

REPUBLIQUE DU CAMEROUN

TALISATION DES ACTES DE MARIAGE

REPUBLIC OF CAMEROON

Paix – Travail – Patrie

\*\*\*\*\*

MINISTERE DE  
L'ENSEIGNEMENT SUPERIEUR

\*\*\*\*\*

IAI-CAMEROUN – CENTRE DE  
DOUALA

Peace – Work – Fatherland

\*\*\*\*\*

MINISTRY OF HIGHER  
EDUCATION

\*\*\*\*\*

AICS – CAMEROON – DOUALA  
CENTER



Etablissement Inter – Etats d'Enseignement Supérieur

CENTRE D'EXCELLENCE TECHNOLOGIQUE PAUL BIYA

BP 13 719 Yaoundé (Cameroun) Tel. (+237) 242 72 99 57/ (+237) 242 72 99 58

EXPOSE DE PLATEFORME ET OUTILS DE DEVELOPPEMENT

*THÈME : DIGITALISATION DES ACTES DE  
MARIAGE*

FILIERE : Systèmes et Réseaux (SR2)

Noms des membres de groupe

- ✚ MOUNCHILI ISSAH YAZID
- ✚ FOTSO SIMO ANGE
- ✚ EMATOU CRESUS
- ✚ NJIGANG
- ✚ TINWE ULRICH

Sous la supervision de :

Mr TCHAKOUTIO LUDOVIC

ANNEE ACADEMIQUE

2023- 2024

10 mai 2024

1

## INTRODUCTION

La digitalisation des actes de mariage vise à remplacer les procédures traditionnelles basées sur des documents papier par des solutions numériques. L'objectif est de simplifier et d'accélérer le processus d'enregistrement des mariages, ainsi que de faciliter l'accès et la gestion des données relatives aux mariages.

# CHAPITRE1 : ANALYSE

L'analyse est primordiale dans tout projet informatique. Ici, nous allons décrire le problème posé, présenter les limites et proposer des solutions, faire une étude comparative des différentes méthodes d'analyse, présenter notre méthode d'analyse ainsi que tous les diagrammes représentant les différentes fonctionnalités de notre solution

## Aperçu

### INTRODUCTION

- I. RECEUIL DE L'EXISTANT
- II. ETUDE COMPARATIVE DES METHODES ET LANGAGES DE MODELISATION
- III. PRESENTATION DE LA METHODE D'ANALYSE
- IV. DIAGRAMMES D'ANALYSE

### CONCLUSION

## I. ETUDE DE L'EXISTANT

Au Cameroun pour établir un acte de mariage dans une mairie, il faudrait au préalable que les personnes désirantes s'unir se rapprochent des locaux d'une mairie un ou deux mois avant le jour du mariage pour inscrire le mariage.

Pour inscrire le mariage on demande un témoin à chacun c'est-à-dire que le marié vient avec un témoin et la mariée vient également avec un témoin. Ils doivent aussi fournir les pièces d'identification de chacun d'entre eux ainsi que de leurs témoins, il faut aussi des demi cartes photos des deux conjoints réunis.

Ces informations permettront à la mairie de faire la publication des bans pour le mariage en question. Les actes de mariage sont établis de façon manuscrite.

Le tableau ci-dessous représente les critique les conséquences et une solution

Critique	Conséquence	Solution
Gestion inefficace des documents	Vol	DIGITALISATION DES ACTES DE MARIAGE
Délai de traitement long	Détérioration des données	
Difficultés d'accès aux informations	Pertes des actes dans les archive	

## II. ETUDE COMPARATIVE DES METHODES ET LANGUAGES DE MODELISATION

### II.1. Généralité sur les méthodes d'analyses UML et MERISE

Une méthode définit une démarche reproductible qui produit des résultats fiables. Une méthode d'élaboration d'application décrit comment modéliser et construire les systèmes logiciels de manière fiable et reproductible. De manière générale, une méthode définie :

- Des éléments de modélisation ;
- Une représentation graphique ;

- Du savoir-faire et des règles.

Il existe plusieurs méthodes d'analyse, mais les plus utilisées sont MERISE et UML :

- **MERISE** (Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise) est une méthode d'analyse, de conception structurelle et de réalisation des systèmes d'information qui est élaborée en plusieurs étapes : schéma directeur, étude préalable, étude détaillée et la réalisation.
- **UML** (Unified Modeling Language), est un langage de modélisation des systèmes standards, qui utilise des diagrammes pour représenter chaque aspect d'un système : statique, dynamique ... en s'appuyant sur la notion d'orienté objet qui est un véritable atout pour ce langage.

## II.2. Etude comparative entre MERISE et UML

Tableau 2 : Etude comparative UML MERISE

MERISE	UML
Méthode systématique d'analyse et de conception des systèmes d'information :	Langage de représentation d'un système d'information
Représentation sous forme de model	Représentation sous forme de diagrammes
Approche systématique	Approche Objet
Franco-Français	Internationale

Propose de considérer le système réel selon deux aspects : Une vue statique (données) Une vue dynamique	Propose une approche différente en ce sens qu'elle associe les données et les traitements. Avec UML, les données d'un type et les traitements associés
Avec MERISE, nous avons une étude séparée des données et des traitements	UML permet de limiter les points de maintenances dans le code et facilite l'accès à l'information en cas d'évolution
Schéma directeur, étude préalable, Etude détaillée et la réalisation	Langage de modélisation des systèmes standards, qui utilise des diagrammes pour représenter chaque aspect d'un système : statique, dynamique... en s'appuyant sur la notion d'orienté objet

### III. PRESENTATION DE LA METHODE D'ANALYSE

#### III.1. Choix de la méthode d'analyse

Au regard de la complexité croissante des systèmes d'informations et de cette étude comparative, **UML** est le mieux adapté dans le cadre de notre projet. UML est un langage unifié de modélisation qui permet de spécifier, visualiser, construire et documenter de manière précise un système d'information. UML grâce à ses langages offre une description statique, dynamique et fonctionnelle du système à réaliser en plus il est idéal pour :

- Concevoir et déployer une architecture logicielle développée dans un langage objet. Certes UML, dans sa volonté "unificatrice" a proposé des formalismes ;
- Pour modéliser les données (le modèle de classe réduit sans méthodes et stéréotypé en entités), mais avec des lacunes que ne présentait pas l'entité relation de Merise ;
- Pour modéliser le fonctionnement métier (le diagramme de cas d'utilisation) qui est un formalisme très ancien.

