rapport par une conclusion générale exposant une synthèse du travail et quelques perspectives.

CHAPITRE 1 : ETUDE
PRÉALABLE

1. Introduction

Dans ce présent chapitre, nous commençons par mettre le projet dans son cadre général en présentant le sujet. Ensuite analyser le système actuel, fixer les objectifs à atteindre et proposer notre solution. Nous finissons notre chapitre par une conclusion.

2. Cadre général du projet

2.1. Problématique et motivation

Trouver une place libre en bibliothèque universitaire (BU) relève parfois du défi, surtout en période de révisions. De plus, un encombrement de places qui dépasse parfois la capacité maximale de la salle et un control non sérieux de l'état de la BU a rendu la situation plus grave. Dans le cadre de Smart Enis, une université doit certainement garantir un accès facile et un control efficace à sa bibliothèque.

Notre projet consiste à développer une application mobile sous plateforme Android, destinés aux étudiants et un site web destinés aux bibliothécaires, pour consulter l'état de la BU.

Dans ce contexte se situe notre contribution de trouver une nouvelle solution pour remédier à tous les problèmes précédemment mentionnés. Mais avant de démarrer la phase de réalisation de notre futur système, il est judicieux de faire une étude approfondie des solutions existantes qui fera l'objet de la section suivante.

2.2. Solution proposée

Pour surmonter les problèmes identifiés dans les solutions existantes, nous proposons concevoir et développer d'une part, une solution web permettant d'automatiser la gestion manuelle des bibliothèques et d'aider bibliothécaire à bien exécuter ses tâches. D'autre part, une application mobile qui facilite à l'étudiant de consulter l'état de la bibliothèque en réduisant le temps de la recherche une place libre et l'encombrement dans la bibliothèque.

Les fonctionnalités principales qui seront prises en compte dans notre solution sont les suivantes :

- Décharger le bibliothécaire des tâches lourdes et délicates et prévoir une meilleure gestion administrative.
- Offrir aux administrateurs un accès rapide et confidentiel aux informations.

- Faciliter la gestion des empruntes.
- Garantir au bibliothécaire de bien gérer les prêts, les retours des documents.
- Avertir les usagers au cas de retard.
- Avoir la possibilité de consulter en temps réel les nombre de chaises et tables disponibles dans la bibliothèque.
- Recevoir et traiter les réclamations envoyées par les étudiants.
- Permettre à l'étudiant d'avoir une idée sur les places disponibles dans la bibliothèque.
- Avoir des statistiques sur l'accès des étudiants à la bibliothèque (année, mois, jour) et sur le taux d'utilisation de l'application.

3. Conclusion

Dans ce chapitre, nous avons situé notre projet dans son contexte. En effet, nous avons présenté le sujet du projet abordé. En second lieu, nous avons passé à étudier l'existant ce qui nous a permis de dégager les défaillances et les lacunes et nous avons donné notre propre vision ainsi qu'une nouvelle solution.

L'analyse des besoins fonctionnels et non fonctionnels de notre projet fera l'objet du chapitre suivant.

CHAPITRE 2 : SPÉCIFICATION ET ANALYSE DES BESOINS

1. Introduction

Dans ce présent chapitre, nous commençons par présenter la première phase du cycle de développement du logiciel qui consiste à l'analyse et l'extraction des besoins fonctionnels et non fonctionnels de notre application. Ensuite, nous présentons les acteurs réactifs du système ainsi que leurs diagrammes de cas d'utilisations.

2. Spécification des Besoins

Dans cette partie, nous allons préciser les besoins fonctionnels et non fonctionnels de notre application pour s'assurer du bon fonctionnement du système afin de satisfaire les exigences et les attentes de nos clients.

2.1. Identification des acteurs

Dans le tableau ci-dessous nous avons représenté les acteurs principaux avec une description de chacun.

Tableau 1:Représentation générale des acteurs et leurs Rôles

Acteur	Description
Bibliothécaire	C'est le gestionnaire de la bibliothèque, il gère principalement l'approvisionnement des livres et les emprunts de la part des étudiants (les retours et les retards), fixe les horaires de la salle de lecture , contrôle son état (nombre de places et climatisation) et répond aux réclamations reçues de la part des étudiants.
Etudiant	C'est le consommateur de service, un acteur dont le rôle est d'une emprunter un livre, laisser un avis concernant un livre qu'il a déjà emprunté et envoyer une réclamation aux bibliothécaires s'il en a besoin.

2.2. Besoins fonctionnels

Notre application doit garantir plusieurs fonctionnalités destinées aux acteurs définis par notre système que nous allons les détailler en ce qui suit.

Rappelons que notre projet est composé de deux modules :

- Le premier module, est une application web consacrée à la gestion administrative du système destiné au bibliothécaire.
- Le second module, est une application mobile qui traitera et répondra aux besoins fonctionnels des étudiants.

Les principales fonctionnalités que nous avons dégagées pour :

- La partie Web : (Bibliothécaire)
 - La gestion des emprunts et des retours des documents.
 - Le control du nombre des chaises existants dans la salle de lecture.
 - Consultation du nombre de places disponibles dans la salle de lecture.
 - Consultation de la température et de la climatisation de la salle de lecture.
- La partie mobile : (Etudiant)
 - Consultation des horaires de la salle de lecture.
 - Consultation du nombre des places disponibles dans la salle de lecture.
 - Recevoir des notifications sur la disponibilité d'une place de nouveau après l'indisponibilité totale des places.
 - Consultation des livres disponibles au magasin de la BU.
 - Consultation des avis des Etudiant sur un livre disponible.
 - Consultation des livres qu'il les a empruntés.
 - Recevoir des notifications sur les dates de retour des emprunts.
 - Ajouter un avis sur un livre qui a déjà pris.
 - Envoyer une réclamation aux bibliothécaires.

2.3. Besoins non fonctionnels

Après avoir identifié les besoins fonctionnels de notre système, nous avons veillé également à la prise en considération de certains besoins non fonctionnels tout au long de notre processus de développement à savoir :

- **La Simplicité, l'ergonomie et la clarté :**
L'application doit fournir une interface simple et conviviale qui respecte bien les règles d'ergonomie afin de faciliter l'accès à l'information et de garantir aussi la facilité d'exploitation et de manipulation du système.
- **La rapidité :**
Le système doit garantir l'accès à l'information à un temps de réponse minimal.

- **La compatibilité :**

Le système doit fonctionner correctement et être compatible à n'importe quel navigateur pour la partie web et à n'importe quelle version du système d'exploitation à la partie mobile.

- **L'efficacité et la performance :**

Le système doit être efficace et répond exactement aux exigences de l'utilisateur d'une manière raisonnable et optimale en toutes les circonstances.

- **La modularité et l'extensibilité**

Le système doit être modulaire et répond au besoin de l'évolutivité c'est à dire notre code doit respecter les bonnes pratiques de développement pour garantir la lisibilité et la compréhension afin de pouvoir le maintenir rapidement et l'ajouter de nouvelles fonctions au cas de besoin.

- **La sécurité**

Le système doit assurer la confidentialité et la sécurité des données par deux mécanismes : le premier mécanisme est l'authentification qui repose sur l'utilisation d'un login et un mot de passe pour pouvoir prouver l'identité de la personne qui demande de se connecter au système et le deuxième c'est l'autorisation qui empêche les utilisateurs connectés d'accéder à des informations dont ils n'ont pas de privilège.

3. Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes qui permettent de délimiter le système en illustrant une vision globale sur le comportement fonctionnel de ce dernier.

Ces diagrammes décrivent l'ensemble d'interactions (actions et réactions) possibles entre les différents intervenants et le système et elles permettent de présenter et de comprendre les besoins réels des utilisateurs.

3.1 Diagramme du cas d'utilisation général

La figure 7 montre le diagramme de cas d'utilisation général de la partie Web de notre application.

