

Sommaire

Introduction générale	1
Chapitre I : Présentation de l'entreprise.....	2
Introduction.....	2
1. Fiche d'identité de l'entreprise	2
2. Historique	3
3. Activité de l'organisme.....	3
Conclusion	3
Chapitre II: Analyse et spécification de besoin.....	4
Introduction.....	4
1. Présentation de sujet.....	4
2. Spécification de besoin	5
2.1 Besoins fonctionnels.....	5
2.2 Besoins non fonctionnels.....	5
3. Diagrammes de cas d'utilisation.....	6
3.1 Présentation des Acteurs	6
3.2 Description des cas d'utilisation.....	6
3.3 Présentation de diagramme de cas d'utilisation	7
3.3.1 Diagramme de cas d'utilisation de l'acteur « Visiteur ».....	7
3.3.2 Diagramme de cas d'utilisation de l'acteur « Utilisateur »	8
Conclusion	13
Chapitre III : Conception	14
Introduction.....	14
1. Diagramme de classes.....	14
1.1 Présentation de diagramme de classes.....	14
2. Diagramme de séquence	15
2.1 Diagramme de séquence relatif à l'authentification	15
2.2 Diagramme de séquence relatif à la création d'un formulaire	16
2.3 Diagramme de séquence relatif à « voir les formulaires soumis ».....	17
2.4 Diagramme de séquence relatif à «voir les réponses d'un formulaire»	18
Conclusion	18
Chapitre IV Réalisation.....	19

Introduction	19
1. Environnement de développement	19
1.1 Environnement matériel	20
1.2 Environnement logiciel	20
1.2.1 AGL : Power AMC.....	20
1.2.2 SGBD: MYSQL	20
1.2.3 Serveur d'application GlassFish 4.0.....	20
1.2.4 IDE: Eclipse Luna	20
1.3 Frameworks et librairies	21
1.3.1 JSF (Java Server Faces).....	21
1.3.2 Primefaces	21
1.3.3 JPA (Java Persistence API)	21
2 Le concept MVC	21
3 Le diagramme de déploiement	23
4 Les interfaces	24
Conclusion générale	29
Bibliographie et Neto graphie	30
1. Bibliographie	30
2. Neto graphie	30

Liste des figures

Figure 1: Logo de la société.....	2
Figure 2 : Les Acteurs.....	6
Figure 3 : Diagramme de cas d'utilisation de l'acteur Visiteur.....	7
Figure 4 : Diagramme de cas d'utilisation de l'acteur Utilisateur	8
Figure 5 : Diagramme de classes.....	14
Figure 6 : Diagramme de séquence "Authentification"	15
Figure 7 : Diagramme de séquence "Création d'un Formulaire"	16
Figure 8 : Diagramme de séquence " voir les formulaires soumis "	17
Figure 9 : Diagramme de séquence " voir les réponses d'un formulaire "	18
Figure 10 : Architecture logicielle de l'application	19
Figure 11 : Modèle MVC	22
Figure 12 : Diagramme de déploiement	23
Figure 13 : Interface d'authentification.....	24
Figure 14 : Interface Principal.....	25
Figure 15 : Interface de recherche des formulaires.....	25
Figure 16 : Interface de création d'un formulaire.....	26
Figure 17 : Interface de gestion de formulaire.....	26
Figure 18 : Interface d'affichage de formulaires soumis.....	27
Figure 19 : Interface d'affichage des Statistiques général d'un formulaire.....	27
Figure 20 : Interface d'affichage des Statistiques d'un formulaire par question	28

Introduction générale

Le stage de perfectionnement est une occasion exceptionnel pour l'étudiant afin d'exercer la côté pratique profondément. Ceci va nous permettre d'enrichir nos connaissances, d'appliquer ces théories à la réalité et de se confronter au domaine réel de travail.

Notre projet consiste a créé une application web dynamique permettant la création des formulaires, et des questionnaires pour recueillir des informations.

Pour réaliser notre projet on va suivre la démarche suivante :

- **Chapitre 1** : ce chapitre est consacré à la présentation de la société **Rayen Soft**,
- **Chapitre 2** : ce chapitre est une prise de connaissance pour mieux définir les besoins et les fonctions de notre application, ainsi que pour collecter, analyser et définir les besoins de haut niveau (les besoin fonctionnels et les besoin non fonctionnels),
- **Chapitre 3** : Conception, dans ce chapitre sera consacrée à la conception de l'application : il s'agit d'une phase de modélisation théorique de l'application,
- **Chapitre 4**: Réalisation, ce chapitre contient une description détaillée des outils utilisés pour développer l'application web, l'architecture du système, le modèle de déploiement et des interfaces de l'application.

Chapitre I : Présentation de l'entreprise

Introduction

Ce chapitre présente l'organisme d'accueil, dans lequel le projet a été effectué, en spécifiant son identité et en évoquant ses activités.

1. Fiche d'identité de l'entreprise

- **Raison sociale** : Rayen Soft
- **Statut juridique** : Société Unipersonnelle À Responsabilité Limitée (SARL)
- **Adresse** : Avenue Habib Bourguiba, 1152 Hammem Zriba, Zaghouan, Tunisie
- **Tel** : 72667035
- **Email** : info@rayensoft.com
- **Site Web** : www.rayensoft.com
- **Nombre d'employé** : 10
- **Responsable** : 1



Figure 1: Logo de la société

2. Historique

Rayen soft est une société, créée le 1 octobre 2010 par Mr Housseem Bdoui spécialisé dans le développement de contenu web et des solutions numériques.

Elle est une société moderne de développement de logiciels basée en Tunisie, avec une grande expérience dans le développement des applications de bureau, applications mobiles et sites web, ainsi que le marketing en ligne.

3. Activité de l'organisme

Rayen Soft a été créé pour répondre aux attentes des entreprises et des particuliers en matière de :

➤ Création de sites web

La société Rayen Soft spécialise dans la création des sites et des applications web ainsi que toute solution web : intranet, microblogging, etc.

➤ Développement Mobile

Rayen Soft vous offre des services pour le développement d'applications mobiles destinées aux utilisateurs Ios et Android.

➤ Web-Marketing

Le web marketing est une vaste discipline qui comprend plusieurs types d'activités Internet. Rayen Soft propose une gamme de services complets pour augmenter votre visibilité sur le web.

➤ Maintenance et évolution

Si vous avez besoin de consultation informatique ou bien si vous désirez optimiser la structure informatique de votre entreprise, Rayen Soft met à la position de ces clients ce service.

➤ Design et impression

Raye soft réalise des chartes graphiques, logo, plaquette d'entreprise, affiche, carte de visite, flyer, brochure, stand, catalogue, illustrations, animations.

Conclusion

Dans ce chapitre, nous avons pu présenter la société et identifier les services de ce dernier.

Chapitre II: Analyse et spécification de besoin

Introduction

Dans ce chapitre nous précisons le cadre du projet tout en effectuant une étude fonctionnelle du système. Pour ce faire, nous présenterons le sujet et ses objectifs et nous finissons par dégager les besoins des différents utilisateurs que le système doit accomplir avec une présentation du diagramme des cas d'utilisation.

1. Présentation de sujet

Dans le cadre de l'application réalisé nous avons décidé de développer un site web dynamique « Dyna Form » qui a pour l'objectif d'offrir à l'utilisateur un espace pour créer des formulaires, réaliser des enquêtes ou questionnaires pour recueillir des informations.

Le site offre aussi un moyen de recherche des formulaires crée par d'autre utilisateurs afin de trouver le formulaire souhaité pour soumettre ses réponses.

Le site possède un espace de création de formulaire dynamique : l'utilisateur peut ajouter plusieurs questions sous différent type de réponse (zone de texte, paragraphe, date, des boutons radios, etc.)

Lorsque l'utilisateur crée un formulaire, il peut le modifier, voir les réponses et consulter les statistiques.

Pour chaque formulaire, le site fournit un module de statistiques avec quelques graphiques qui fournit des informations sur :

- Les nombres de réponses totales.
- Les nombres de réponses par genre.
- Le Moyen âge.
- Le résumé des réponses par questions.

2. Spécification de besoin

Cette partie va servir à poser les bases du recueil des besoins du système à réaliser. Pour pouvoir clarifier les besoins des utilisateurs de mon application, nous venons présenter les besoins fonctionnels ainsi que les besoins non fonctionnels.

2.1 Besoins fonctionnels

- Recherche de formulaires.
- Création de formulaire.
- Gestion de formulaire :
 - Modifier formulaire.
 - Supprimer formulaire.
 - Afficher les réponses.
 - Accéder à la statistique du formulaire.