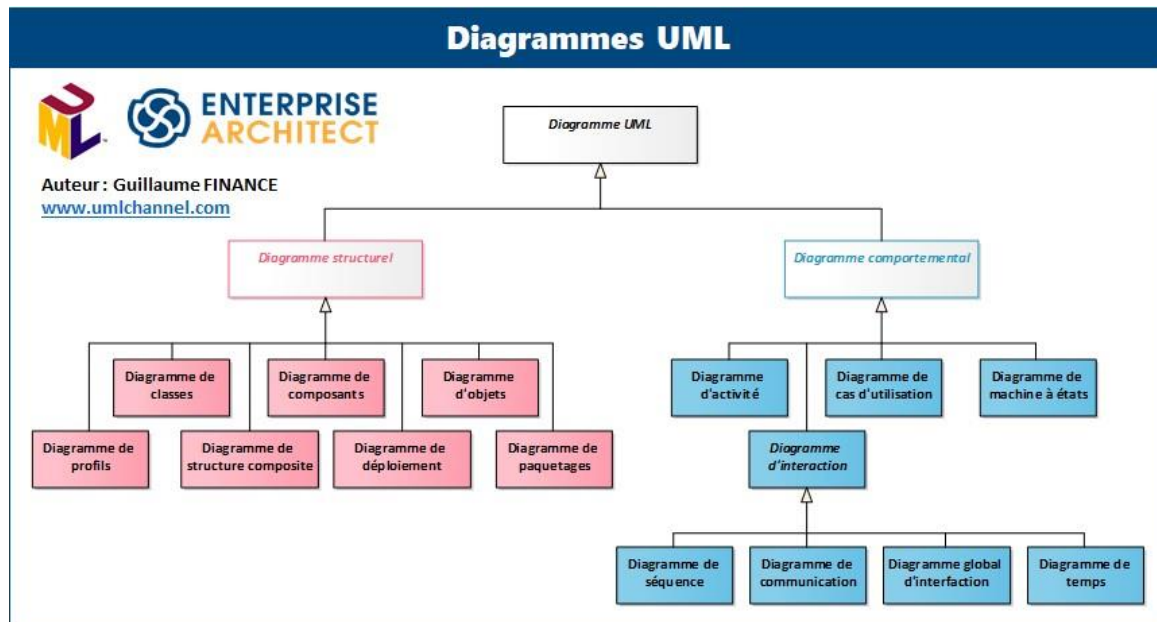
### III.2. Présentation de UML

La principale avancée des 15 dernières années réside dans la POO (programmation orientée objet). Face à ce nouveau mode de programmation, les méthodes de modélisation classiques tel que **MERISE** ont rapidement montré certaines limites et ont dû s'adapter. Dans ce contexte, l'Object management group (OMG) a eu pour objectif de définir une notation standard utilisable dans le développement informatique basé sur l'objet. C'est ainsi que naît **UML** qui est issue de la fusion des langages BOOCH, OMT et OOSE.

#### III.2.1 Version

La version d'UML que nous allons utiliser dans notre projet est la version 2.4 qui présente 14 diagrammes.

Figure 1 : diagramme UML version 2.5



Source : [www.umlchannel.com](http://www.umlchannel.com)

#### III.2.2. Les diagrammes d'UML

La modélisation d'un projet informatique effectué à l'aide d'UML version 2.5 présente 14 diagrammes. Ces diagrammes sont regroupés de la manière suivante :

- **Diagrammes comportementaux** : Diagramme de cas d'utilisation, d'activité, diagramme d'état transition et des diagrammes d'interactions (diagramme de séquence, de communication, le diagramme global d'interaction, le diagramme de temps).

- **Diagrammes structurels** : Diagramme de classe, diagramme de composants, diagramme d'objets, diagramme de profils, diagramme de structure composite, diagramme de déploiement et le diagramme de paquets.

## IV. DIAGRAMMES D'ANALYSE

### IV.1. Diagrammes de cas d'utilisation

#### IV.1.1. Définition



**Le diagramme de cas d'utilisation** identifie les fonctionnalités fournies par le système, les utilisateurs qui interagissent avec le système, et les interactions entre ces derniers. Les cas d'utilisations sont utilisés dans la phase d'analyse pour définir les acteurs de haut niveau du système. Les principaux objectifs d'un diagramme de cas d'utilisation sont les suivants :

- Fournir une vue de haut niveau de ce que fait le système ;
- Identifier les utilisateurs (acteurs) du système ;
- Déterminer des secteurs nécessitant des interfaces homme-machine.





#### IV.1.2. Formalisme

Dans un diagramme de cas d'utilisation on retrouve les éléments suivants :

Tableau 3 : éléments d'un diagramme de cas d'utilisation

REPRESENTATION	NOM	DESCRIPTION
	Acteur	Utilisateurs qui interagissent avec un système. Un acteur peut être une personne, une organisation ou un système externe qui interagit avec votre application ou votre système. Il s'agit nécessairement d'objets externes qui produisent ou consomment des données.
	Cas d'utilisation	Représente les différentes applications possibles pour un utilisateur.

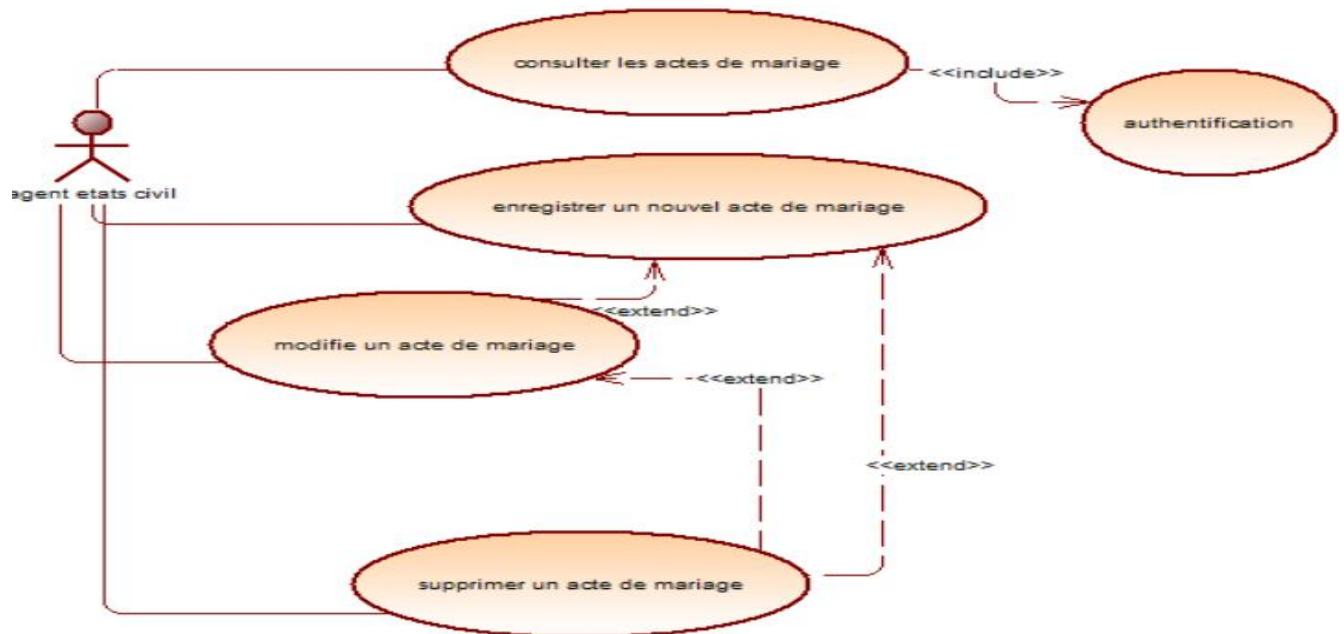


	Système	Séquence spécifique d'actions et d'interactions entre les acteurs et le système. Un système peut également être appelé scénario.
	Association	Lignes reliant les acteurs aux cas d'utilisation. Dans les diagrammes complexes, il est important de pouvoir identifier les acteurs associés à chaque cas d'utilisation.
	Relation	Les relations sont utilisées pour lier les cas d'utilisation entre eux. Elles permettent de définir quel cas d'utilisation dépend de l'autre (relation <b>&lt;&lt;include&gt;&gt;</b> ou d' <b>inclusion</b> ) ou encore quel cas d'utilisation est une possibilité étendue d'un autre (relation <b>&lt;&lt;extend&gt;&gt;</b> ou d' <b>extension</b> ). Les relations sont représentées par une flèche à trait interrompu allant d'un cas d'utilisation à un autre.
	Généralisation	Les généralisations sont utilisées pour lier les acteurs entre eux. Elles permettent de dire qu'un acteur effectue tous les cas d'utilisation d'un autre en plus de ce qu'il peut faire. Cette relation est aussi appelée relation d' <b>héritage</b> .
		Les généralisations sont représentées par une flèche allant d'un acteur (celui qui hérite) à un autre.