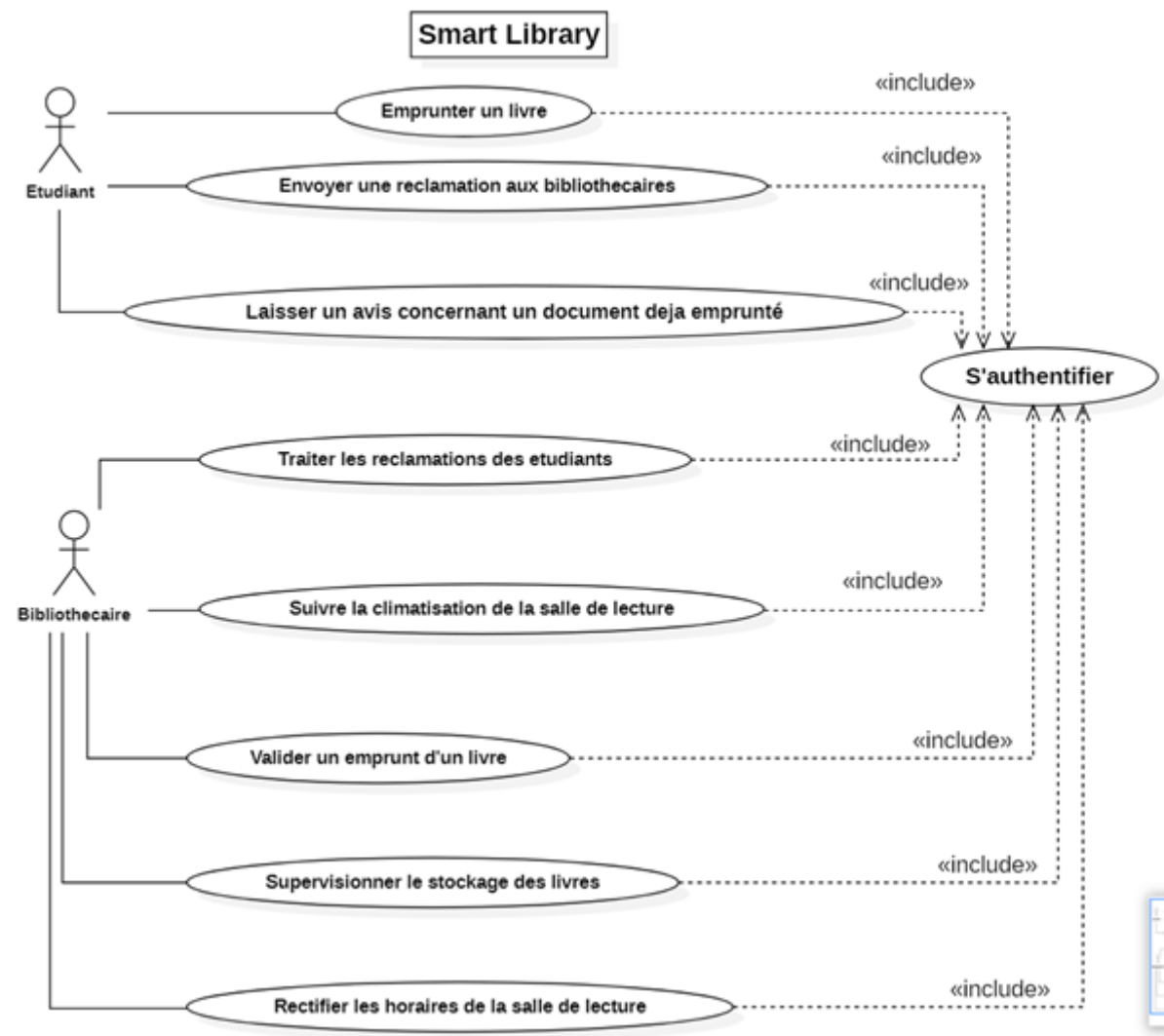


Figure1 : Diagramme de cas d'utilisation générale

Tous les diagrammes de cette partie nécessitent l'authentification de l'intervenant via un login et un mot de passe pour prouver son identité avant qu'il ait la permission d'accéder aux différentes fonctionnalités de notre système.

Nous Présentons dans ce qui suit une description Textuelle des scénarios du cas d'utilisation " S'authentifier", "Laisser un avis concernant un document emprunté" et " Valider un emprunt d'un livre ".

3.2 Description textuelle du cas d'utilisation "s'authentifier "

C'est le premier cas d'utilisation qui sera demandé par notre système car tous les autres scénarios exigent que l'intervenant doive être authentifié.

Tableau 2:Description textuelle du cas d'utilisation “s'authentifier ”

Sommaire	
Nom du cas d'utilisation	s'authentifier
Acteur	Bibliothécaire ou Etudiant
Résumé	L'acteur concerné saisit son login et son mot de passe pour pouvoir accéder à son espace.
Description de l'enchaînements	
Pré-condition	L'acteur concerné doit avoir un compte déjà
Post-condition	Accès à l'espace approprié.
Scénario nominal	<p>1- L'acteur concerné introduit son login et son mot de passe.</p> <p>2-Le système vérifie si tous les champs obligatoires sont renseignés et valides.</p> <p>3- L'application vérifie si l'acteur existe dans la liste des bibliothécaires (partie web) ou étudiants (partie mobile) .</p> <p>4- L'application redirige l'administrateur à l'espace approprié.</p>
Scénario alternatif	<p>A2 : Les données saisies sont invalides :</p> <p>1-Le système affiche un message d'erreur.</p> <p>2- Le scénario nominal reprend au point 1.</p> <p>A3 : L'administrateur n'existe pas ou le mot de passe est invalide :</p> <p>1- Le système affiche un message d'erreur.</p> <p>2- Le scénario nominal reprend au point 2.</p>

3.3 Description textuelle du cas d'utilisation “Valider un emprunt d'un livre”

Tableau 3 : Description textuelle du cas d'utilisation “ Valider un emprunt d'un livre ”

Sommaire	
Nom du cas d'utilisation	Ajouter une emprunte
Acteur	Bibliothécaire
Résumé	Le bibliothécaire ajoute une nouvelle emprunte d'un document au système
Description de l'enchaînements	
Pré-condition	Le bibliothécaire doit être authentifié. Le livre est disponible.
Post-condition	Afficher la liste des empruntes
Scénario nominal	<p>1- Le bibliothécaire clique sur le bouton ajouter une nouvelle emprunte.</p> <p>2- Le système affiche le formulaire d'ajout.</p> <p>3- Le bibliothécaire entre l'id de l'emprunteur et choisit le livre concerné par cette emprunte en précisant son nom et son id.</p> <p>4- Le bibliothécaire clique sur le bouton ajouter.</p> <p>4- Le système affiche un message de succès et le bibliothécaire sera redirigé vers la liste de tous les empruntes.</p>
Scénario alternatif	<p>A1: Les données sont invalides :</p> <p>1- Le système affiche un message d'erreur.</p> <p>2- Le scénario nominal reprend au point 2.</p>

3.4 Description textuelle du cas d'utilisation ‘‘Laisser un avis concernant un document emprunté’’

Tableau 4 : Description textuelle du cas d'utilisation ‘‘Laisser un avis concernant un document emprunté’’

Sommaire	
Nom du cas d'utilisation	Préciser sa satisfaction concernant un document emprunté.
Acteur	Etudiant
Résumé	L'étudiant note le degré de sa satisfaction de 0 (minimum) à 5 (maximum) à propos un document déjà emprunté
Description de l'enchaînements	
Pré-condition	L'Etudiant doit être authentifié. Le document à commenter est emprunté.
Post-condition	Afficher la liste des satisfactions des documents déjà empruntés par l'étudiant.
Scénario nominal	1- L'Etudiant clique sur la ligne du document concernée. 2- Le système affiche la fenêtre de la précision du degré de satisfaction (le degré de satisfaction par défaut égal à 0). 3- L'Etudiant précise son degré de satisfaction qui varie de 0 à 5. 4- L'Etudiant clique sur le bouton changer. 6-Le système affiche un message de succès de changement .

4. Conclusion

La spécification des besoins procure une vision globale et claire du système et elle permet également à l'utilisateur d'appréhender rapidement le fonctionnement général et de comprendre les détails de chaque fonctionnalité.

Dans ce chapitre nous avons essayé de bien présenter les besoins fonctionnels et non fonctionnels. Nous avons présenté les acteurs de notre système tout en spécifiant les différentes interactions entre eux en illustrant les diagrammes de cas d'utilisation de chaque acteur.

Les besoins identifiés dans ce chapitre vont être la base sur laquelle nous allons réaliser la conception de notre application. Cette conception fera l'objet du prochain chapitre.

CHAPITRE 3 :CONCEPTION

1. Introduction

Après l'identification des besoins fonctionnels de notre projet, nous concentrons dans le présent chapitre à présenter la partie conception de notre application.

C'est une étape ayant le but de décomposer un système complexe en plusieurs modules ce qui facilitera sa compréhension et par suite sa mise en disposition et sa maintenance.

Pour ce faire, nous commençons dans un premier temps, de définir le langage de modélisation ainsi que l'outils de conception que nous avons utilisé. Et puis, dans un deuxième temps, nous présentons la conception détaillée en illustrant quelques diagrammes de séquences et le diagramme de classes de notre système.

2. Langage de modélisation

La modélisation est une projection de l'image du monde réel sous forme de graphe et de schéma permettant de faciliter la compréhension des activités d'un système.

Pour la représentation des diagrammes de notre projet, nous avons choisi d'utiliser le langage de modélisation UML qui permet une représentation claire des activités, des données et des traitements.

2.1. Unified Modeling Language UML



Figure 1:logo UML

UML est un langage orienté objet permet la modélisation standardisée de l'architecture du système. Ce langage définit des fondements normalisés basés sur la notation graphique qui couvre le cycle de développement d'un logiciel de la spécification des besoins jusqu'à l'implémentation. Il offre également de différents diagrammes permettant de représenter le système en son ensemble et de couvrir les différents niveaux d'abstraction ce qui aide à la bonne compréhension du fonctionnement du système.

2.2. Choix de l'outil de conception



Figure 2:logo StarUML

Le logiciel StarUML est un logiciel open-source cédé par son ancien éditeur sous licence GNU GPL, dédié aux plateformes Windows, il est développé en Delphi.[URL3]

C'est un outil permettant principalement d'éditer les différents diagrammes UML et ayant plusieurs avantages notamment la possibilité de générer le squelette des classes en langages en java php# et c++ à partir des diagrammes élaborés en reposant sur l'utilisation de plusieurs plugins et l'exportation des données sur format XML

Dans notre projet, Nous avons utilisé le logiciel StarUML Version 3 comme outil de modélisation UML

3. Diagrammes de Séquence

Dans cette section, nous décrivons les diagrammes de séquences relatifs à quelques scénarios des cas d'utilisation de notre future application. Ces diagrammes sont la représentation graphique des interactions entre le système et les acteurs tout en respectant un ordre chronologique.

3.1. Diagramme de Séquence du cas d'utilisation "Supervisionner le stockage des livres"

Après authentification, le bibliothécaire peut ajouter un nouvel livre.

D'emblée, ce dernier doit se déplacer vers la rubrique de documents, cliquer sur ajouter un nouveau livre et remplir le formulaire, le système va vérifier la possibilité d'ajouter le nouveau livre : cela se fait en vérifiant les données entrées.

Si l'ajout d'un nouvel livre est possible, le système va afficher toute ses données, sinon le système va afficher un message d'erreur indiquant la donnée erronée.