2.2 Besoins non fonctionnels

L'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

L'application doit aussi assurer ces besoins :

- Performance : l'application traite souvent un grand volume de données, elle doit alors effectuer ces traitements d'une façon rapide et optimale afin d'éviter une longue durée d'attente.
- Robustesse : l'application doit pouvoir gérer un très grand volume de données sans se bloquer.
- Portabilité : l'application doit être facilement utilisable et ne doit pas nécessiter des étapes de configuration ou d'installation.
- L'ergonomie de l'interface : Les interfaces doivent être simple et conviviale : On doit essayer le maximum d'éliminer l'encombrement.

3. Diagrammes de cas d'utilisation

Le diagramme des cas d'utilisation est un diagramme dynamique orienté utilisateur qui donne une vision globale ou détaillée des fonctionnalités offertes par le système et leurs interactions avec les utilisateurs.

3.1 Présentation des Acteurs

Un acteur désigne le rôle joué par une personne ou une entité externe qui interagit avec le site. En observant les utilisateurs externes du site nous pouvons déterminer les acteurs qui sont responsables de son exploitation. Pour cela nous avons identifié deux acteurs :

- Visiteur : c'est lui qui va s'inscrire au site et qui effectue des recherches des formulaires.
- Utilisateur : il s'agit du profil de la personne qui va accéder aux services du site.

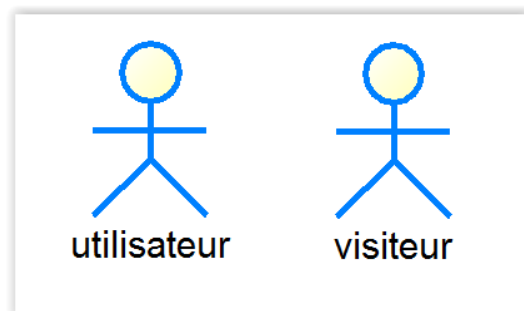


Figure 2 : Les Acteurs

3.2 Description des cas d'utilisation

Dans notre système, nous allons classer les cas d'utilisations selon leurs acteurs.

- **Visiteur**
 - ✓ S'inscrire.
 - ✓ Rechercher des formulaires.

▪ Utilisateur

- ✓ Créer des formulaires
- ✓ Consulter le compte
- ✓ Récupérer son compte
- ✓ Rechercher des formulaires
- ✓ Remplir des formulaires
- ✓ Voir les réponses des formulaires
- ✓ Gérer ces formulaires
- ✓ Gérer son compte

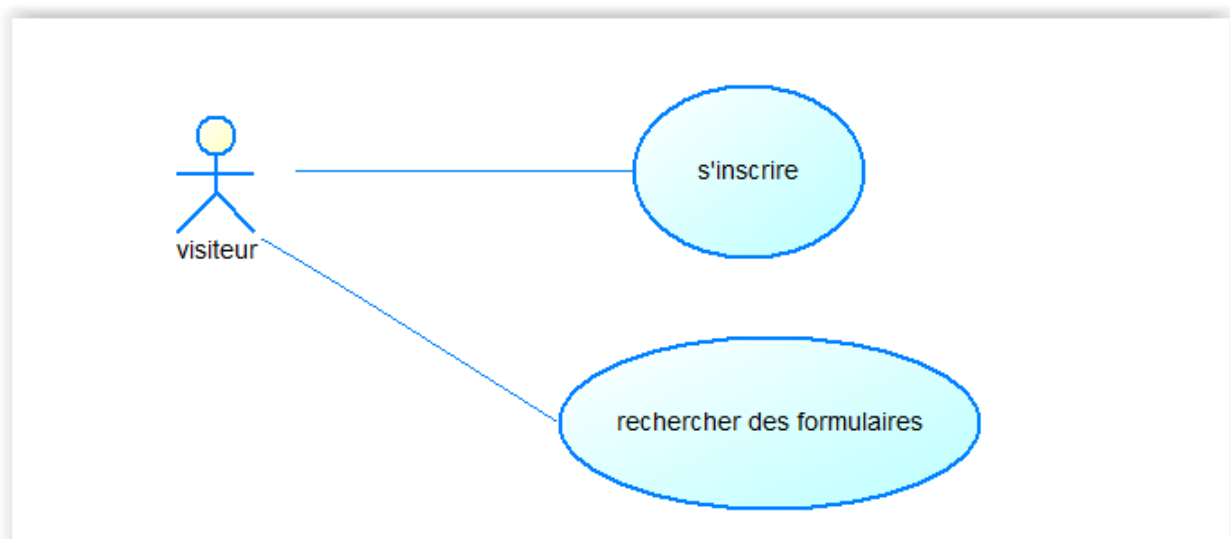
3.3 Présentation de diagramme de cas d'utilisation**3.3.1 Diagramme de cas d'utilisation de l'acteur « Visiteur »**

Figure 3 : Diagramme de cas d'utilisation de l'acteur Visiteur

3.3.2 Diagramme de cas d'utilisation de l'acteur « Utilisateur »

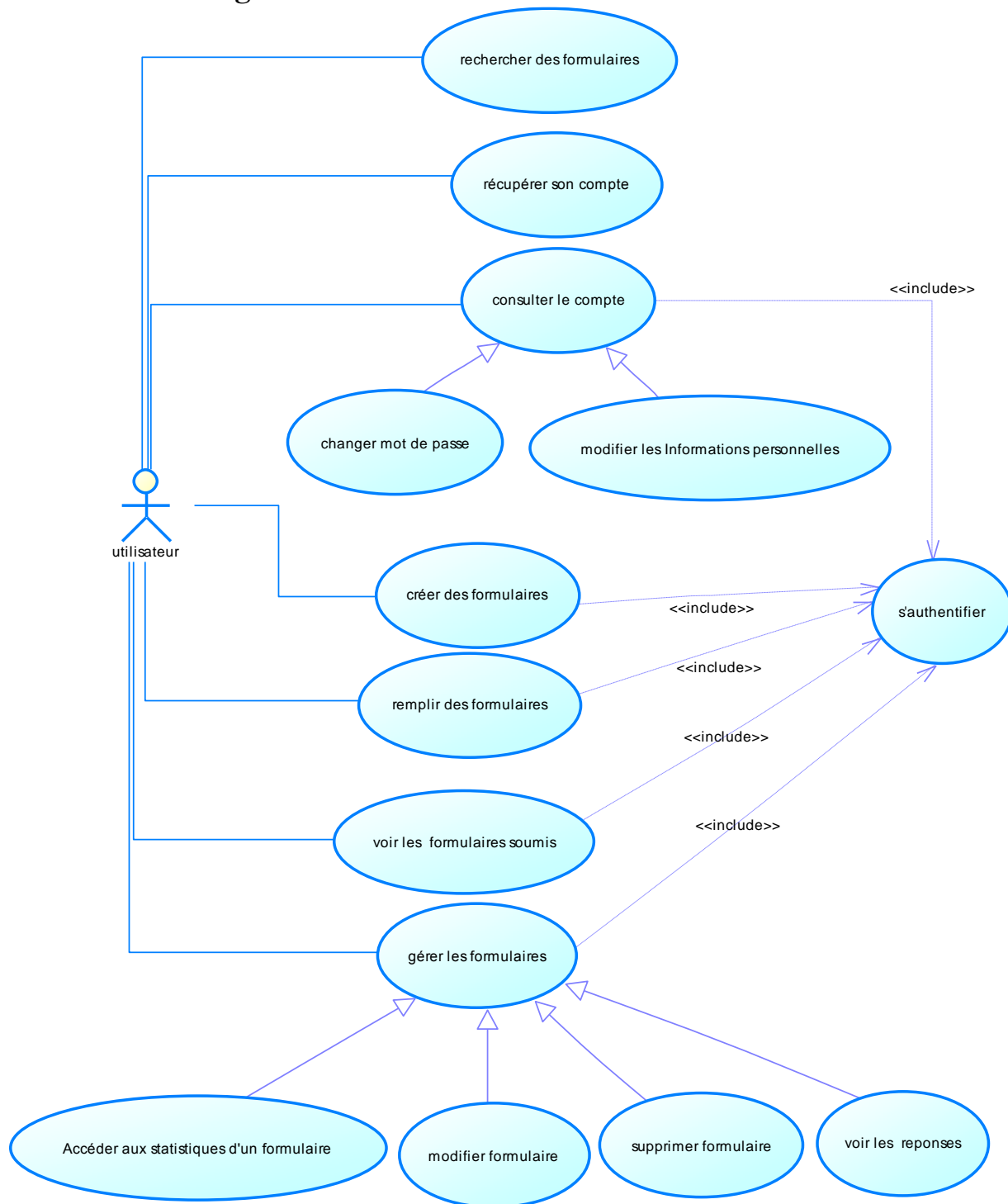


Figure 4 : Diagramme de cas d'utilisation de l'acteur Utilisateur

➤ **Description textuelle du cas « s'authentifier »**

Acteur : Utilisateur

But : S'authentifier au site

Pré condition : l'utilisateur doit être existant dans la base de données

Scénario nominal :

Acteur	Système
1. L'utilisateur saisie son login et mot de passe.	
	2. Le système vérifie l'existence et la conformité de login et mot de passe .
	3. Accéder à l'espace privé d'utilisateur.
Enchaînement alternatif	
E1 : login et mot de passe non valide	
1. Le système affiche un message d'erreur.	
2. Le système reprend au point 1.	