### IV.1.3. Diagrammes de cas d'utilisation de la solution

#### IV.1.3.1. Diagramme de cas d'utilisation global du système

Figure 2 : diagramme de cas d'utilisation global du système



## IV.2. Diagrammes de séquence





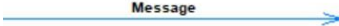
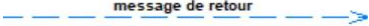

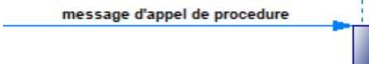
### IV.2.1. Définition

Vue de l'extérieur du système, Les Diagrammes de séquence permettent de montrer les interactions entre les utilisateurs et le système à travers une représentation linéaire et chronologique des cas d'utilisations du système. Chaque colonne correspond à un objet ou éventuellement à un acteur, introduit dans le diagramme des cas d'utilisation. La ligne de vie de l'objet représente la durée de son interaction avec les autres objets du diagramme.

### IV.3.2. Formalisme

Tableau 4 : Eléments d'un diagramme de séquence

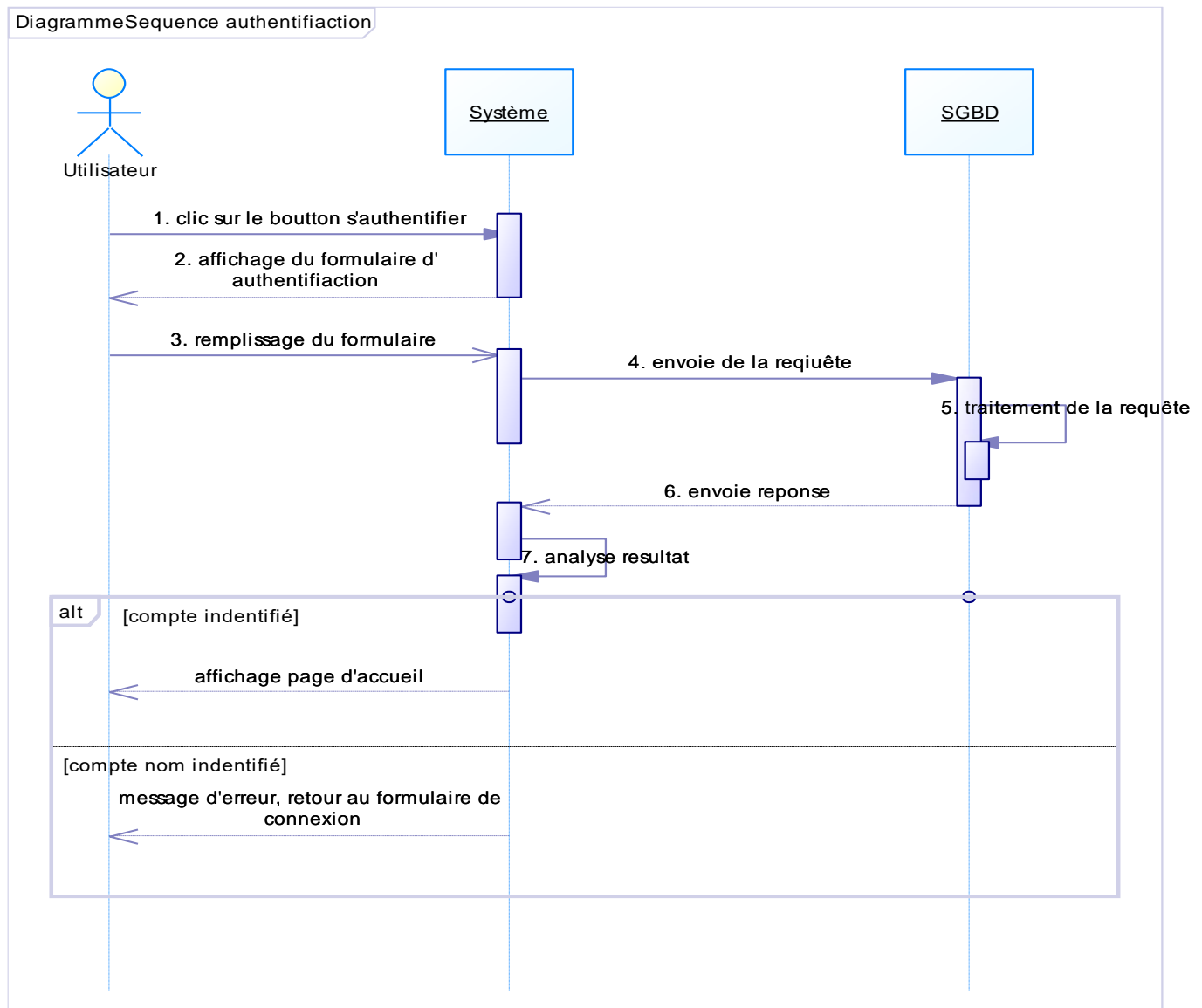
REPRESENTATION	NOM	DESCRIPTION

	Ligne de vie Acteur	La ligne de vie représente l'ensemble d'activité effectué par l'acteur
	Ligne de vie Objet	La ligne de vie représente l'ensemble d'activité effectué par un objet de l'application
	Activation	Indique qu'une action est lancée
	Alternatif	Condition à l'envoi d'un message
	Message asynchrone	Message ou l'émetteur n'est pas bloqué lors de l'envoi et peut continuer son exécution
	Message de retour	Réponse d'un message synchrone
	Message d'appel Réflexif	Lorsqu'un acteur ou un Objet s'auto envoie un message
	Message d'appel de procédure ou Message synchrone	Message synchrone ou l'émetteur est bloqué en attente de retour

## IV.2.3. Diagrammes de séquence de la solution

## IV.2.3.1. Diagramme de séquence « authentification »

Figure 3 : diagramme de séquence "authentification"



## CONCLUSION

L'analyse d'un projet est une étape cruciale qui doit être effectuée avant de concevoir son application car elle nous permet de cerner les différentes fonctionnalités de notre application. Dans la suite nous allons ressortir le dossier de conception de notre projet.

## CHAPITRE 2 : CONCEPTION

Le chapitre de conception vient concrétiser les résultats obtenus lors de l'analyse. Dans cette section nous parlerons des objectifs de ce document, et des diagrammes y intervenant.

### Aperçu

#### INTRODUCTION

#### I. DIAGRAMME DE CLASSE

#### II. DIAGRAMME DE PAQUETAGE

#### III. DIAGRAMME DE COMPOSANT

#### CONCLUSION

## INTRODUCTION

Dans la réalisation d'un projet informatique, le dossier de conception est celui qui présente l'architecture du système étudié. L'analyse UML nous permet à travers ses diagrammes structurels de représenter la structure de notre application ainsi que les différents composants matériels et logiciels. Dans cette partie, nous vous présenterons le diagramme de classe, son model physique de données associé.