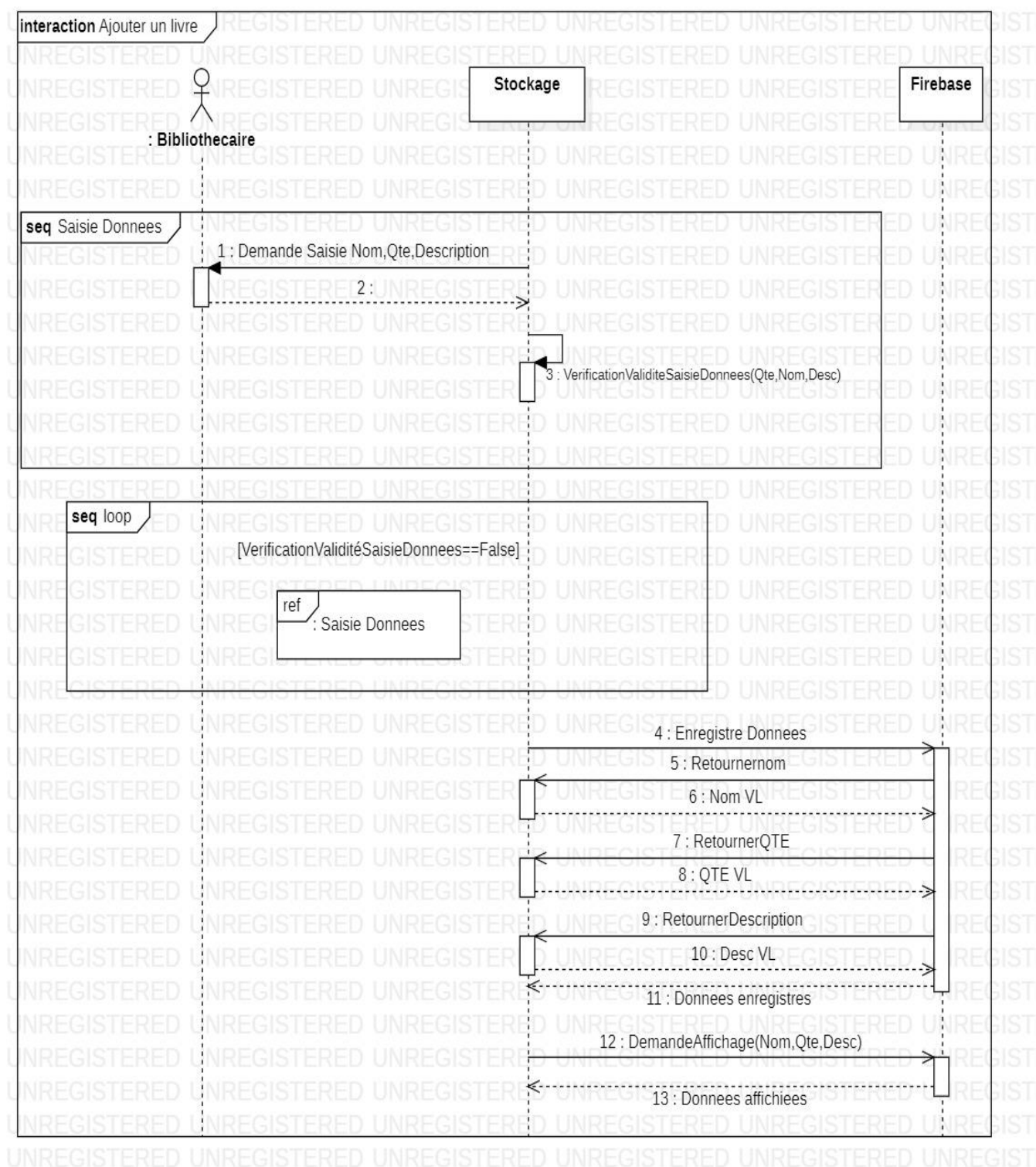
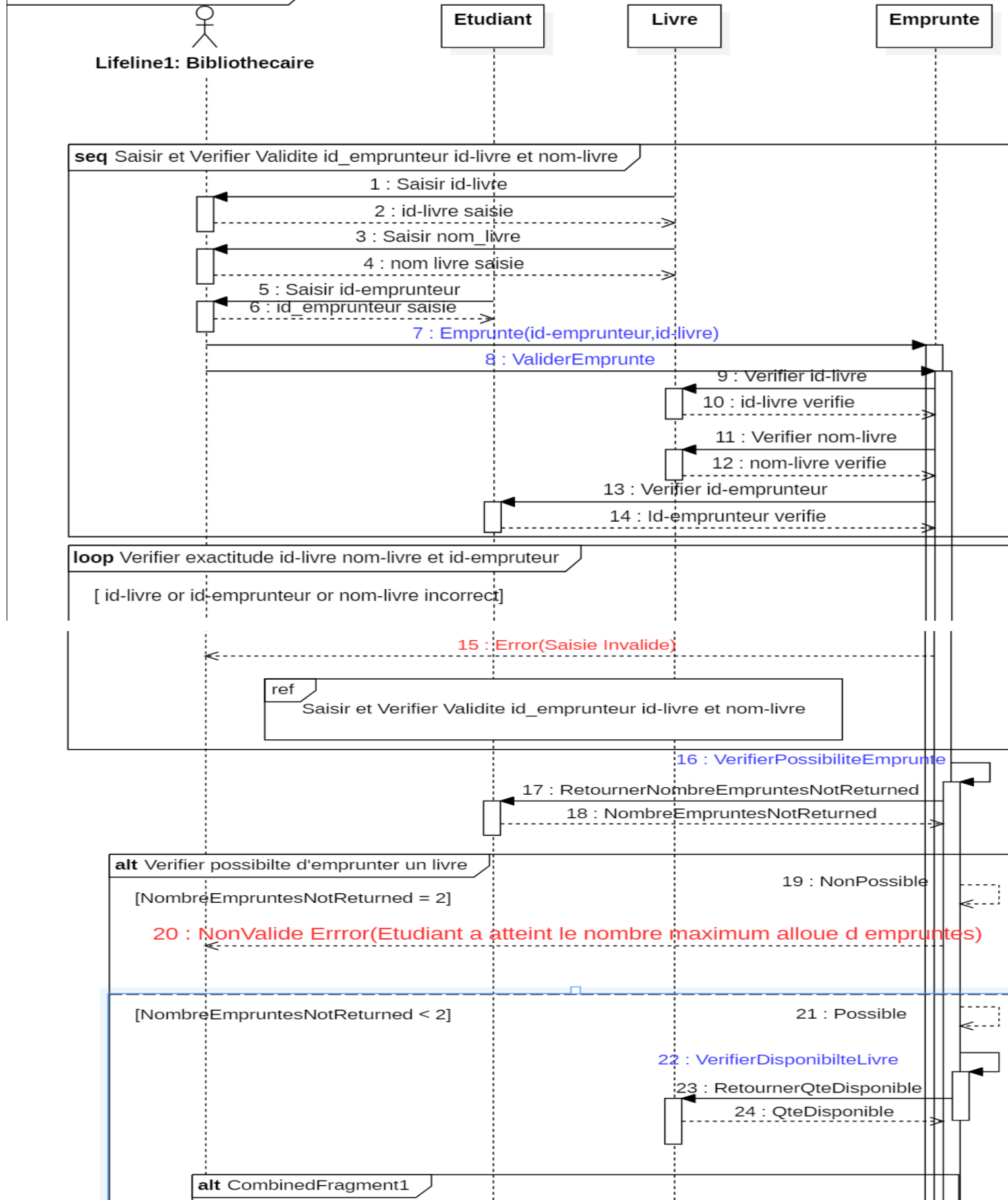


Figure2 : Diagramme de Séquence du cas d'utilisation «Ajouter un livre»

3.2. Diagramme de Séquence du cas d'utilisation “ Valider un emprunt d’un livre “

interaction SequenceDiagram1



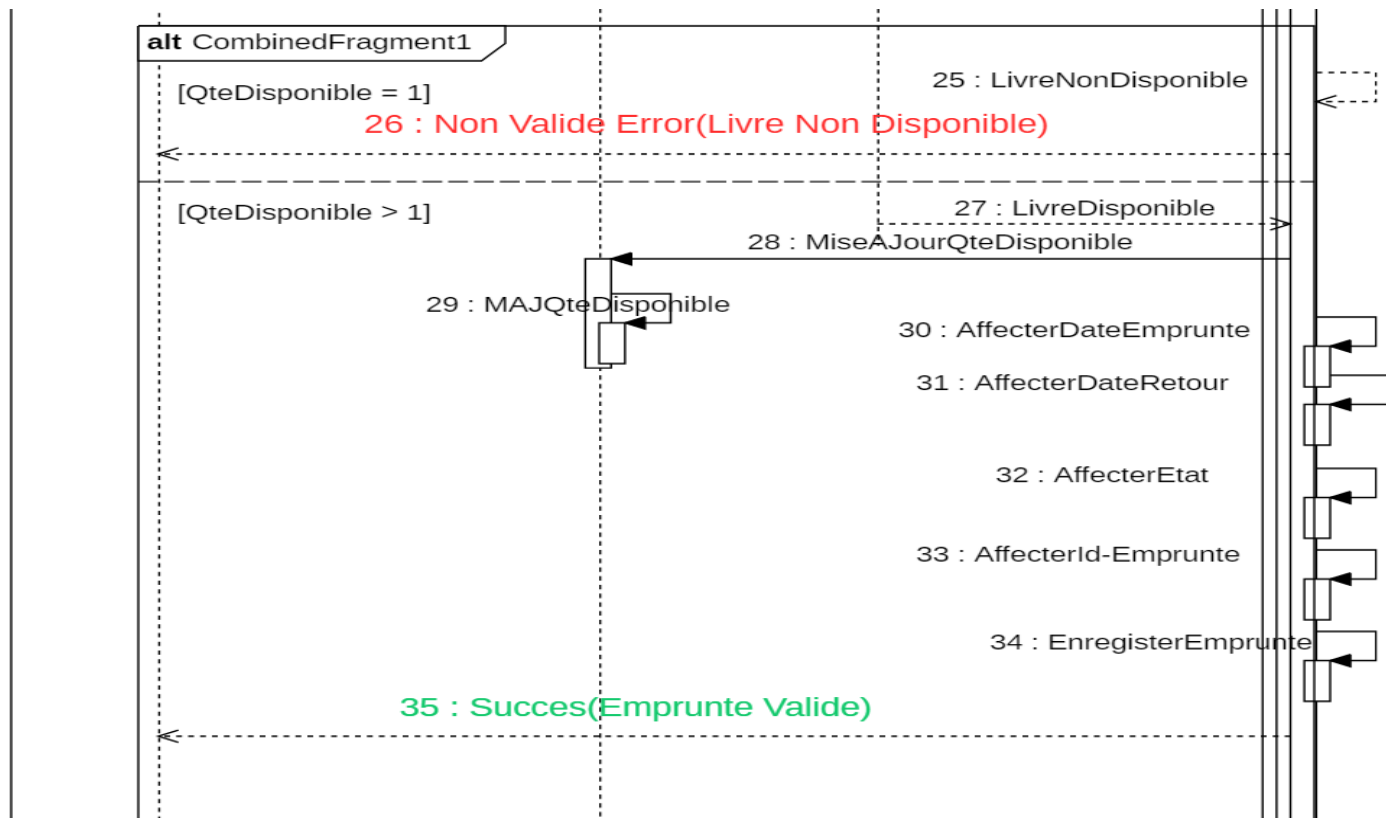


Figure3 : Diagramme de Séquence de cas d'utilisation ‘ Valider un emprunt d’un livre ‘,

4. Diagramme de classes

Le diagramme de classes est un diagramme d'un aspect statique dans le standard UML permettant de modéliser les entités du système ainsi que les relations établies entre elles.