Post Condition : Accès à son espace privé

➤ **Description textuelle du cas « créer des formulaires »**

Acteur : Utilisateur

But : créer un formulaire

Pré condition : l'utilisateur doit être authentifié

Scénario nominal :

Acteur	Système
1. L'utilisateur demande la création d'un nouveau formulaire	
	2. Le système affiche l'interface de création d'un formulaire.
3. L'utilisateur saisie le nom de son formulaire.	
4. L'utilisateur change éventuellement l'image de formulaire par défaut.	
5. L'utilisateur peut ajouter des questions et choisit le type de l'élément graphique présentant chaque question.	
6. L'utilisateur demande l'enregistrement du formulaire dans le système	
	7. Le système enregistre le formulaire dans la base de données.
Enchaînement alternatif	
E1 : Le formulaire ne contient aucune question	
1. Le système affiche un message d'erreur.	
2. Le système reprend au point 5.	

Post Condition : Formulaire créé

Description textuelle du cas « modifier formulaire »**Acteur :** Utilisateur**But :** modifier un formulaire après la création**Pré condition :** l'utilisateur doit être authentifié**Scénario nominal :**

Acteur	Système
1. L'utilisateur demande l'accès à l'interface gestion des formulaires	
	2. Le Système affiche l'espace de gestion des formulaires.
3. L'utilisateur choisit le formulaire approprié pour le modifier.	
	4. Le système affiche l'interface de modification du formulaire.
5. L'utilisateur peut modifier le nom de formulaire ajouter ou supprimer des questions.	
6. L'utilisateur demande l'enregistrement du formulaire dans le système	
	7. Le système enregistre le formulaire modifié.
Enchaînement alternatif	
E1 : L'utilisateur a supprimé tous les questions.	
3. Le système affiche un message d'erreur.	
4. Le système reprend au point 4.	

Post Condition : Formulaire modifié

➤ **Description textuelle du cas « supprimer formulaire »**

Acteur : Utilisateur

But : supprimer un formulaire

Pré condition : l'utilisateur doit être authentifié

Scénario nominal :

Acteur	Système
1. L'utilisateur demande l'accès à l'interface de gestion des formulaires	
	2. Le Système affiche l'espace de gestion des formulaires.
3. L'utilisateur choisit le formulaire approprié et clique sur « supprimer ».	
	4. Le système demande la confirmation de suppression.
5. L'utilisateur confirme la suppression.	
	6. Le système supprime le formulaire dans la base de donné.

Post Condition : Formulaire supprimé

➤ **Description textuelle du cas « voir les formulaires soumis »**

Acteur : Utilisateur

But : l'utilisateur peut consulter les formulaires qu'il a soumis

Pré condition : l'utilisateur doit être authentifié

Scénario nominal :

Acteur	Système
1. L'utilisateur demande l'accès à ces formulaires soumis	
	2. Le système accède à la page d'affichage de formulaires soumis
	3. Le système affiche la liste des formulaires avec le nom, le créateur et la date de soumission,
4. L'utilisateur sélectionne un formulaire pour visionner ses détails	
	5. Le système affiche les questions et les réponses du formulaire que l'utilisateur a choisi.

Conclusion

Dans ce chapitre, nous avons présenté une description du projet en énumérant ces objectifs et nous avons dégagé les besoins fonctionnels et non fonctionnels qui ont permis de mieux expliciter le système à réaliser.

Chapitre III : Conception

Introduction

La phase de la conception est la phase initiale de la création et de la mise en œuvre de notre projet. En fait, elle représente une étape importante de réflexion dans le cycle de développement logiciel après la phase de l'analyse et de spécification.

Dans ce chapitre nous abordons la partie conception du projet, dans laquelle, nous détaillons les différents éléments de conception, à savoir les diagrammes de séquences, et les diagrammes de classes.

1. Diagramme de classes

Le diagramme de classes est un diagramme statique qui permet de fournir une représentation des classes intervenant dans le système et leurs relations indépendamment d'un langage de programmation particulier.

1.1 Présentation de diagramme de classes

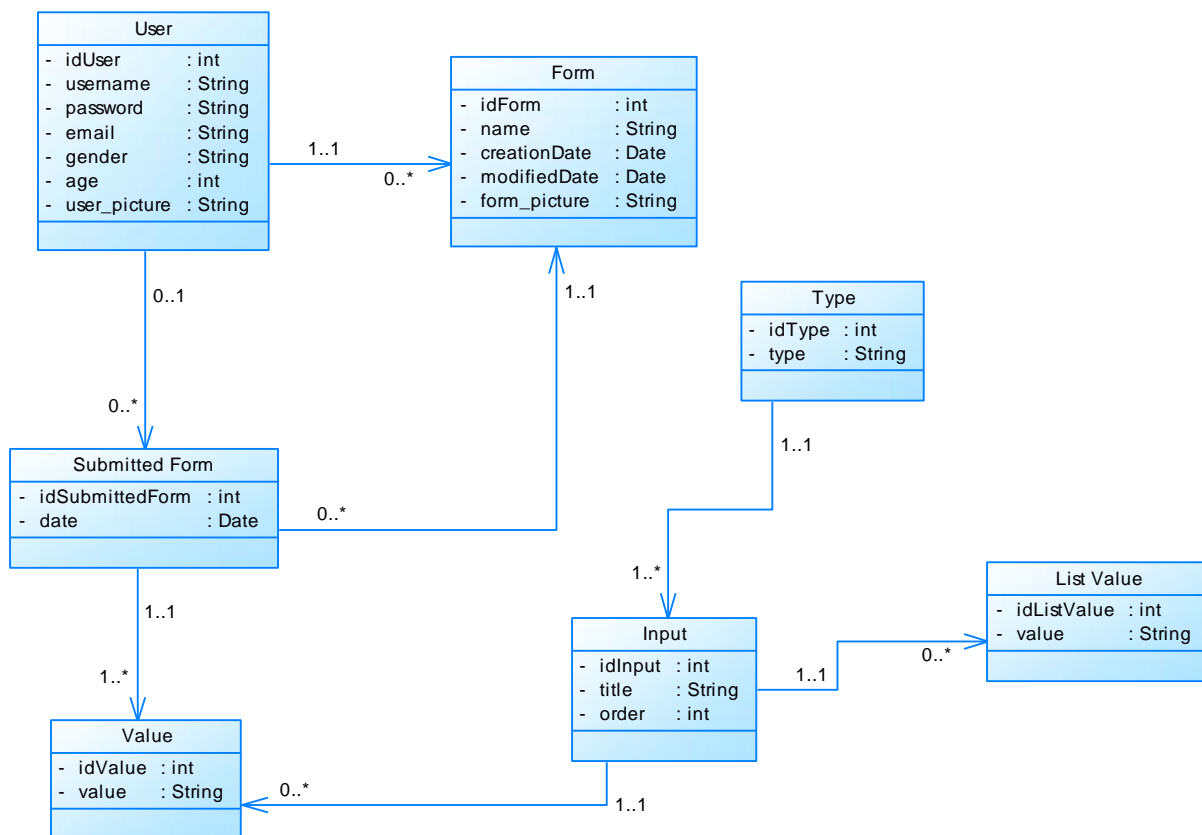


Figure 5 : Diagramme de classes

2. Diagramme de séquence

Le diagramme de séquence décrit l'aspect dynamique du système. Il modélise les interactions entre les objets ou entre utilisateur et objet, en mettant l'accent sur la chronologie des messages échangés.