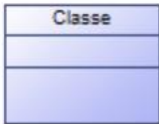
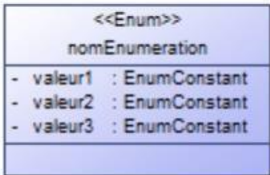
### I. DIAGRAMME DE CLASSE




#### I.1. Définition

Le diagramme de classe est l'un des types de diagrammes les plus utiles dans une modélisation UML, car il décrit clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. Une classe définit un type d'objet et possède un nom, des attributs et des opérations. Le nom d'une classe commence par une majuscule, le nom d'une propriété commence par une minuscule, les types de base (Int, long, double, boolean) sont en minuscules, il n'y a pas d'espace dans les noms des classes ou de propriétés.

#### I.2. Formalisme

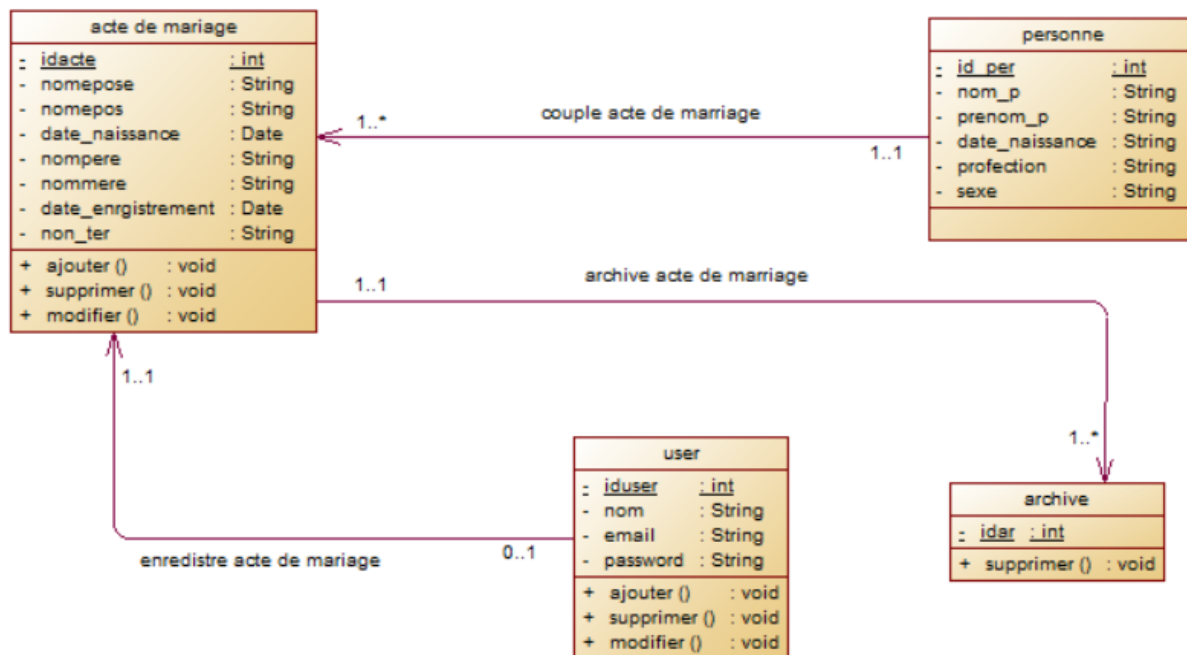
Tableau 5 : éléments d'un diagramme de classe

REPRESENTATION	NOM	DESCRIPTION
	Classe	Une classe est une représentation abstraite d'un ensemble d'objets, elle contient les informations nécessaires à la construction de l'objet (c'est-à-dire la définition des attributs et des méthodes).
	ENUMERATION	L' <b>énumération</b> UML est particulièrement adaptée pour modéliser un ensemble fini de valeurs possibles que peut recevoir l' <b>attribut d'une classe</b> .

nomAttribut : Type	ATTRIBUT	Fin de vie du système, destruction de l'objet
nomMéthode () : Type	METHODE	Abstraction d'un moment de la vie d'une entité pendant laquelle elle satisfait un ensemble de conditions
	ASSOCIATION	L'association est représentée par un simple trait continu, reliant deux classes. Le fait que deux instances soient ainsi liées permet la navigation d'une instance vers l'autre, et vice versa (en générale une classe possède un attribut qui fait référence à l'autre classe)
	AGREGATION	C'est une relation particulière qui attribue à l'une des classes le rôle d' <b>agrégat</b> et à l'autre classe le rôle d' <b>agrégé</b> . L'agrégation peut être assimilée à une appartenance faible.
	COMPOSITION	Il s'agit d'une appartenance forte. La vie de l'objet composant est liée à celle de son composé.

### I.3. Représentation du diagramme de classe de la solution

Figure 4 : diagramme de classe



## II. Diagramme de package

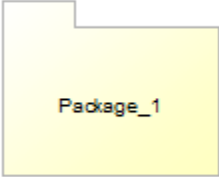
### II.1. Définition

Le **diagramme de paquetages** permet de décomposer le système en catégories ou parties plus facilement observables, appelés « packages ». Cela permet également d'indiquer les acteurs qui interviennent dans chacun des packages.

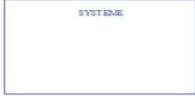

### II.2. Formalisme

Dans un diagramme de packages on retrouve les éléments suivants :

Tableau 1 : éléments d'un diagramme de cas d'utilisation

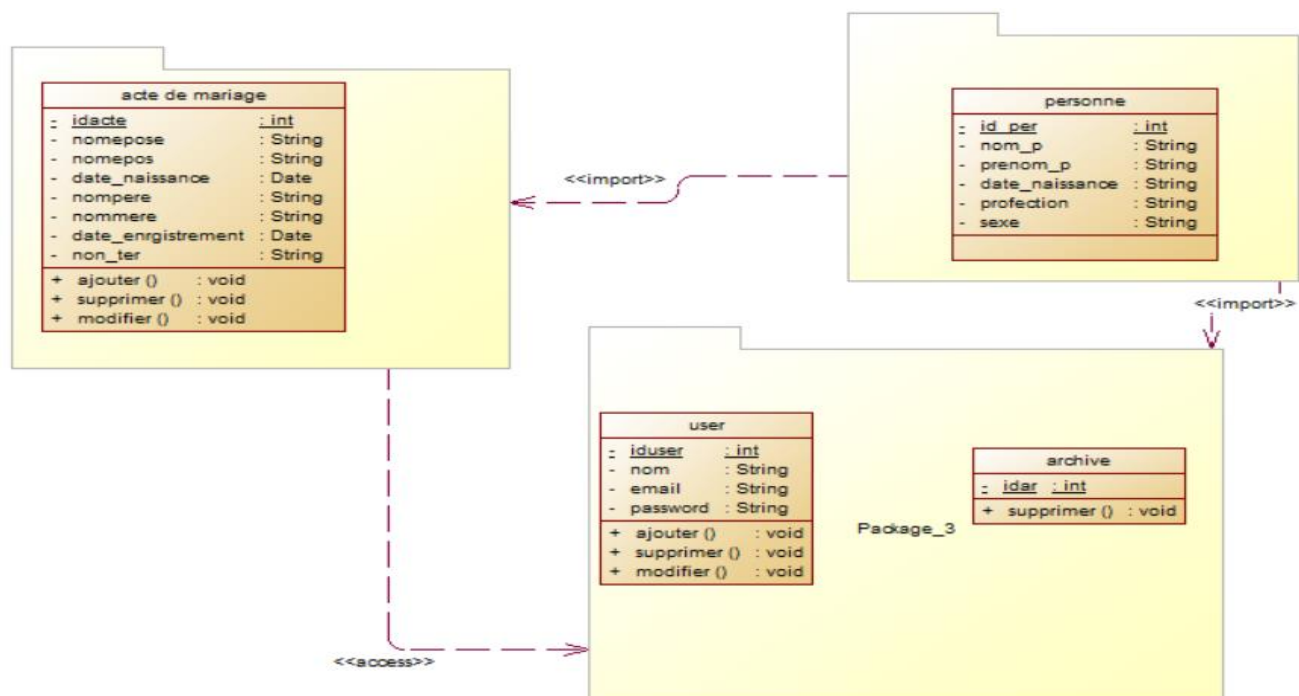
REPRESENTATION	NOM	DESCRIPTION
	Package	Représente les différentes familles de fonctionnalités du système.