La figure X représente le diagramme de classes de notre application.

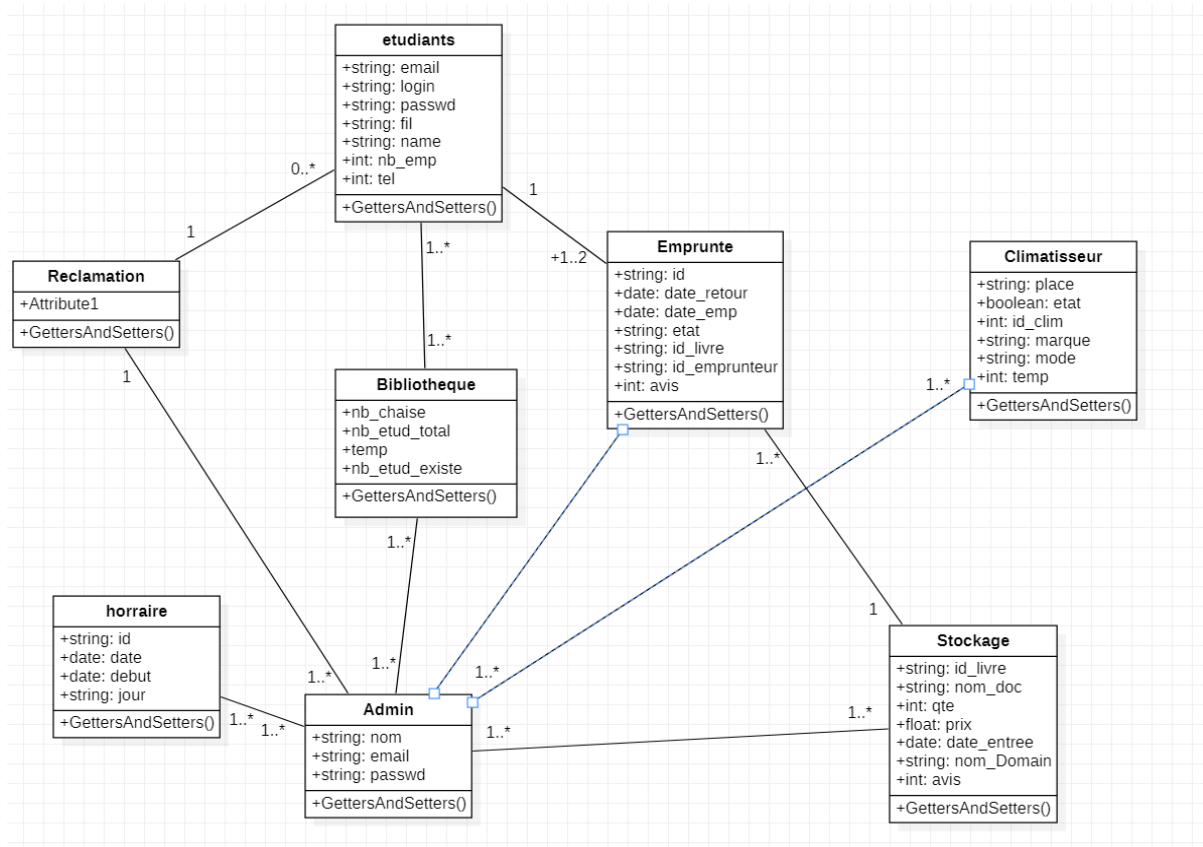


Figure4:Diagramme de classes

5. Conclusion

Tout au long de ce chapitre, nous avons présenté en premier lieu la langage et l'outils de modélisation que nous avons utilisé pour illustrer la conception détaillée de notre système. Ensuite, dans un deuxième lieu, nous avons détaillé les scénarios d'interaction entre les différentes entités de système à travers l'élaboration des diagrammes de séquences et le diagramme de classes.

Dans le chapitre suivant, nous allons nous intéresser par présenter la partie de réalisation de notre application.

CHAPITRE 4 : REALISATION

1. Introduction

La phase de réalisation est l'aboutissement de toutes les phases précédentes et c'est dans ce présent chapitre que nous allons exposer les différentes étapes de la réalisation de notre application.

Dans un premier instant, nous présentons les outils de développement utilisés ainsi que l'architecture que nous avons adopté dans travail. Puis, dans un deuxième instant, nous exposons les captures d'écran décrivant le fonctionnement de la solution développée.

2. Environnement de travail

Pour la réalisation de notre application, nous avons eu recours à plusieurs moyens, qui sont matériel et logiciel.

2.1. Environnement matériel

Ce travail a été réalisé avec un PC portable et un Smartphone dont les caractéristiques suivantes :

Tableau 3:Environnement matériel utilisé

Caractéristiques	Ordinateur	Smartphone
Marque	DELL inspiron	Samsung
Processeur	Intel® Core™ i7-7700HQ CPU @2.8GHz Up to 3.8GHz	Kirin 650
Mémoire vive	16 GO DDR4	3GO
Disque dur	1 TO	16 GO
Système d'exploitation	Windows® 10 64bit	Android 7.0

2.2. Environnement logiciel

Cette section est réservée pour présenter les différents outils que nous avons utilisé tout au long du phase de développement de notre projet

2.2.1. Outils Logiciels

- **GitHub Desktop**

GitHub Desktop est un logiciel client qui permet aux utilisateurs d'accéder à des dépôts Git depuis leur ordinateur pour Windows et Mac. Ce logiciel permet de gérer simplement les projets en affichant une interface très simple à utiliser, permettant d'exécuter les commandes Git (commit, pull, push, fetch, merge, etc.) sans avoir à recourir aux lignes de commande.



Figure5 : Logo GitHub Desktop

- **Android Studio**

Android Studio [URL5] est l'environnement de développement proposé par Google à ses développeurs pour créer des applications Android. Cet IDE possède un éditeur de code intégrant des fonctions intelligentes comme l'autocomplétions et l'analyse du code.

Il est équipé aussi par de outils permettant d'émuler le résultat du travail développé sur plusieurs dispositifs et équipements mobiles ayant des configurations différentes avec des tailles d'écran différentes, ainsi qu'un moniteur permettant de vérifier l'optimisation de l'application en affichant l'utilisation de la mémoire durant son fonctionnement.



Figure6 : Logo Android Studio

- **Sublime Text**

Sublime Text est un éditeur de code open-source, gratuit et multi codé en [C++](#) et [Python](#), disponible sur [Windows](#), [Mac](#) et [Linux](#). Il prend en charge un certain nombre de langages de

programmation différents et assure la coloration syntaxique pour HTML, Java, JSP, JavaScript, JSON, etc.



Sublime Text

Figure7 : Logo Sublime Text

- **IDLE**

IDLE est un environnement de développement intégré pour le langage Python. IDLE signifie « Integrated DeveLopment Environment » (« environnement de développement intégré », en français). Les principales fonctionnalités de IDLE sont :

- l'éditeur de texte avec coloration syntaxique, l'autocomplétion, l'indentation ;
- le terminal Python avec coloration syntaxique ;
- le débogueur intégré avec avancement par étape, point d'arrêts persistants et pile d'appels.



Figure8 : Logo IDLE

- **StarUML**

StarUML est un logiciel de modélisation UML, qui a été "cédé comme open source" par son éditeur. Étant simple d'utilisation, nécessitant peu de ressources système, supportant UML 2, ce logiciel constitue une excellente option pour une familiarisation à la modélisation.



Figure6 : Logo StARUML

- **Google Chrome**

Chrome est un navigateur web propriétaire développé par Google basé sur le projet libre Chromium fonctionnant sous Windows, Mac, Linux, Android et iOS.



Figure6 : Logo Google Chrome

2.2.2 Technologies utilisées

- **HTML5**

HTML5 [URL9] pour « HyperText Markup Language 5 », est une version du célèbre format HTML utilisé pour concevoir les sites Internet. C'est un langage de balisage permettant la mise en forme d'une page Web. HTML5 est une version successeur de la version HTML4.0 Lancée en octobre 2014, cette version a ajouté certaines nouveautés par rapport à la version précédente notamment la possibilité de délimiter le contenu de la page Web en plusieurs sections (Header, Footer, Nav, Main et Aside)



Figure7 : Logo HTML5

- **CSS3**

CSS3[URL10] (Cascading Style Sheets) les feuilles de styles en cascade servent à mettre en forme des documents web de type page HTML ou XML et ayant comme but de dissocier le contenu de la page de son aperçu visuelle. Par l'intermédiaire de propriétés d'apparence et de placement, le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. CSS3 est une version récente du CSS qui apporte de plusieurs nouveautés en termes d'effets et d'apparences des éléments HTML



Figure8 : Logo CSS3

- **Javascript**

Javascript [URL11] est un langage de script orienté objet principalement utilisé pour ajouter un aspect dynamique et interactif aux pages HTML statiques C'est un langage exécuté coté client permettant d'interagir avec l'utilisateur en fonction de ses actions



Figure9 : Logo Javascript

- **Bootstrap**

Bootstrap [URL13] est une infrastructure de développement frontale, gratuite et open source pour la création de sites et d'applications Web. Le Framework Bootstrap repose sur HTML, CSS et JavaScript pour fournir aux développeurs un outils facile et efficace pour la création rapide des sites web responsives qui peuvent s'adapter sur n'importe quelle taille d'écran.