2.1 Diagramme de séquence relatif à l'authentification

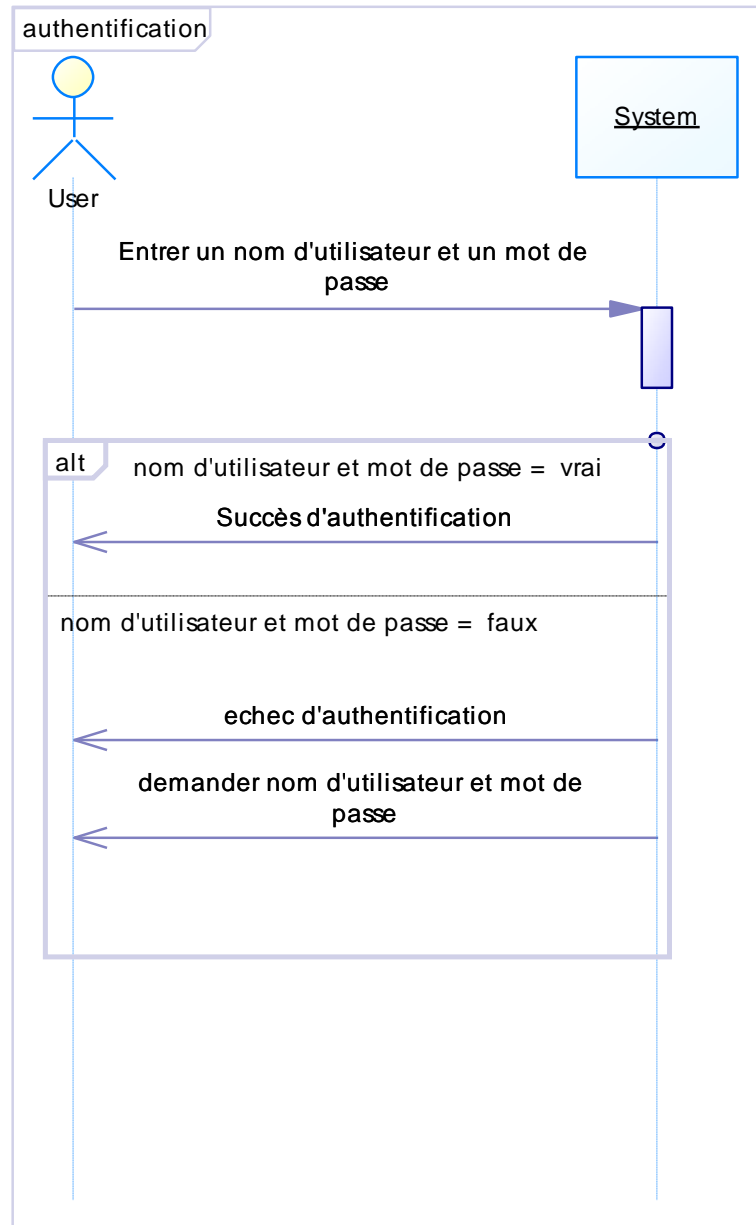


Figure 6 : Diagramme de séquence "Authentication"

2.2 Diagramme de séquence relatif à la création d'un formulaire

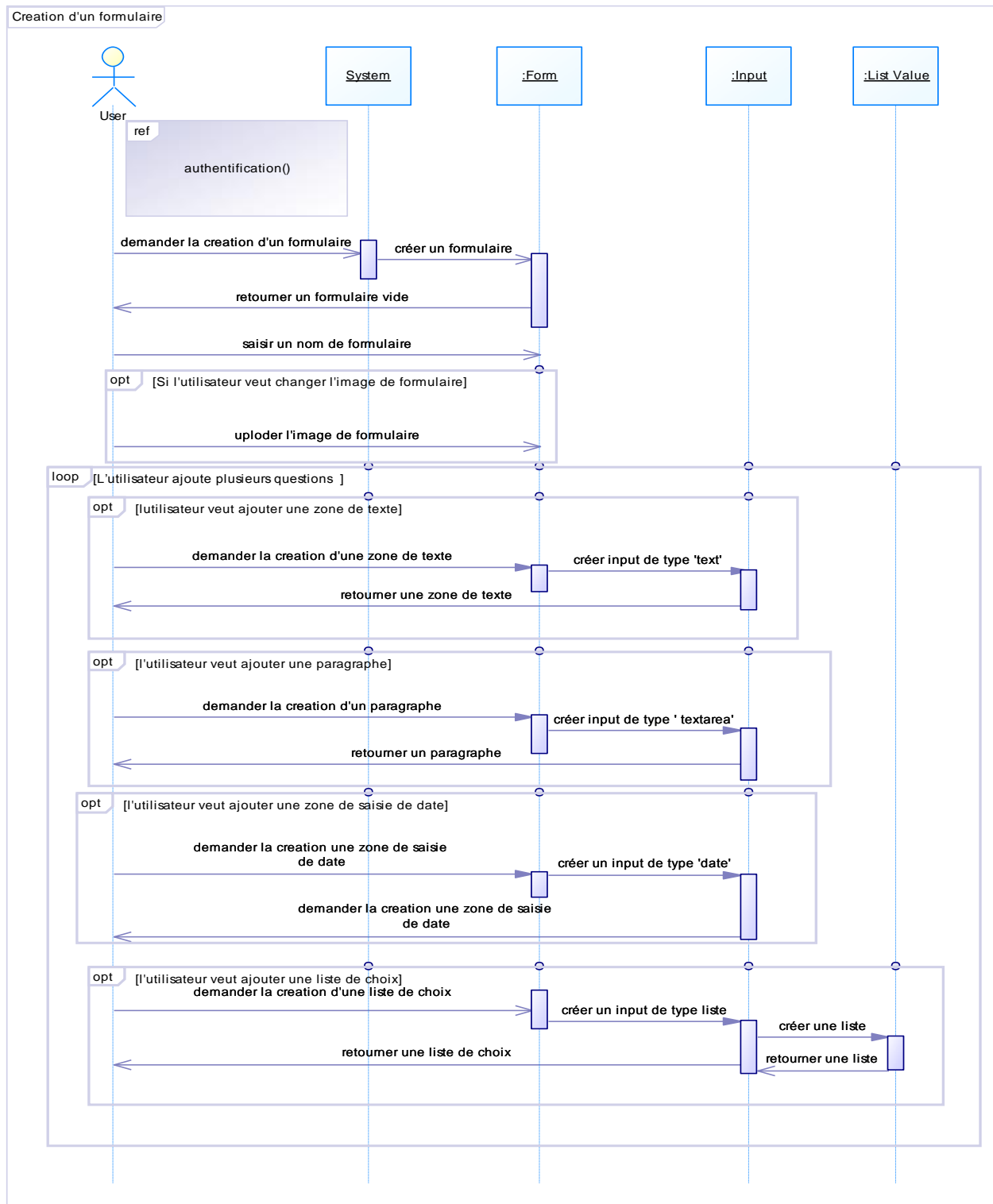


Figure 7 : Diagramme de séquence "Création d'un Formulaire"