	Système	Séquence spécifique d'actions et d'interactions entre les acteurs et le système. Un système peut également être appelé scénario.
	Association	Lignes reliant les différents paquets.

### II.3. Diagramme de paquetage de la solution

Figure5 : diagramme de paquetage de la solution



## CONCLUSION

La phase de conception est une étape primordiale dans la conception logicielle, car elle nous donne une vue globale de la structure de notre application. Après cette étape nous pouvons passer à la réalisation de notre projet.

## CHAPITRE 3 : REALISATION

Cette partie présente les différentes technologies, langages de programmation, architectures utilisées nécessaires pour la réalisation de notre application ainsi que son déploiement.

### Aperçu

#### INTRODUCTION

- I. PRESENTATION DE L'ENVIRONNEMENT LOGICIEL ET TECHNOLOGIES UTILISEES
- II. PRESENTATION DE L'ARCHITECTURE UTILISEE
- III. DIAGRAMME DE DEPLOIEMENT

#### CONCLUSION




## INTRODUCTION


Le dossier de réalisation permet de représenter l'environnement logiciel utilisé, les différentes technologies utilisées pour la réalisation de notre projet ainsi que son déploiement. Dans cette partie, nous présenterons les différentes technologies et environnement logiciel utilisé ainsi que l'architecture de notre application.

### I. Langages de programmation et technologies utilisées

Dans le cadre de notre projet et parmi le grand nombre de technologies existantes nous avons opté pour les technologies suivantes :

*Figure 1 : technologies et langages de programmation utilisés*

TECHNOLOGIE et LANGAGES	ROLE
	HyperText Preprocessor langage serveur utilisé pour traiter du contenu dynamique
	JQuery est ce qu'on appelle une « librairie » ou une « bibliothèque » JavaScript. Le rôle d'une librairie, en informatique, est de simplifier l'utilisation d'un certain langage de programmation en fournissant un ensemble de codes déjà prêt à l'emploi. Nous l'utilisons pour effectuer des requêtes de type Ajax qui s'exécute de façon asynchrone grâce aux callbacks, et aussi pour manipuler facilement un tableau HTML à l'aide de son plugin DataTables
	Html est un langage de programmation utilisé pour gérer contenu de notre application Le CSS est utilisé pour gérer la forme

 <p>Version4</p>	<p><b>Bootstrap</b> est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de site et application web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScripts en option.</p>
---	--

## II. PRESENTATION DE L'ARCHITECTURE DE L'APPLICATION

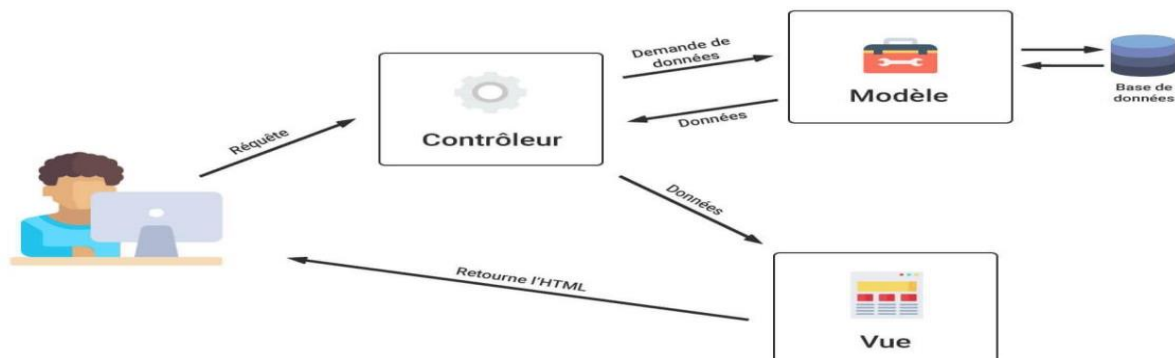
### II.1. Architecture logique

Notre projet suit une logique applicative standardisé et utilisée dans la plupart des entreprises de nos jours. Cette logique est le MVC (Modèle-Vue-Contrôleur) qui est un design pattern permettant de mieux organiser son code.

- **Le model** contient les données à afficher
- **La vue** contient la présentation de l'interface graphique
- **Le contrôleur** contient la logique concernant les actions effectuées par l'utilisateur.

Dans ce motif d'architecture logicielle, le code est organisé de manière à faciliter la détection des erreurs, facilité la collaboration et la réutilisabilité du code.

Figure 2 : représentation de l'architecture MVC



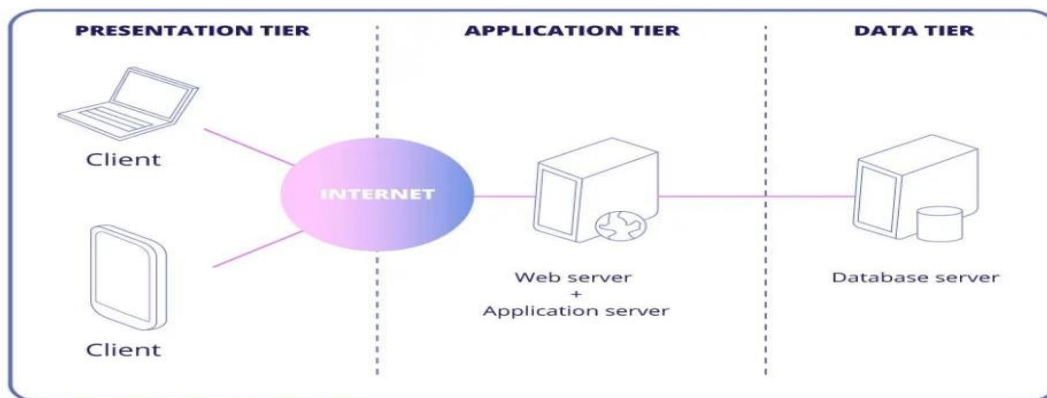
Source : [www.architecturel.com](http://www.architecturel.com) 28/08/2022, 14 :45

## II.2. Architecture physique

Notre application a été implémentée suivant le modèle d'architecture 3-tiers encore appelée Architecture à 3 niveaux. Cette architecture est l'application du modèle plus général qu'est le multi-tiers. L'architecture logique du système est divisée en trois niveaux ou couches :

- **Une couche présentation** qui correspond à l'affichage, la restitution des données sur le poste de travail, le dialogue avec un utilisateur.
- **Une couche de traitement** qui correspond à la logique métier des données, la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative.
- **Une couche d'accès aux données** qui correspond aux données qui sont destinées à être conservées de manière persistante.