Figure10 : Logo Bootstrap

- **Python**

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

Python est un langage qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses bibliothèques optimisées destinées au calcul numérique.



Figure11 : Logo Android Studio

- **Android**

Android [URL20] est un système d'exploitation open source pour les applications mobiles basé sur le noyau linux et développé par Google.



Figure 12 : Logo Android

- **Firebase**

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel.



Figure13 : Logo Firebase

2.2.3. Choix Technologique

Nous justifions, dans ce qui suit, nos choix technologiques.

- **Choix du système d'exploitation Android pour la partie mobile**

Pour le développement de l'application mobile, nous avons choisi de développer une application Android pour raisons suivantes :

- ✓ Android est le système d'exploitation le plus utilisé dans le monde avec plus de 80 % de parts de marché dans les smartphones, devant IOS d'Apple. La majorité des smartphones utilisent l'Android comme système d'exploitation
- ✓ Un public qui ne cesse pas à grandir d'une année à une autre
- ✓ Le contrôle de publication d'une application sous le store d'Android est beaucoup moins strict et moins cher par rapport à ses concurrents (IOS, Windows phone)
- ✓ Google fournis à ses développeurs un grand support et les produits et les APIs google, tel que google Maps par exemple, peuvent être facilement intégrés dans une application Android

3. Architecture de l'application

La phase de réalisation est l'aboutissement de toutes les phases précédentes et c'est dans ce présent chapitre que nous allons exposer les différentes étapes de la réalisation de notre application.

Dans un premier instant, nous présentons les outils de développement utilisés ainsi que l'architecture que nous avons adopté dans travail. Puis, dans un deuxième instant, nous exposons les captures d'écran décrivant le fonctionnement de la solution développée.

Notre application se compose de 3 modules qui sont :

- Module de la partie backend développé par PHP avec l'utilisation de Firebase comme une base de données
- Module de la partie frontend développé par CSS, JAVASCRIPT.
- Module de l'application mobile développé sous Android.




Figure 18 : Architecture globale de l'application

4. Interfaces de l'application réalisée

Nous présentons, dans cette section, les interfaces développées pour la partie web and mobile

4.1. Interfaces de l'application web

Lors du lancement de l'application, la page d'authentification est affichée, comme le montre la figure X



Email

Password

[Register](#)

Figure14 : Page d'authentification

Pour faire l'authentification, l'administrateur doit saisir son email et son mot de passe. Ces données seront vérifiées par le système. Dans un scénario nominal, le système va envoyer ces données à la partie backend, une fois ses données sont correctes, l'administrateur sera redigé au tableau de bord approprié à son rôle

Dans un scénario alternatif, un message d'erreur est affiché. Ce message est dû à l'invalidité des données entrées ou à l'inexistence du compte saisie.

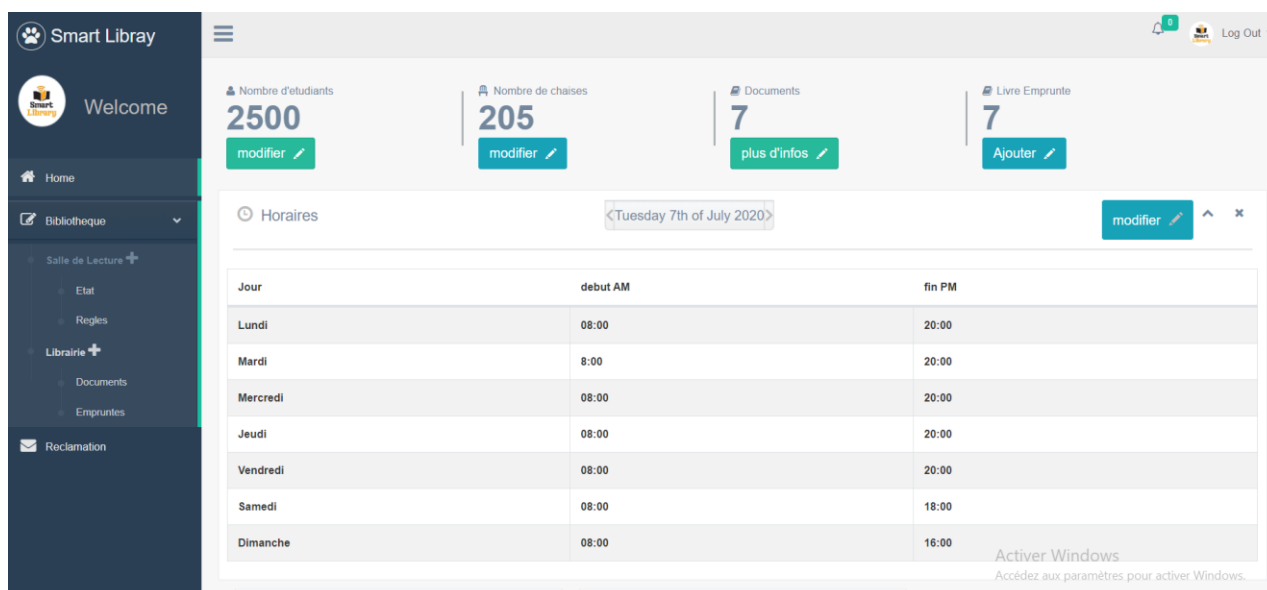


Figure20:Tableau de bord du admin

Cette interface représente le tableau de bord du admin. A travers cette interface cet acteur peut visualiser l'état générale de la bibliothèque notamment le nombre de chaises, le nombre de livres total, le nombre des étudiants existant dans la bibliothèque, les horaires, l'état des climatiseurs aussi cette interface est conçue pour aider le bibliothécaire de savoir quels sont les livres empruntés et qui doivent être retournés. Cela facilite la tâche de cet acteur et lui rappeler d'avertir les adhérents de la date limite de retour du livre afin d'éviter les retours.

- **Gestion des livres**

Cette interface représente le formulaire d'ajout d'un livre, le bibliothécaire doit ajouter les « livres » qui portent des informations génériques sur un livre tel que le titre, le nom d'auteur, la maison d'édition, quantité etc.

La Figure X ci-dessous montre le formulaire d'ajout d'un nouveau livre

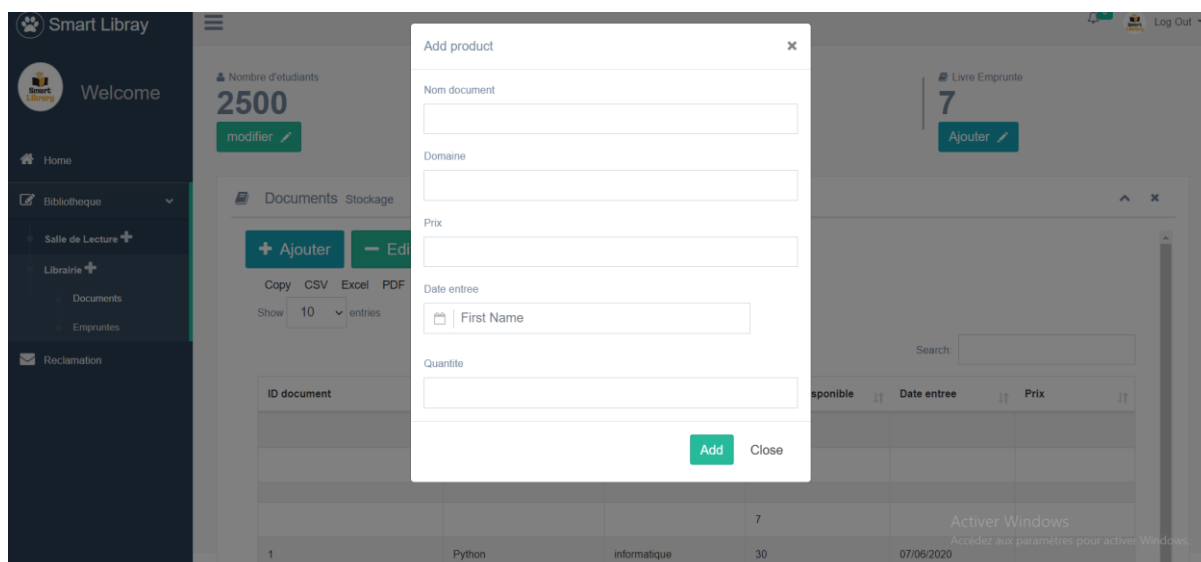


Figure21 : Interface d'ajout d'un nouveau livre

En cliquant sur « Add » le système va afficher tous les « livres » existants dans notre système comme il est indiqué par la figure X.