2.3 Diagramme de séquence relatif à « voir les formulaires soumis »

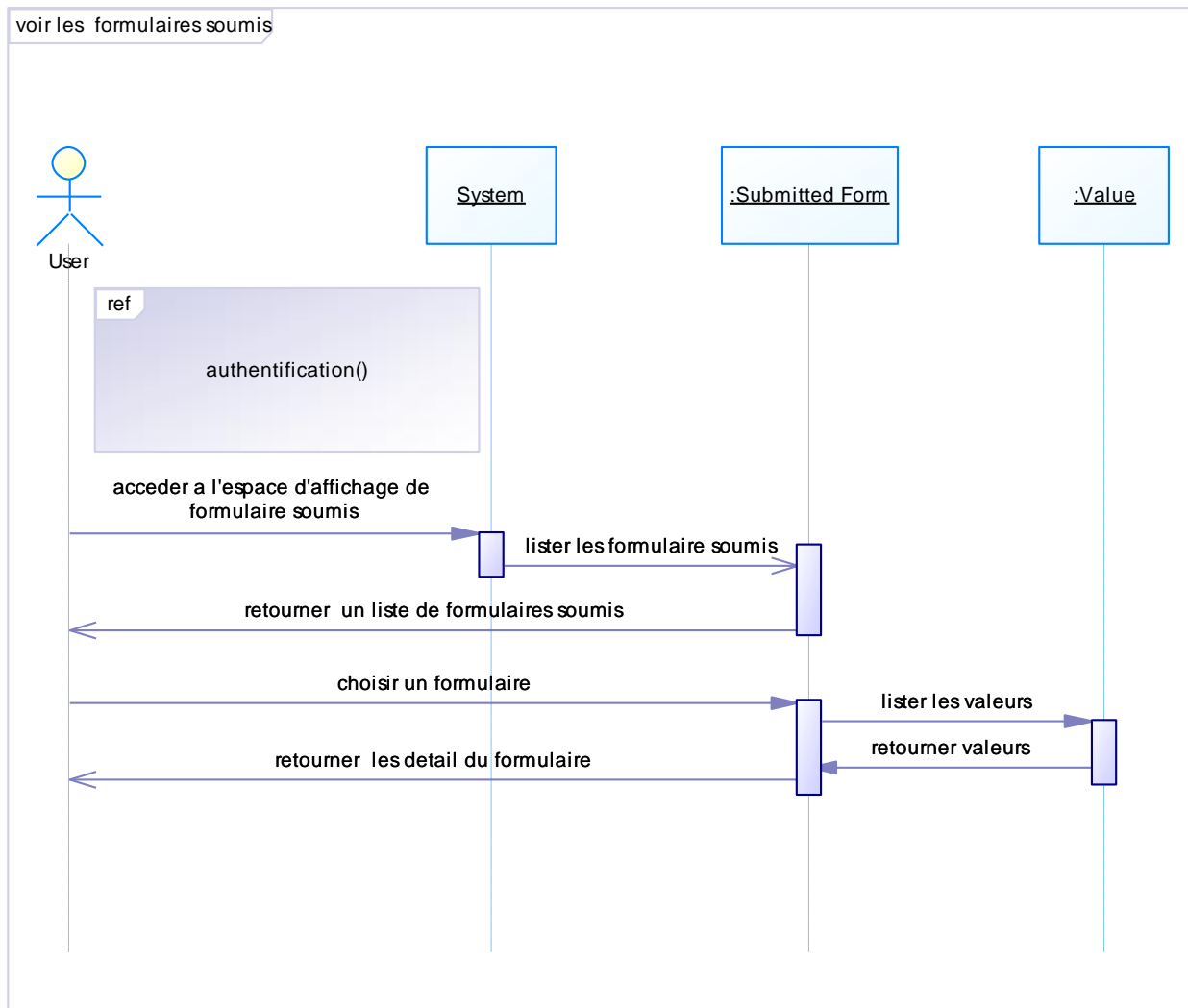


Figure 8 : Diagramme de séquence " voir les formulaires soumis "

2.4 Diagramme de séquence relatif à «voir les réponses d'un formulaire»

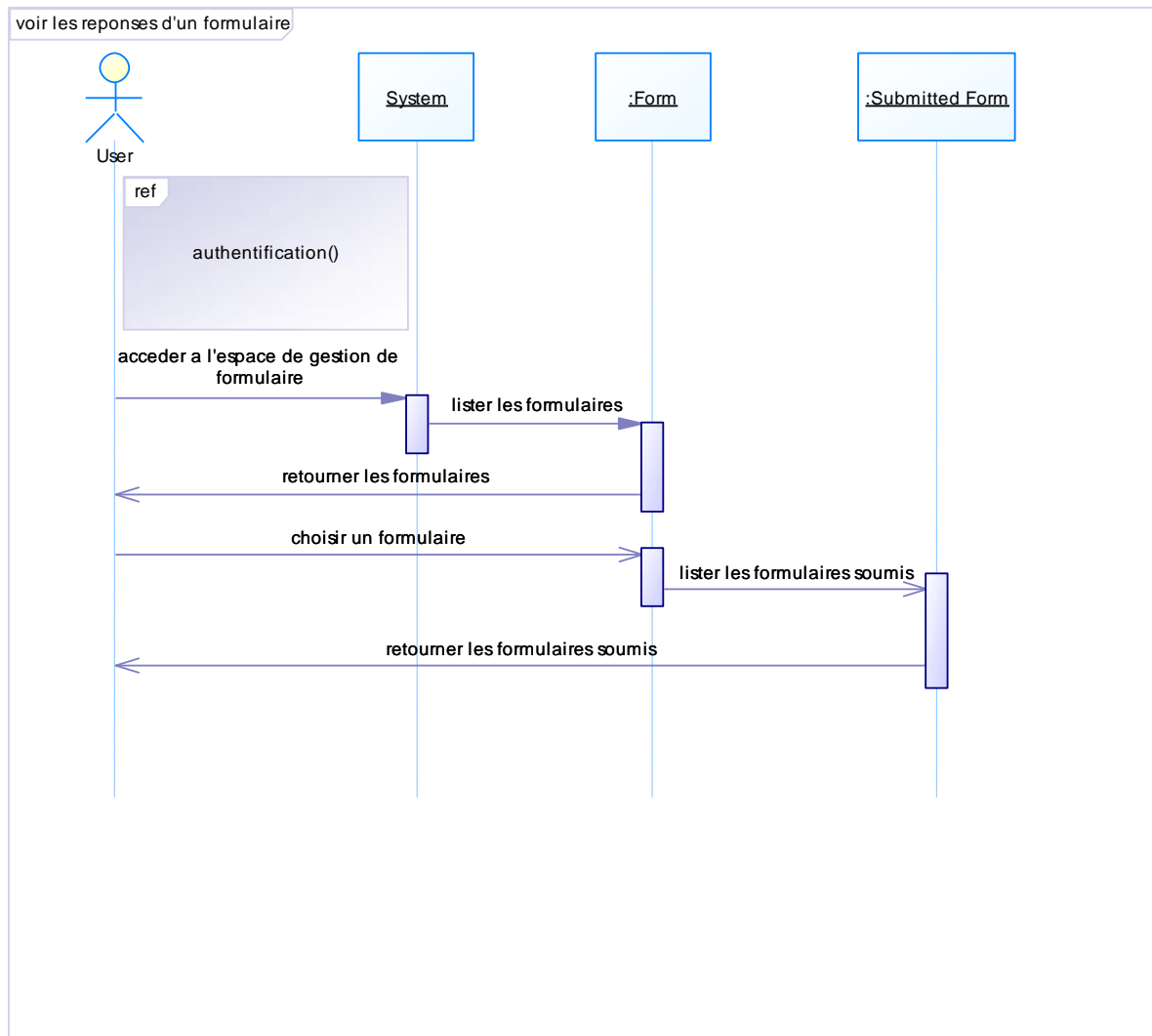


Figure 9 : Diagramme de séquence " voir les réponses d'un formulaire "

Conclusion

Dans ce chapitre nous avons détaillé les différentes vues conceptuelles des applications à réaliser à travers les diagrammes de séquence et de classe.

Dans le chapitre suivant, nous allons nous intéresser à l'implémentation de notre système.

Chapitre IV Réalisation

Introduction

Dans ce chapitre nous traitons les différentes étapes d'implémentation de l'application : nous allons commencer par la description de l'environnement matériel et logiciel tout en donnant par la suite un aperçu sur le travail accompli au cours de la période de développement.

1. Environnement de développement

Dans cette partie, nous avons étudié le choix des outils matériels et surtout les outils logiciels du développement web.