Figure 3 : Présentation de l'architecture 3-tiers



Source : [www.architecturel.com](http://www.architecturel.com) 28/08/2022, 14 :45

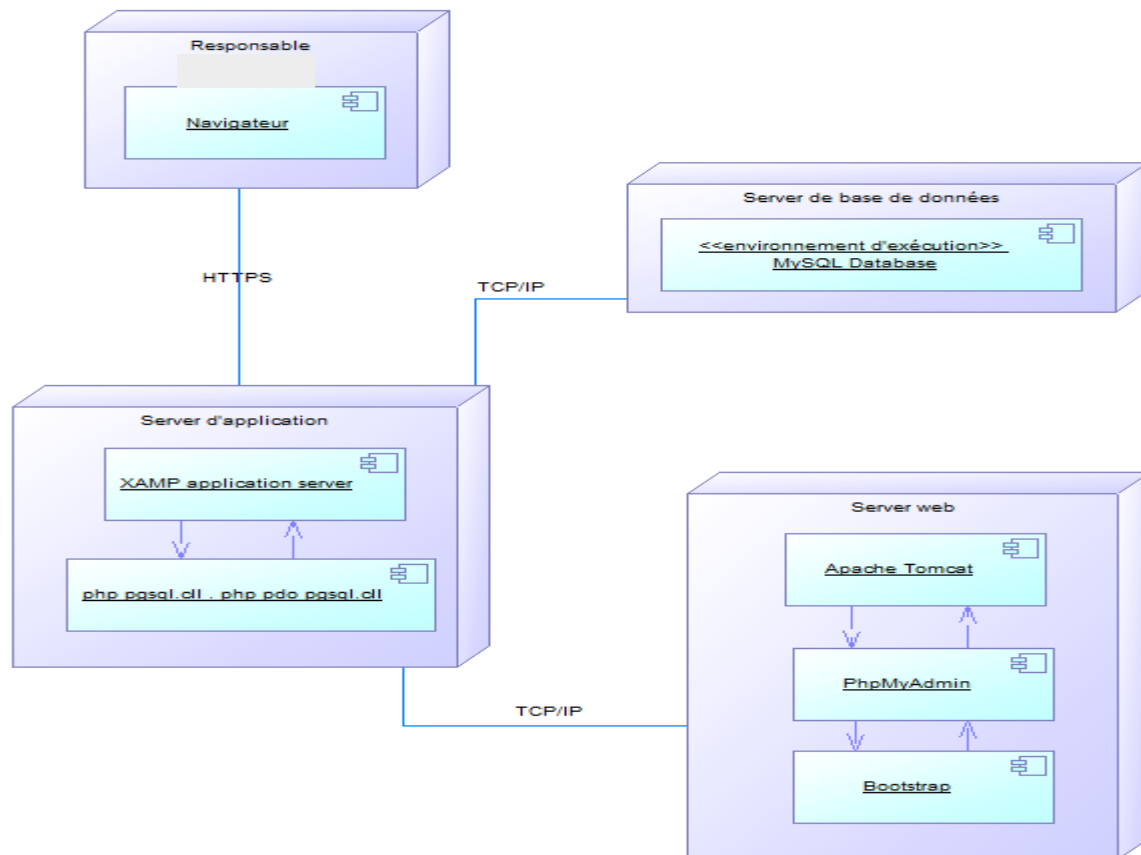
### III. DIAGRAMME DE DEPLOIEMENT

#### III.1. Définition

Le diagramme de déploiement permet de représenter d'une part la disposition physique des ressources matérielles qui constituent le système et montre la répartition des composants (éléments logiciels) qui s'exécutent sur ces matériels et d'autres parts les chemins de communication entre les différentes ressources matérielles.

#### III.3. Diagramme de déploiement de la solution

Figure 6 : diagramme de déploiement de la solution



## CONCLUSION

La phase de réalisation est une étape très importante où nous nous concentrons sur l'aspect technique de notre application, des différentes technologies qui sont utilisées pour un résultat optimal. Le choix de ces technologies nécessite une bonne étude car toute application mal réalisée conduit à la production d'une mauvaise application.

## CHAPITRE 4 : GUIDE D'UTILISATION

Cette partie présente notre application ainsi que quelques fonctionnalités qu'elle regorge. Elle vous présentera un aperçu de notre travail ainsi que quelques indications pour l'utiliser.

### Aperçu

#### INTRODUCTION

##### I. PRESENTATION DE NOTRE APPLICATION

##### II. PRESENTATION DE QUELQUES FONCTIONNALITES

#### CONCLUSION

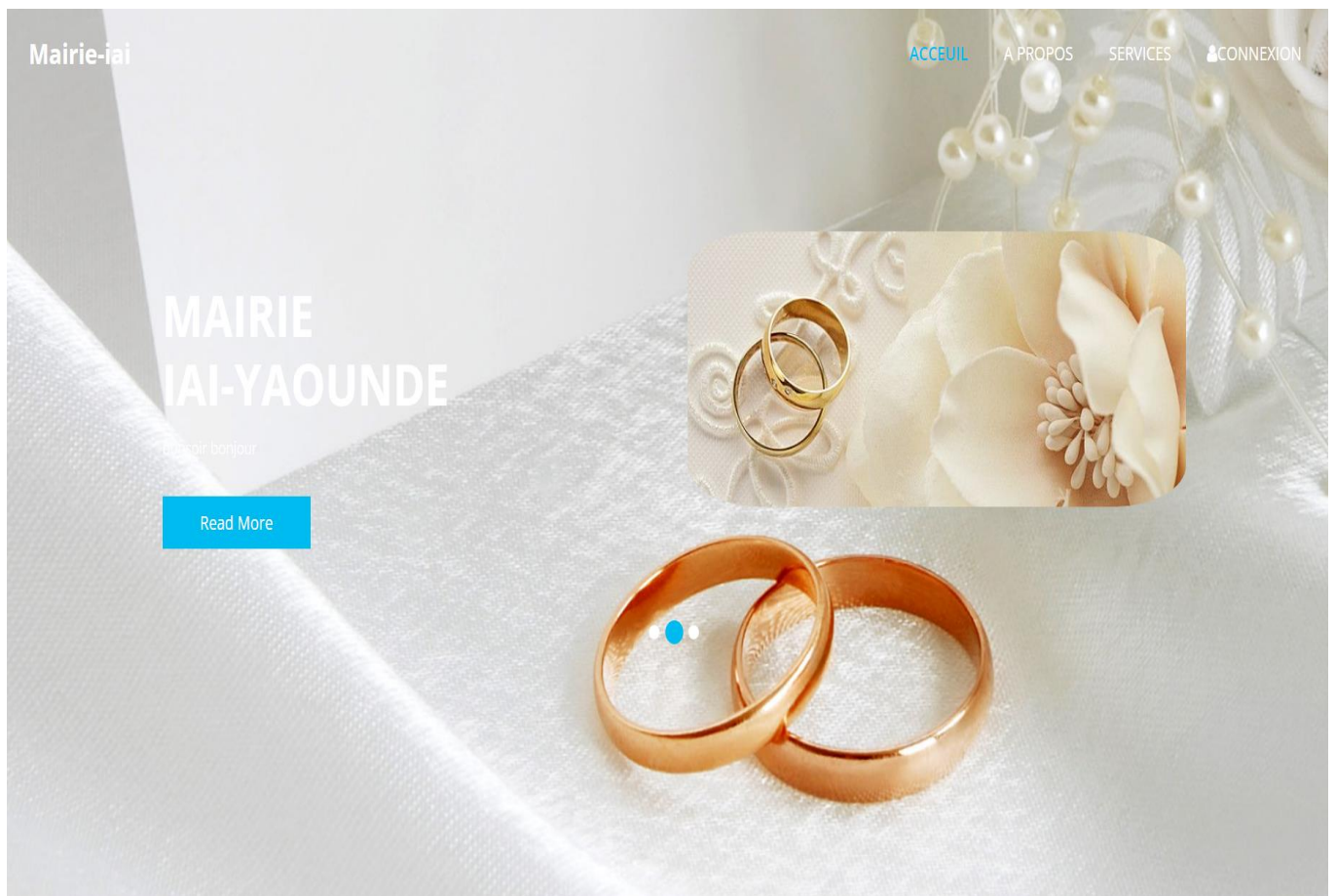
## INTRODUCTION

Le Guide d'utilisation est ce document permettant de présenter les différentes fonctionnalités de notre application. Dans cette partie, nous allons présenter notre application et ensuite étaler quelques fonctionnalités disponibles sur notre application.