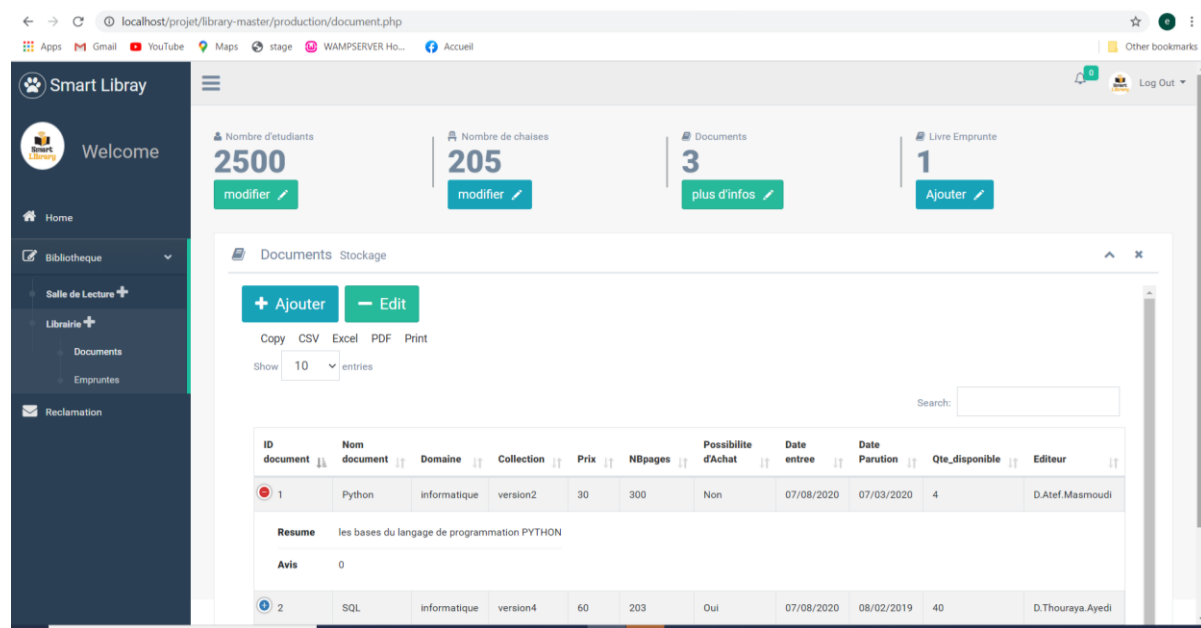
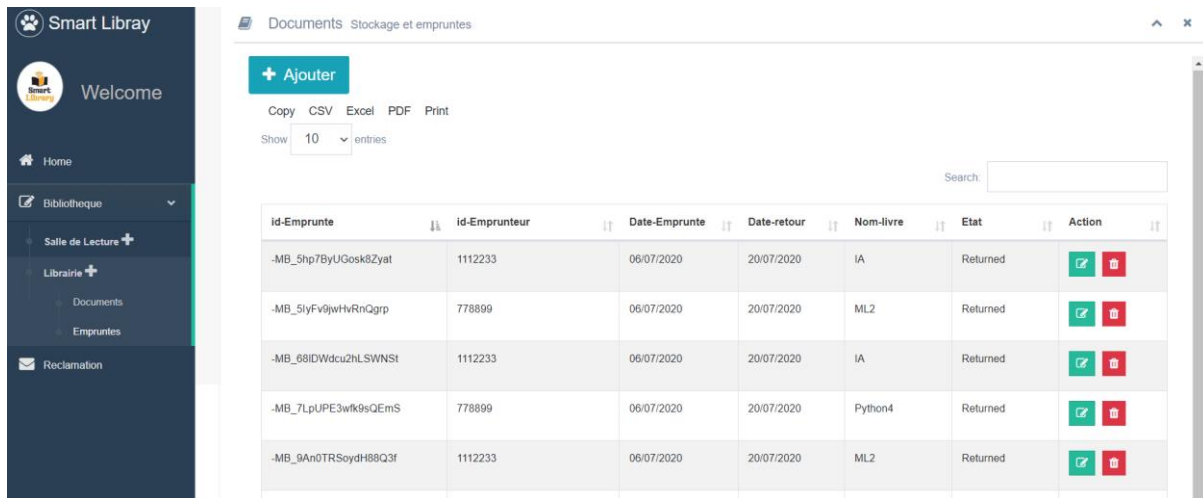


Figure22 : liste des « livres »

- **Gestion des emprunts**

A travers l'interface représentée par la figure X, le bibliothécaire peut consulter la liste des emprunts. Il peut également modifier ou supprimer un prêt existant. Comme il peut adopter les

filtres pour chercher les emprunts effectués dans une période prédéfinie ou pour seulement afficher les emprunts ayant le statut « non encore retourné » par exemple.



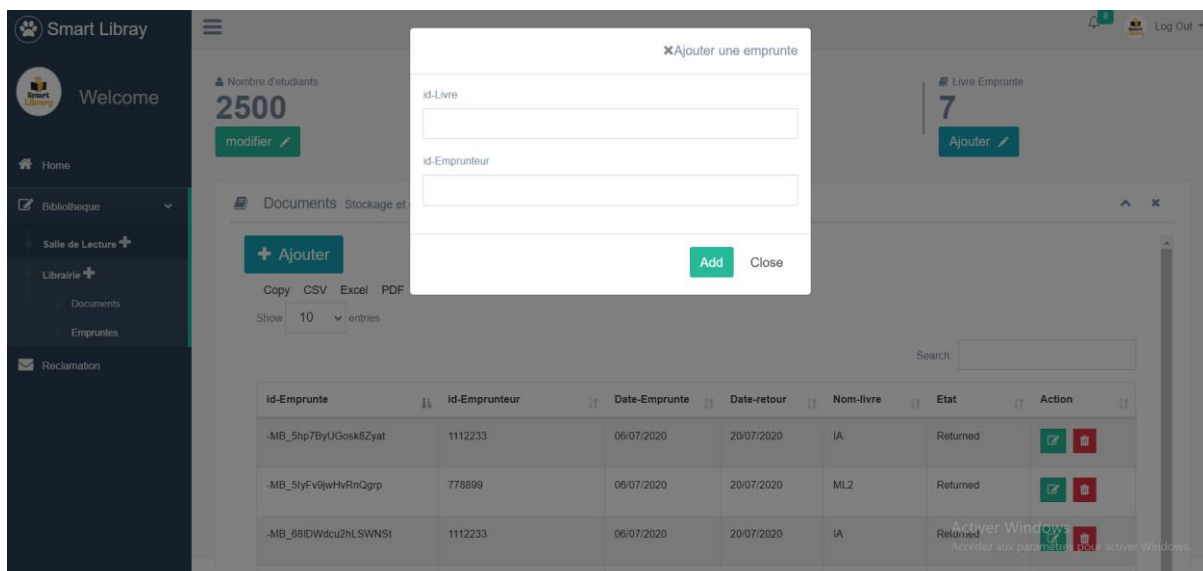
The screenshot shows the 'Smart Library' application interface. On the left is a dark sidebar with navigation options: Home, Bibliothèque (expanded), Salle de Lecture, Librairie, Documents, Emprunts, and Reclamation. The main area is titled 'Documents Stockage et emprunts' and features a '+ Ajouter' button, export options (Copy, CSV, Excel, PDF, Print), and a 'Show 10 entries' dropdown. A table displays borrowing records with columns: id-Emprunte, id-Emprunteur, Date-Emprunte, Date-retour, Nom-livre, Etat, and Action. The table contains five rows of data, all with 'Returned' status. Each row has a green checkmark icon and a red trash icon in the Action column.

id-Emprunte	id-Emprunteur	Date-Emprunte	Date-retour	Nom-livre	Etat	Action
-MB_5hp7ByUGosk8Zyat	1112233	06/07/2020	20/07/2020	IA	Returned	
-MB_5lyFv9jwHvRnQgrp	778899	06/07/2020	20/07/2020	ML2	Returned	
-MB_68IDWdcu2hLSWNSI	1112233	06/07/2020	20/07/2020	IA	Returned	
-MB_7LpUPE3wfk9sQEmS	778899	06/07/2020	20/07/2020	Python4	Returned	
-MB_9An0TRSoydH88Q3f	1112233	06/07/2020	20/07/2020	ML2	Returned	

Figure23 : Interface de visualisation des emprunts

Pour ajouter un nouvel emprunt, le bibliothécaire doit appuyer sur le bouton « Ajouter » qui est placé dans l'interface de gestion des emprunts. Par ce faite une fenêtre contenant le formulaire d'ajout sera affichée. Le bibliothécaire doit saisie l'adhérent concerné et l'identifiant du livre. Selon le choix effectué, le système va vérifier la possibilité du nouvel emprunt

La figure X montre l'interface d'ajout d'un nouvel emprunt.



The screenshot shows the same 'Smart Library' interface as Figure 23, but with a modal window titled 'Ajouter une emprunte' open in the center. The modal contains two input fields: 'id-Livre' and 'id-Emprunteur'. Below these fields are 'Add' and 'Close' buttons. In the background, the 'Documents' section is visible, showing a 'Nombre d'étudiants' of 2500 and a 'Livres Empruntés' count of 7. The table of borrowings is also visible behind the modal.