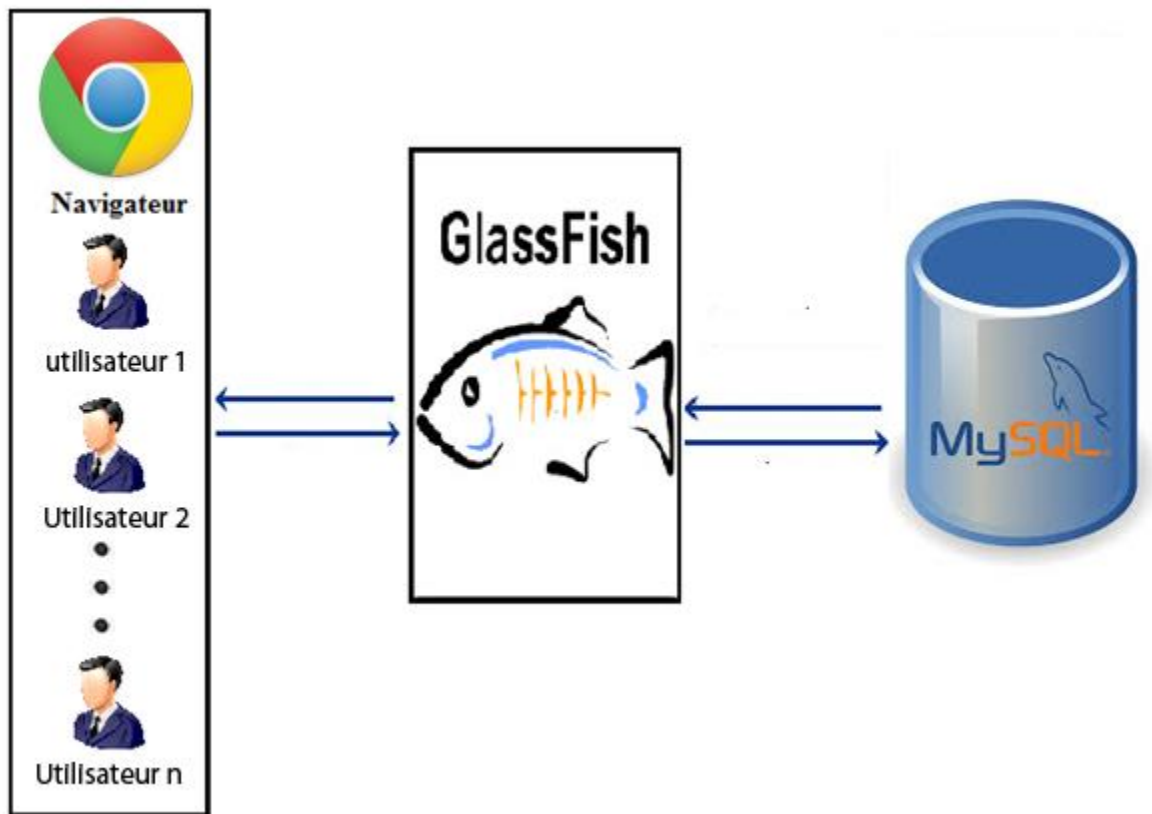


Figure 10 : Architecture logicielle de l'application

1.1 Environnement matériel

Durant la période de réalisation de ce projet, nous avons eu en disposition une machine personnelle avec les configurations suivantes :

Configurations machine:

- **Marque** : Lenovo Y510P
- **Système d'exploitation** : Microsoft Windows 7, Edition professionnel, 64bits
- **Mémoire et CPU** : RAM 8 Go et CPU 2.4 GHz

1.2 Environnement logiciel

Nous avons énuméré au cours de cette partie les différents outils utilisés tout au long de ce projet pour l'étude et la mise en place de notre application.

1.2.1 AGL : Power AMC

C'est une solution de modélisation des systèmes d'information qui supporte plusieurs techniques de modélisation standard : modélisation Merise (données et traitement), modélisation UML particulièrement adaptée à la logique des applications et modélisation des processus métiers dédiée aux non informaticiens pour leur faciliter l'expression des besoins.

1.2.2 SGBD: MYSQL

Oracle Data base est un système de gestion de base de données relationnel (SGBDR) qui implémente le langage de requête SQL

1.2.3 Serveur d'application GlassFish 4.0

GlassFish est un serveur d'applications Open Source. Il est destiné aux environnements de production, et est compatible avec Java EE : nous avons utilisé la version 4.0.

1.2.4 IDE: Eclipse Luna

Eclipse est un environnement de développement intégré (**IDE** : **I**ntegrated **D**evelopment **E**nvironment) dont le but est de fournir une plateforme modulaire pour permettre de réaliser des développements informatiques, tels que les sites web, les programmes java et d'autres.

1.3 Frameworks et librairies

1.3.1 JSF (Java Server Faces)

JSF (Java Server Faces) est un Framework permettant la création d'interfaces WEB, la mise en place du Design Pattern MVC : nous avons utilisé JSF2.2.

1.3.2 Primefaces

Primefaces est une librairie de composants JSF riches et ayant un support de la technologie Ajax. La version que nous avons utilisée est 5.1.

1.3.3 JPA (Java Persistence API)

La Java Persistence API (abrégée en JPA) est l'API standard pour la gestion de la persistance et de mapping objet- relationnel des données.

Le mapping objet-relationnel consiste à associer une ou plusieurs classes avec une table, et chaque attribut de la classe avec un champ de la table.

Puisque nous avons utilisé un langage de POO (Java) avec une base des données relationnelle, nous avons développé la couche d'accès aux données avec l'API JPA. La version utilisée est 2.1.

2 Le concept MVC

Pour le développement, nous avons appliqué le modèle **Model-View-Controller (MVC)**

Ce modèle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application.

Ce patron de conception (design pattern) impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur.

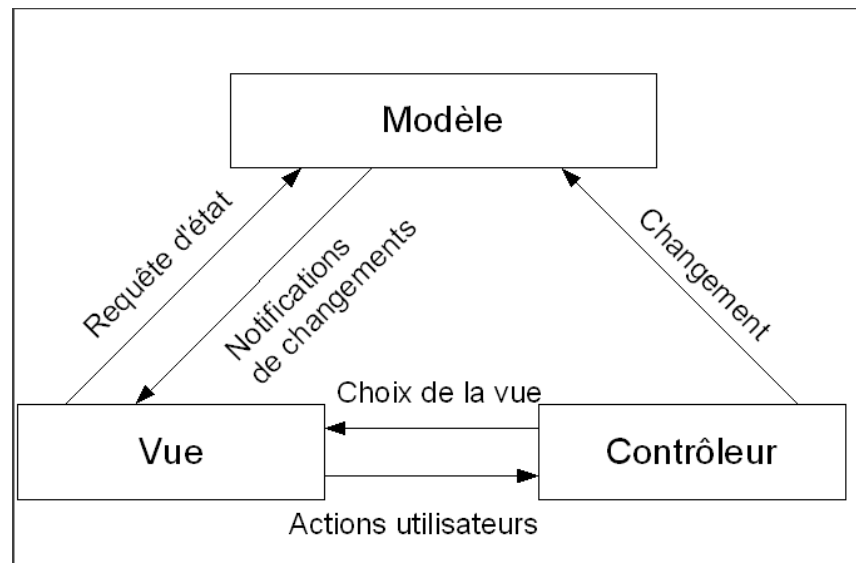


Figure 11 : Modèle MVC

- **Le modèle** représente le comportement de l'application : traitements des données, interactions avec la base de données, etc. Il décrit ou contient les données manipulées par l'application. Le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur).

Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation.

- **La vue** correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, etc.). Ces différents événements sont envoyés au contrôleur.
- **Le contrôleur** prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer.

Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle, et ce dernier notifie la vue que les données ont changée pour qu'elle les mette à jour.

3 Le diagramme de déploiement

Ce diagramme décrit l'organisation physique des ressources matérielles qui composent le système et montre la distribution des composants sur ces matériels.

Chaque ressource étant représentée par un nœud, le diagramme de déploiement précise comment les composants sont distribués sur les nœuds et quelles sont les associations entre les composants ou les nœuds.