### I. PRESENTATION DE NOTRE APPLICATION

#### 1. Présentation des interfaces

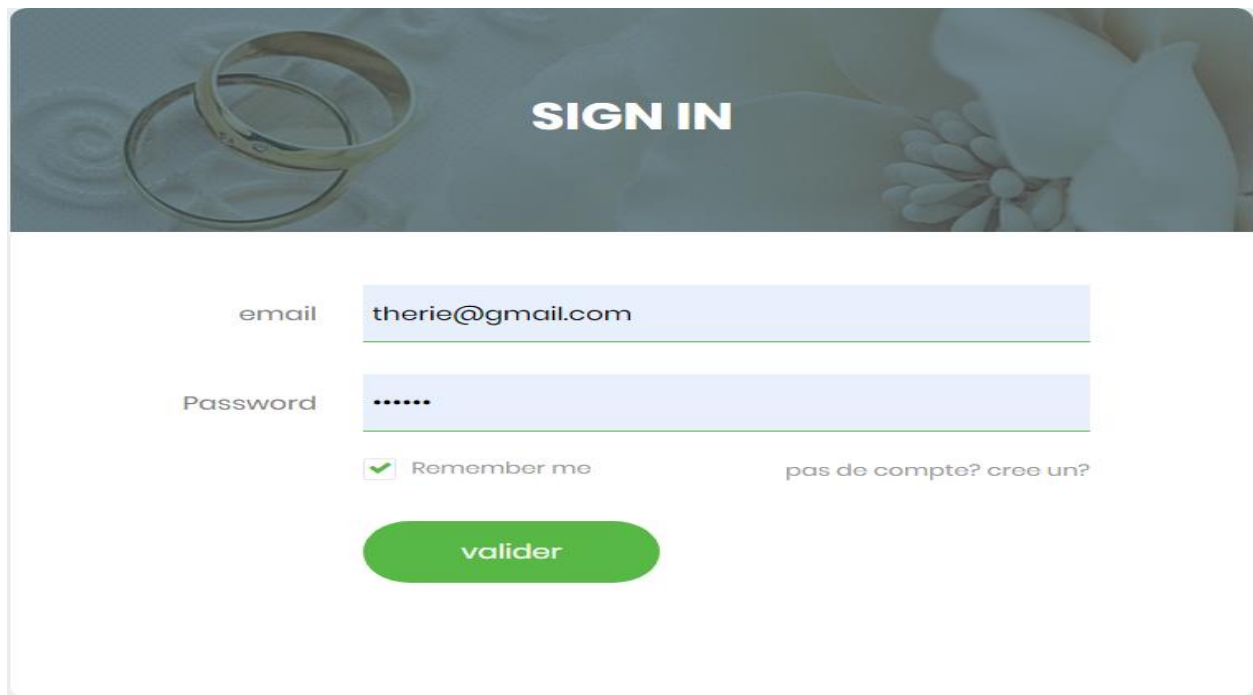
##### 1.2 Interface accueil



© 2024 cree par mounchili yaziddounya83@gmail.com

##### 1.3 Interface de connexion





The SIGN IN form features a header with a background image of wedding rings and flowers. The form includes an email field with the value 'therie@gmail.com', a password field with masked characters '.....', a checked 'Remember me' checkbox, and a link 'pas de compte? cree un?'. A green 'valider' button is at the bottom.

**SIGN IN**

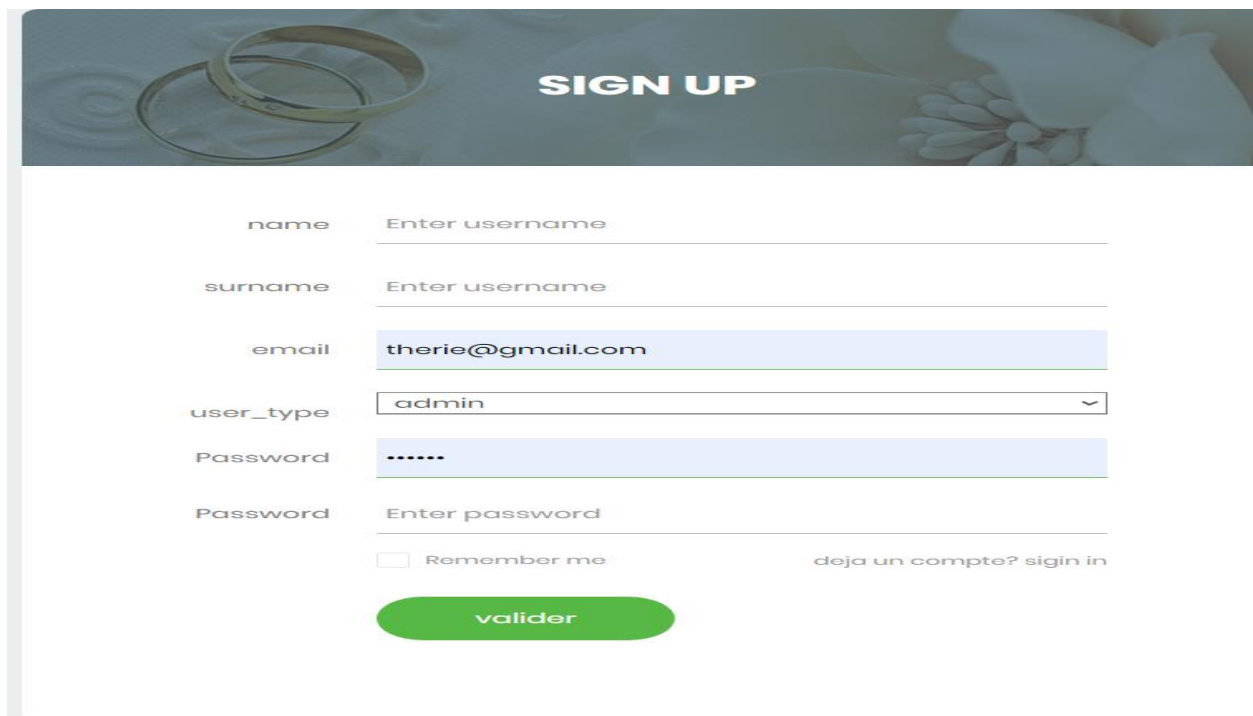
email

Password

☒ Remember me [pas de compte? cree un?](#)

**valider**

#### 1.4 Interface d'inscription



The SIGN UP form features a header with a background image of wedding rings and flowers. The form includes fields for 'name' (placeholder 'Enter username'), 'surname' (placeholder 'Enter username'), 'email' (value 'therie@gmail.com'), 'user\_type' (dropdown menu with 'admin' selected), a password field (masked '.....'), and a second password field (placeholder 'Enter password'). It also has an unchecked 'Remember me' checkbox and a link 'deja un compte? signin in'. A green 'valider' button is at the bottom.