Figure24 : Interface d'ajout d'un nouvel emprunt

4.3. Interfaces de l'application mobile

Dans cette section, nous présentons quelques interfaces de l'application mobile

- **Interface d'authentification**

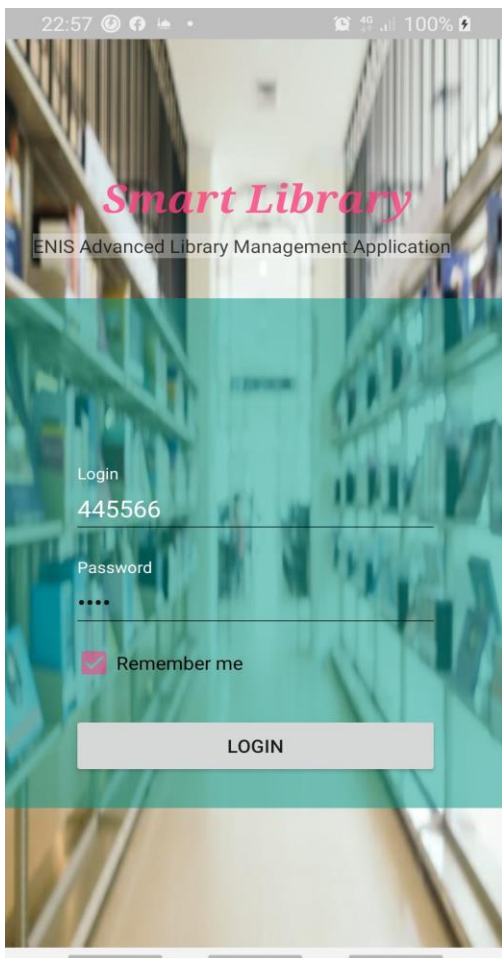
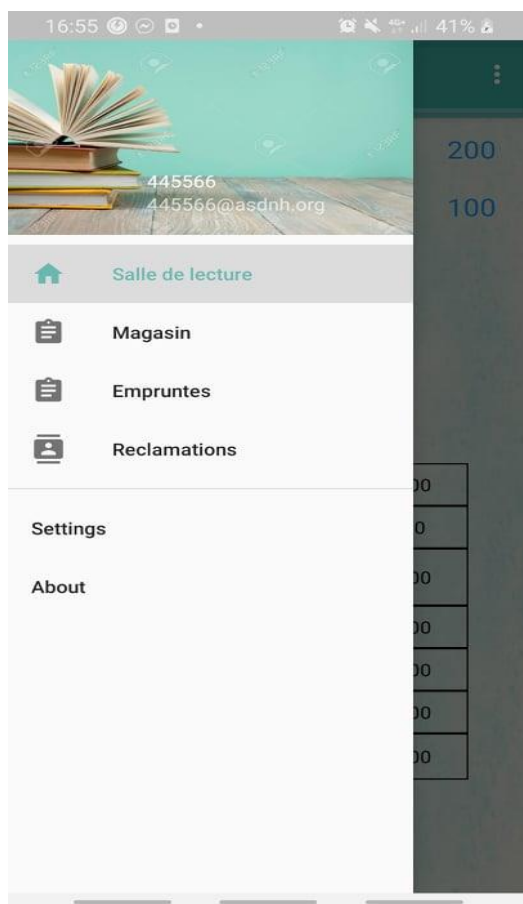


Figure25 : Interface d'authentification mobile

- La première interface représente la page d'authentification. Elle permet à l'Etudiant (déjà inscrit à l'ENIS) de se connecter à notre application via un login (CIN) et un mot de passe.

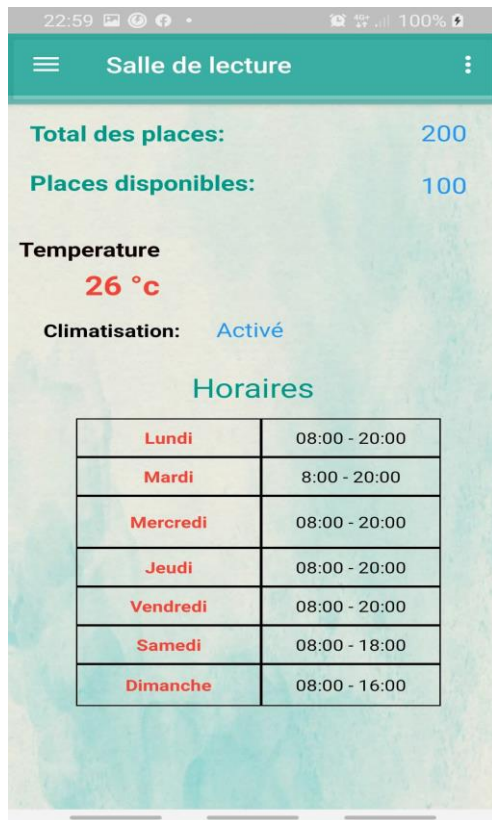
- **Interface de menu principale de l'application**



La figure représente le menu principal de l'application mobile. Il est composé principalement par quatre rubriques Salle de lecture , Magasin , Empruntes et Réclamations.

Figure26 : Le menu de l'application mobile

• Interface de la première rubrique « Salle de lecture »



- La première rubrique du menu « Salle de lecture » permet à l'Etudiant de consulter la page d'actualités de la salle de lecture de l'ENIS. Cette page est conçue principalement pour visualiser le nombre de places disponibles à la salle, ses conditions climatiques (Température et l'activation/désactivation de la climatisation) et la mise à jour de ses horaires.

Figure27 : L'interface des actualités de la Salle de lecture



- En outre, si la salle est pleine, une notification sera envoyée dès qu'une place sera disponible de nouveau.

Figure28 : La fenêtre de dialogue au cas de saturation

- Interface de la deuxième rubrique « Magasin »

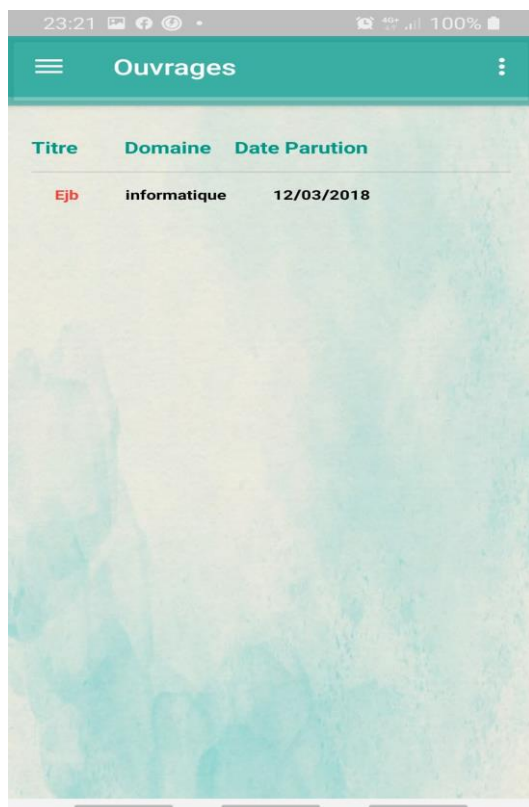


Figure29 : L'interface de consultation des livres disponibles

- La deuxième rubrique « Magasin » permet de consulter les ouvrages disponibles à la BU.



Figure30 : La fenêtre de détails d'un livre

- Tous les détails que l'étudiant en a besoin (Editeur, Nb Pages, Possibilité d'achat, Résumé si disponible, la date d'entrée du livre à la BU et surtout les avis des Etudiants concernant les livres des bibliothèques dont l'Etudiant connecté fait partie)

- **Interface de la troisième rubrique « Empruntes »**

Quant à la troisième rubrique « Empruntes » est composé de deux fragments :

- Le premier fragment « Listes des livres » pour consulter les livres empruntés , le délai de retour de chaque emprunte et son état : retourné / non retourné. En plus si l'application envoie une notification avant un jour du délai de retour

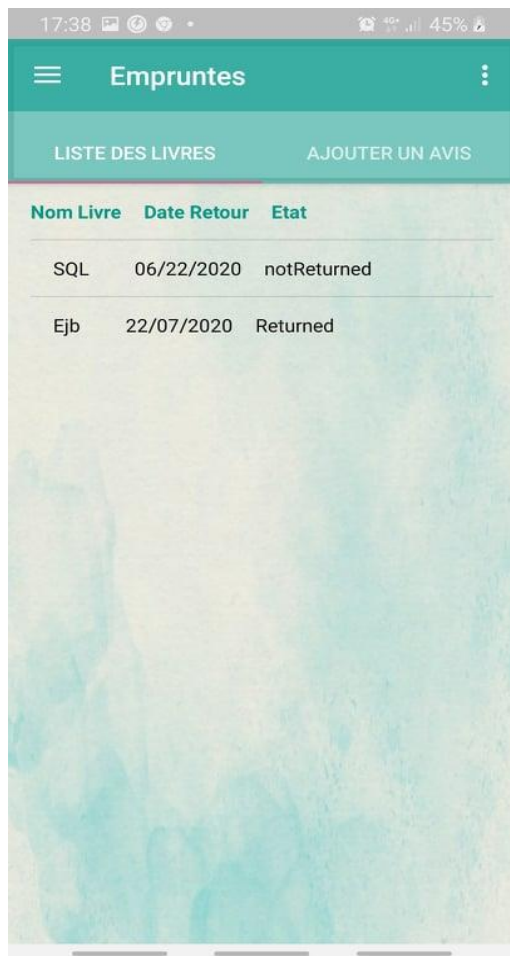


Figure31 : L'interface de consultation des empruntes

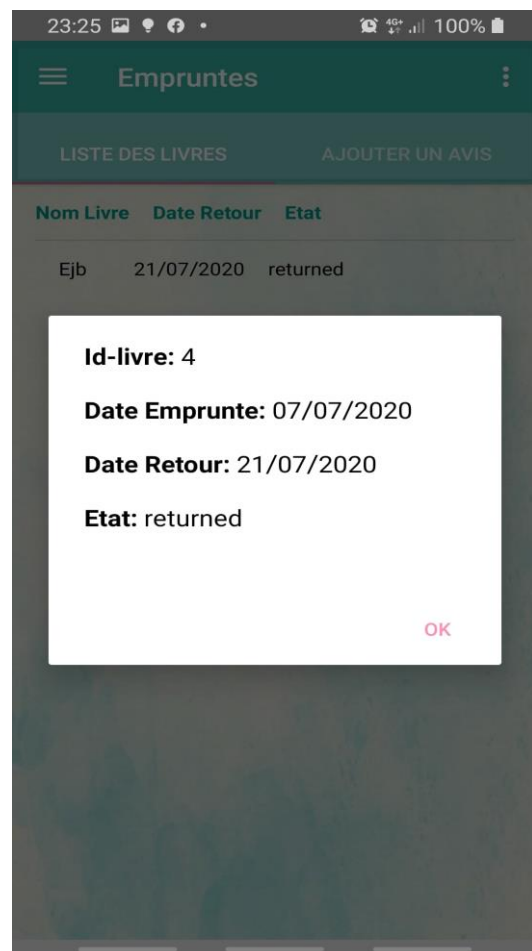


Figure32 : La fenêtre de détails d'une emprunte

- Le deuxième fragment « Ajouter un Avis » et donner son avis (degré de satisfaction). En effet, en consultant les détails d'un livre emprunté, l'Etudiant peut partager un avis concernant ce livre.