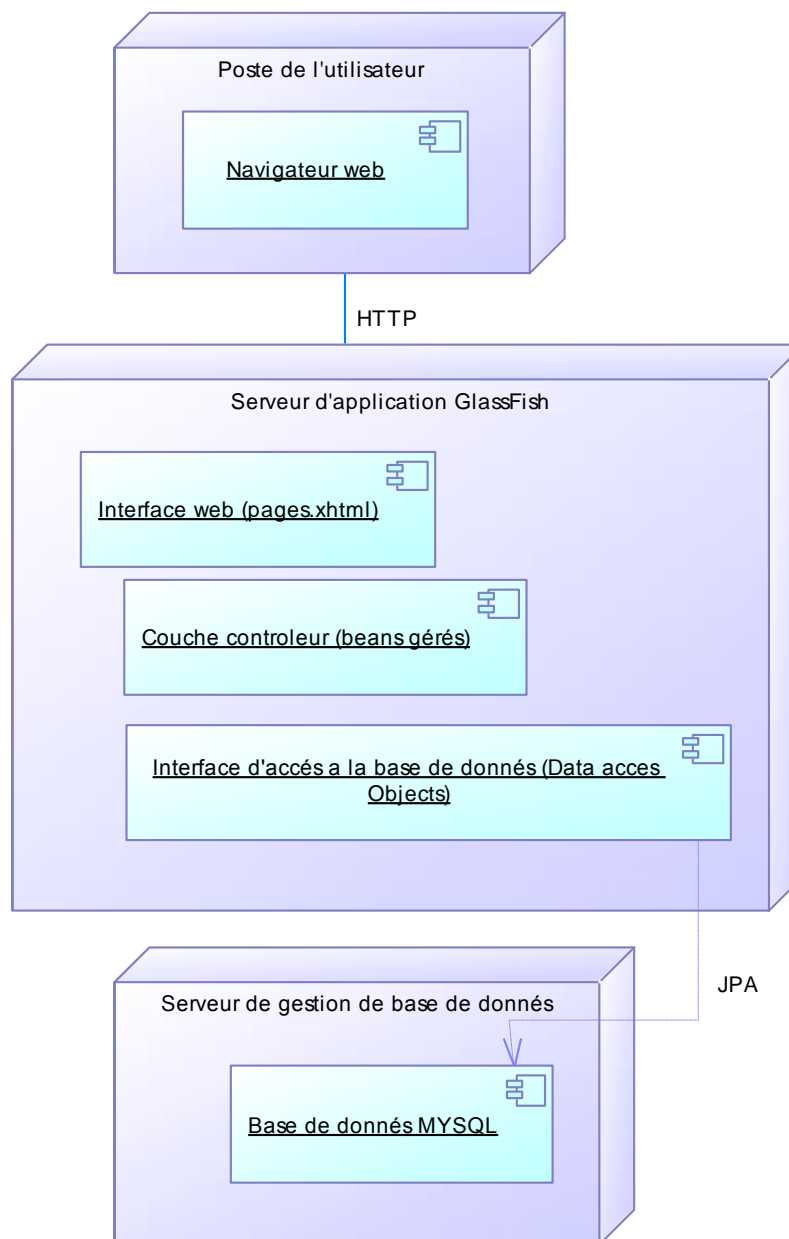
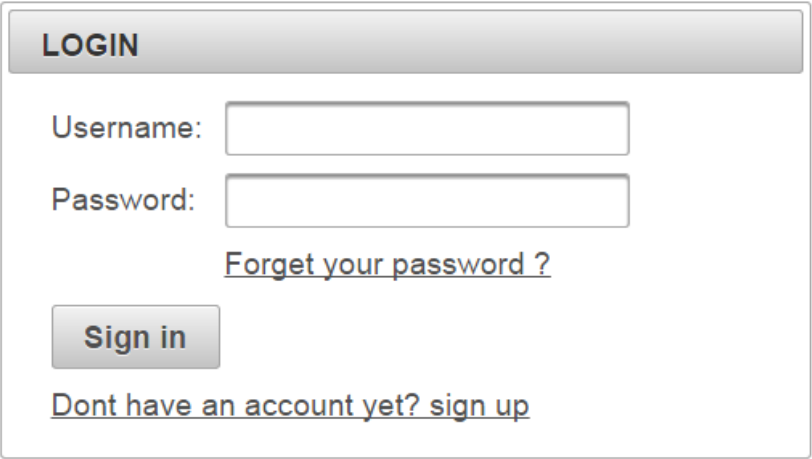


Figure 12 : Diagramme de déploiement

4 Les interfaces

L'interface graphique est une partie très importante pour la réalisation d'une application Web convenable et conviviale offrant un certain plaisir à l'utilisateur lors de sa navigation. Ainsi ce critère peut faire la différence entre une application et une autre bien qu'elles aient les mêmes fonctionnalités.

Voici maintenant un ensemble de captures d'écrans sur les principaux points d'entrées de l'application:



The image shows a login form with the following elements:

- A header box labeled **LOGIN**.
- A label "Username:" followed by a text input field.
- A label "Password:" followed by a text input field.
- A link: [Forget your password ?](#)
- A button labeled **Sign in**.
- A link: [Dont have an account yet? sign up](#)

Figure 13 : Interface d'authentification

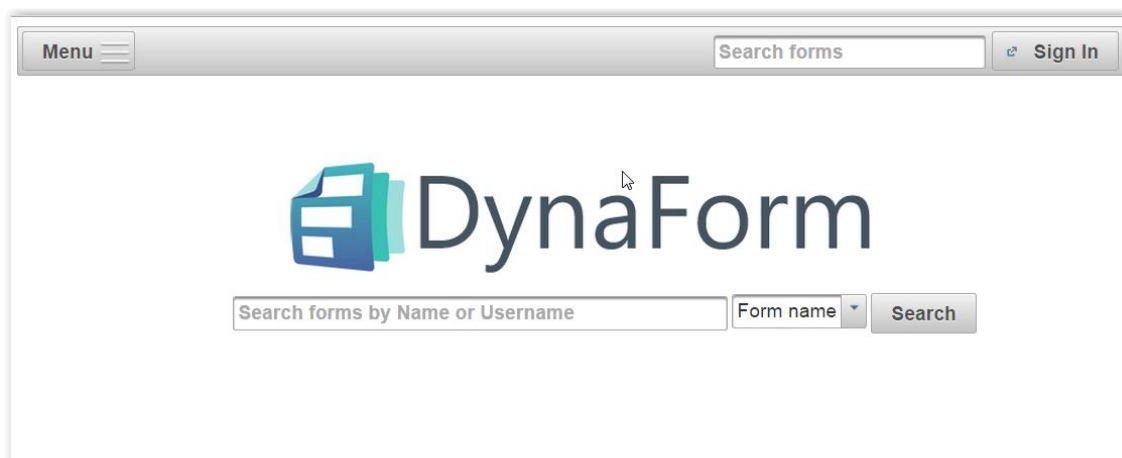


Figure 14 : Interface Principal

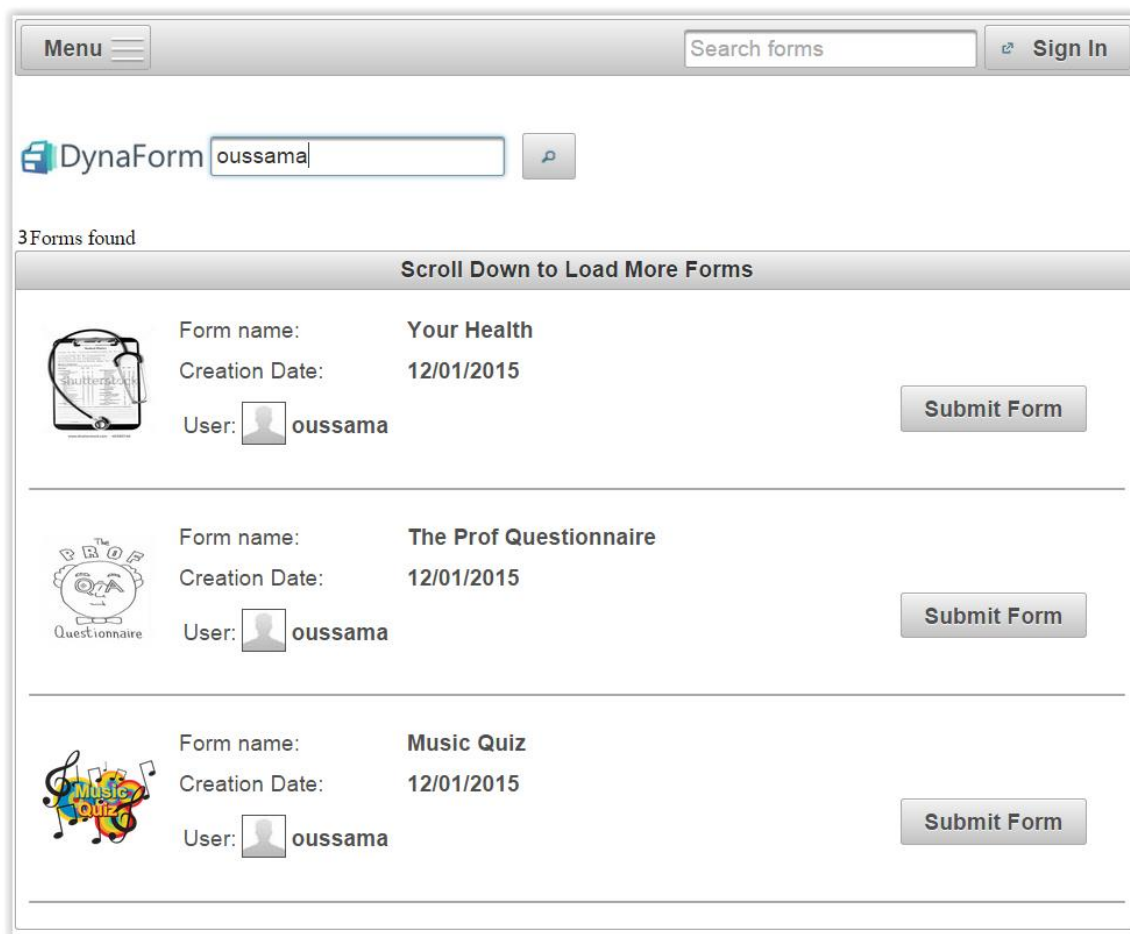


Figure 15 : Interface de recherche des formulaires

Figure 16 : Interface de création d'un formulaire

Forms			
form	creation date	last modification date	Actions
Sport Form	19-02-2015	20-02-2015	
My Form	15-01-2015	15-01-2015	
Your Skills	06-01-2015	07-01-2015	
Doctors Form	01-01-2015	06-01-2015	