**SIGN UP**

name

surname

email

user\_type

Password

Password

☐ Remember me [deja un compte? signin in](#)

**valider**

#### 1.5 Interface Administrateur

mounchili

ADMIN

search

transaction aujourd'hui

PAYEMENT

500.00\$

\*\*\* \*\* 3484

COUPLE FIANCE

12

COUPLE ENREGISTRER

6

vip

ARCHIVE


50

reussi




couple enregistrer avec succès

Nom du marie	nom de la mariee	regime matrimoniale	
<div></div> <div></div> <div>therie</div> <div>mmf</div>	<div></div> <div>yousouf</div> <div>menager</div>	<div></div> monogamie	<div>imprimer</div> <div></div>

## 1.6 Interface enregistre couple et acte de mariage

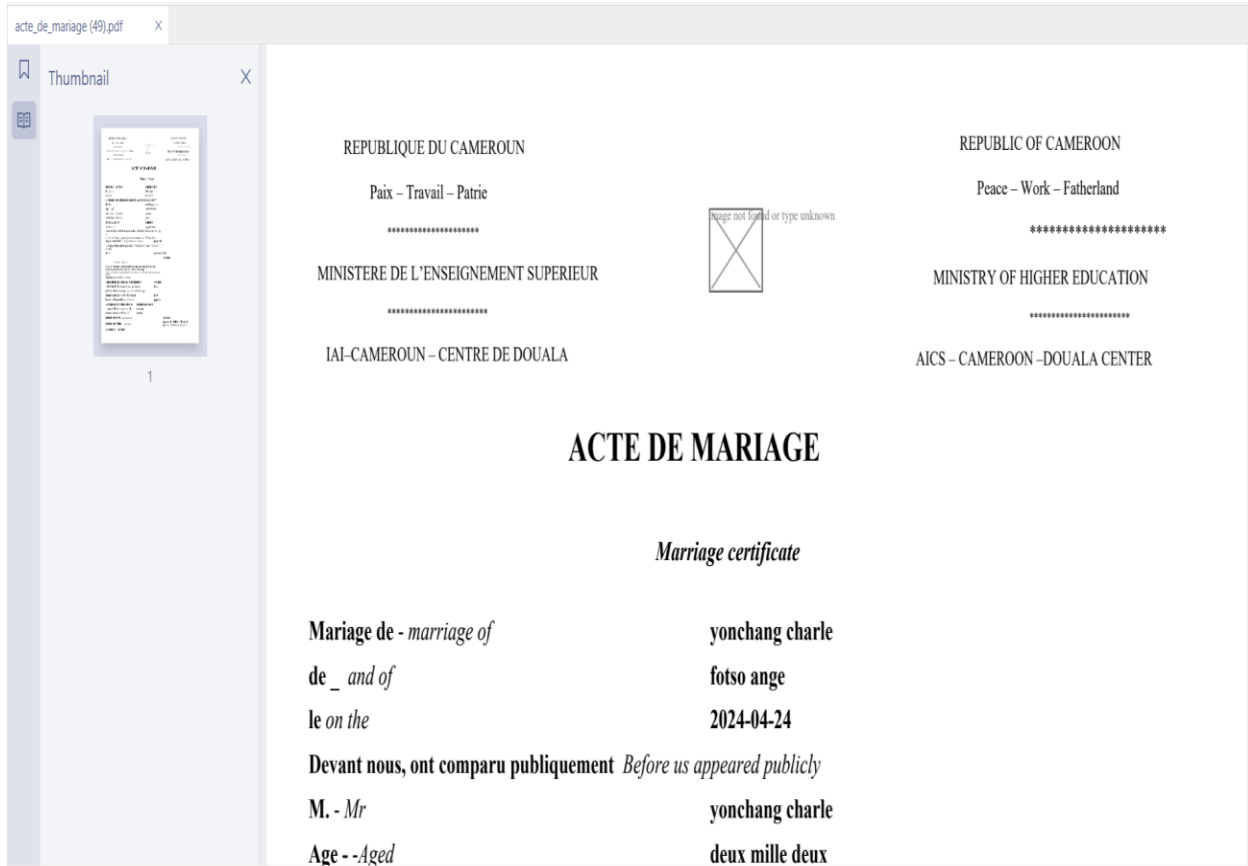


## ENREGISTRE L'ACTE DE MARIAGE

nom de l'époux	entre le nom de l'époux
nom de l'épouse	entre le nom de l'épouse
le	mm/dd/yyyy 
age de l'époux	age de l'époux
profession de l'époux	entre le profession de l'époux
domicile de l'époux	entre le domicile de l'époux
pere de l'époux	entre le nom de pere de l'époux de l'époux
mere de l'époux	entre le nom de mere de l'époux de l'époux
age epouse	age l'épouse
profession epouse	profession de l'épouse
domicile	domicile de l'épouse
pere de l'épouse	nom du pere de l'épouse
mere de l'épouse	nom de la mere de l'épouse
regime matrimoniale	polygamie 
chef famille	chef famille de l'époux
chef famille	chef famille de l'épouse
temoin epoux	nom temoin de l'époux
temoin epouse	nom temoin de l'épouse
dresser par	acte dresser par
	Le présent acte a été dressé par
	Assisté de Secrétaire d'Etat-civil
fait le	mm/dd/yyyy 

valider

### 1.7 Aperçu du fichier PDF de l'acte de mariage.



## CONCLUSION

Le but de ce dossier était de mieux étayer les futurs utilisateurs de notre solution, sur son utilisation. Ayant présenté quelques fonctionnalités de notre application, Ce chapitre est donc apte à remplir sa mission et pourra permettre à n'importe qui d'utiliser l'application.

## APPORT PERSONNEL :

✚	MOUNCHILI ISSAH YAZID	100% concepteur, control
✚	FOTSO SIMO ANGE	95% view
✚	EMATOU CRESUS	95% model
✚	NJIGANG	60%
✚	TINWE ULRICH	50%

## TABLE DE MATIERES :

INTRODUCTION .....	2
I. ETUDE DE L'EXISTANT .....	4
II. ETUDE COMPARATIVE DES METHODES ET LANGUAGES DE MODELISATION .....	4
II.1. Généralité sur les méthodes d'analyses UML et MERISE.....	4
II.2. Etude comparative entre MERISE et UML.....	5
III. PRESENTATION DE LA METHODE D'ANALYSE .....	6
III.1. Choix de la méthode d'analyse .....	6
III.2. Présentation de UML .....	7
IV. DIAGRAMMES D'ANALYSE .....	8
IV.1. Diagrammes de cas d'utilisation.....	8
IV.2. Diagrammes de séquence.....	10
I. DIAGRAMME DE CLASSE .....	14
I.1. Définition.....	14
I.2. Formalisme .....	14
I.3. Représentation du diagramme de classe de la solution.....	15
II. Diagramme de package .....	16
II.3. Diagramme de paquetage de la solution.....	17
I. Langages de programmation et technologies utilisées .....	19
II. PRESENTATION DE L'ARCHITECTURE DE L'APPLICATION.....	20

II.1. Architecture logique .....	20
II.2. Architecture physique.....	21
III. DIAGRAMME DE DEPLOIEMENT .....	22
III.1. Définition .....	22
III.3. Diagramme de déploiement de la solution.....	22
APPORT PERSONNEL : .....	29
TABLE DE MATIERES : .....	29