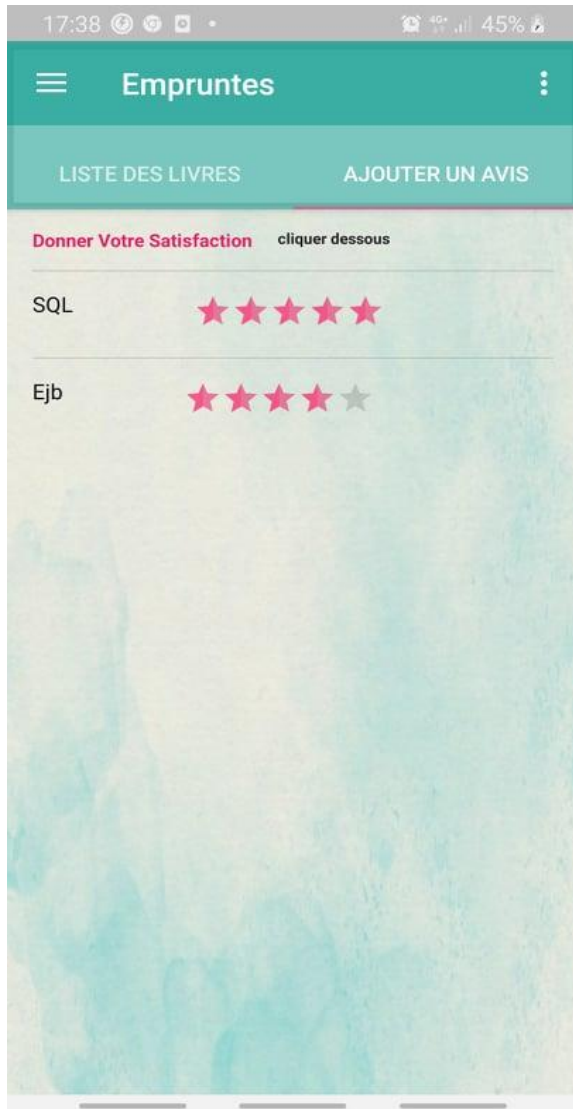


Figure33 : L'interface de consultation des avis personnels

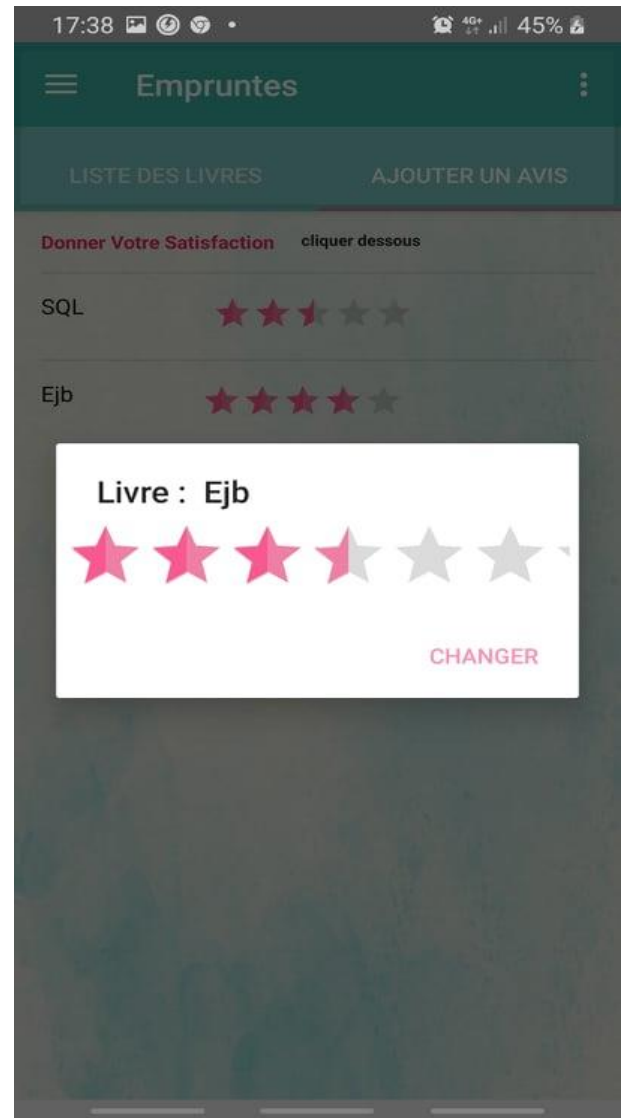


Figure34 : La fenêtre de valider le degré de satisfaction

- **Interface de la dernière rubrique « Réclamations »**

Finalement la dernière rubrique « Réclamations » permet l'envoi d'une réclamation aux bibliothécaires.

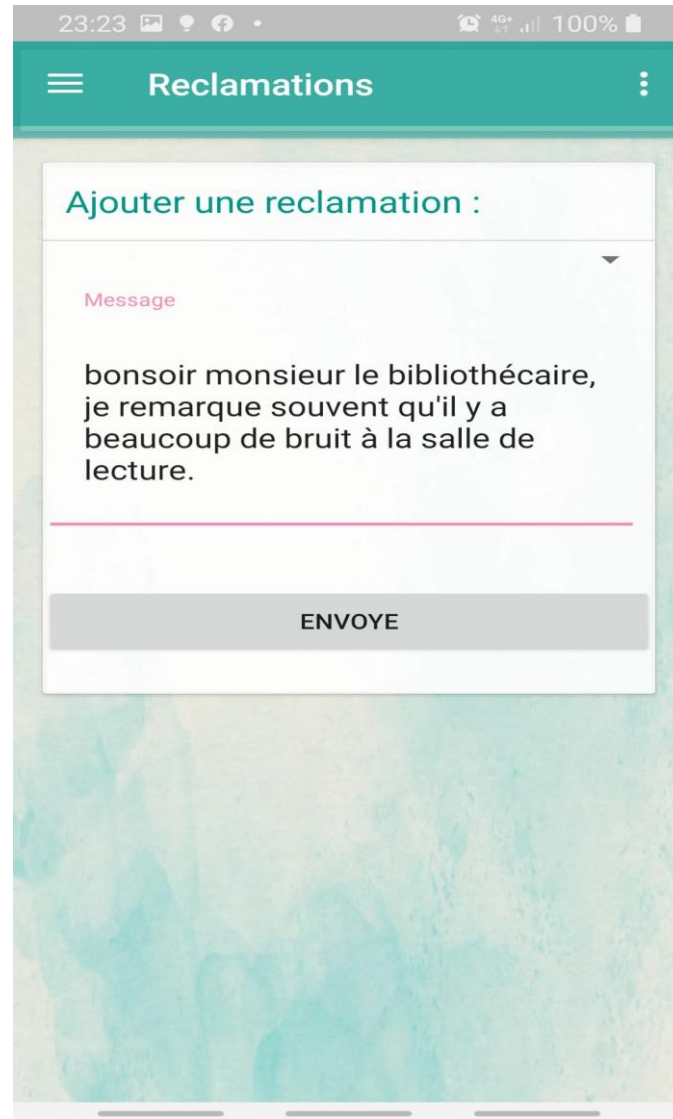


Figure35 : L'interface d'envoi d'une réclamation

5. Conclusion

Dans le présent chapitre, nous avons représenté la partie réalisation de notre projet.

Dans un premier temps, nous avons commencé par définir l'environnement de travail et les différentes technologies que nous avons mis en œuvre pour le développement dans notre solution. Ensuite nous avons détaillé l'architecture physique et logique que nous avons adopté dans notre travail. Finalement nous avons exposé quelques imprimes écran des applications web et mobile réalisées.

Conclusion et perspectives

Ce projet s'inscrit dans le cadre d'un projet de fin d'année ayant comme objectif de réaliser une solution composée d'une application web et une application mobile

Tout à long de ce rapport, nous avons présenté les différents aspects d'élaboration de notre projet de sa phase d'analyse jusqu'à la phase d'implémentation. Nous avons commencé tout d'abord par faire une étude préalable, ce qui nous a permis de dégager les anomalies des solutions existantes et de fixer des objectifs à atteindre pour avoir une solution satisfaisante.

Par la suite nous avons défini les besoins fonctionnels et non fonctionnels à travers le traçage des diagrammes de cas d'utilisation. Partant de ces spécifications, nous avons pu faire la conception de notre application. Finalement nous avons abordé la partie réalisation, dans laquelle nous avons défini l'environnement de travail et les technologies utilisées, ainsi que les différentes interfaces graphiques de notre solution.

Durant ce projet, nous avons eu l'opportunité de renforcer nos connaissances théoriques et de mettre en application nos connaissances acquises le long de nos études. Nous avons eu la chance également de découvrir certaines technologies notamment le PHP, Javascript l'Android.

Bien que notre projet ait atteint ses objectifs, notre travail pourrait être amélioré davantage. En effet, le système réalisé peut être enrichi avec d'autres fonctionnalités telles que l'intégration d'une visite virtuelle au sein d'une bibliothèque permettant d'orienter l'adhérent au place disponible.

Nétographie

[URL1] StarUML, <https://air.imag.fr/index.php/StarUML>

[URL2] Android

studio, <https://www.01net.com/telecharger/windows/Programmation/creation/fiches/129241.html>

[URL3] HTML5, <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203257-html5-hypertext-markup-langage5-definition-traduction/>

[URL4] CSS3, https://fr.wikibooks.org/wiki/Le_langage_CSS

[URL5] Javascript, <https://www.futura-sciences.com/tech/definitions/internet-javascript-509/>

[URL6] Bootstrap, <https://whatis.techtarget.com/fr/definition/Bootstrap>

[URL8] Git, <https://openclassrooms.com/fr/courses/1233741-gerez-vos-codes-source-avec-git>

[URL9] Firebase,

<https://console.firebase.google.com/u/1/project/smartlibrary1/database/smart-library1/data>