Figure 17 : Interface de gestion des formulaires

Menu
Search forms
oussama











Submitted Forms			
Form	Created by	Submission date	Action
 Djob quizz	Ahmed	07/02/2015	
 Sport Q&A	Youssef	08/02/2015	
 Game skills	MTLC	11/02/2015	
 Movies	User789	12/02/2015	
 MyForm	User16	16/02/2015	

Figure 18 : Interface d'affichage de formulaires soumis

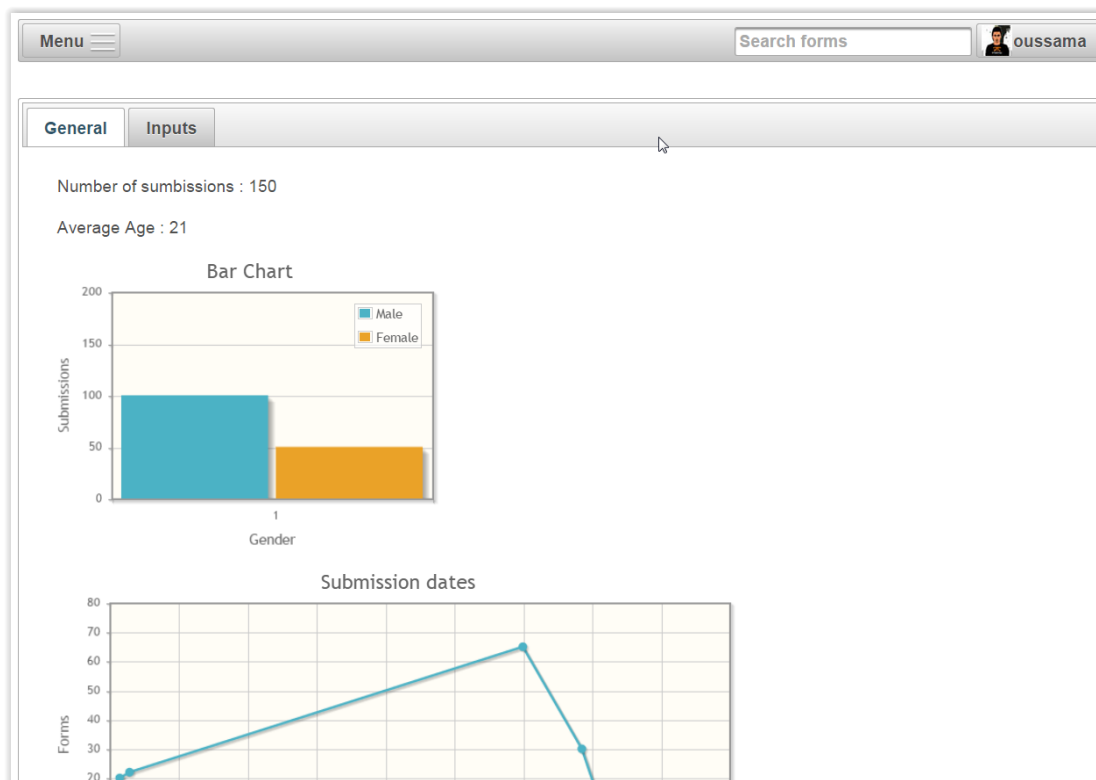


Figure 19 : Interface d'affichage des Statistiques général d'un formulaire

Menu
Search forms
oussama

General
Inputs

whats your favorite movies ?

1
2

oussama reguez
ahmed fateh
karim jlassi
mounir mehrez

describe yourself ?

1
2
3
4
5
6
7
8
9
10

im 20 years old i live with my parent , we have a big house near the sea ,i like playing sports like footibal,tennis

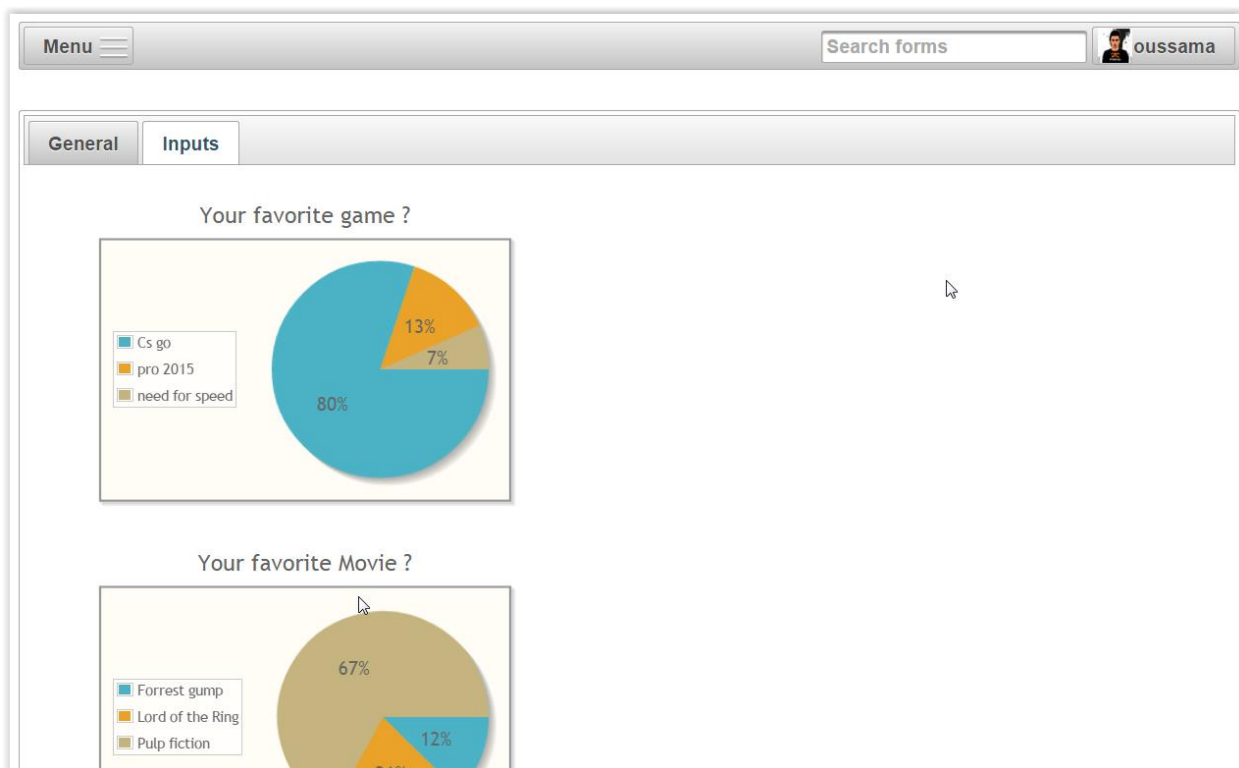


Figure 20 : Interface d'affichage des Statistiques d'un formulaire par question

Conclusion générale

Au cours de ce projet, nous avons mis en place une application pour **la création et la gestion des formulaires dynamiques** qui nous a présenté une opportunité pour s'améliorer en langage de programmation Java/JEE l'un des langages les plus puissants de notre époque, communication client/serveur, conception et interaction avec les bases de données, aussi, il nous a permis d'acquérir des connaissances importants concernant la librairie des composants PrimeFaces et la conception UML.

Le travail présenté ne représente pas la version finale qu'on peut atteindre. En effet, plusieurs fonctionnalités peuvent être ajoutées à notre application telle que l'intégration d'une partie mobile

Enfin, nous espérons que notre projet répond aux attentes des utilisateurs et de mes enseignants

Bibliographie et Netographie

1. Bibliographie

- PrimeFaces Cookbook January 2013 Packtpub, **Mert Çalışkan, Oleg Varaksin,**
- PrimeFaces User's Guide 5.0 Second Edition, **Çağatay Çivici,**
- Beginning Java EE7 First Edition June 2013 Apress, **Goncalves Antonio,**
- Créer des applications web avec java EE, **Médéric Munier**

2. Netographie

- <http://openclassrooms.com/>
- <http://primefaces.org/>
- <http://stackoverflow.com/>
- <http://www.developpez.com/>