[I. Introduction générale 3](#_Toc83607892)

[I.1. Contexte 4](#_Toc83607893)

[I.2. Problématique 4](#_Toc83607894)

[I.3. Objectifs 5](#_Toc83607895)

[II. Analyse et Conception 5](#_Toc83607896)

[II.1. Fonctionnalités du Système 5](#_Toc83607897)

[II.2. Outil d’analyse et de conception 5](#_Toc83607898)

[II.2.1. Méthode MERISE 5](#_Toc83607899)

[II.2.2. Langage UML 6](#_Toc83607900)

[II.2.3. Choix du langage UML 6](#_Toc83607901)

[II.2.4. Présentation des diagrammes d’UML 7](#_Toc83607902)

[II.2.4.1. Diagrammes de cas d’utilisation 7](#_Toc83607903)

[II.2.4.2. Diagrammes de classe 7](#_Toc83607904)

[II.2.4.3. Diagrammes de séquences 7](#_Toc83607905)

[II.2.4.4. Diagrammes de cas d’activités 8](#_Toc83607906)

[II.2.4.5. Autres diagrammes 9](#_Toc83607907)

[II.3. Modélisation de notre application 9](#_Toc83607908)

# Introduction générale

Depuis l’âge de pierre, l’homme a toujours eu des tonnes d’informations, liées à lui-même ou son entourage. Cependant, il y a des informations qui sont de natures très capitales, et d’autres qui sont moins importantes. C’est dans ce sens que le besoin de se donner les moyens pour se rappeler les informations les plus essentielles qui sont liées à sa naissance, son mariage et son décès. Même si certaines personnes ont une mémoire infaillible, d’autres oublient vite. C’est pourquoi, les premiers hommes essayaient de se souvenir avec les moyens de leur époque comme tailler des pierres et écrire des mots ou dessiner une forme ou une image qui a pour but de leurs rappeler les informations qu’ils désirent ne pas oublier. Avec l’apparition des écritures hiéroglyphes, cette manière d’écrire aborde de nombreux aspects de la civilisation égyptienne plus particulièrement les actes administratifs. Plus tard, Johannes Gutenberg inventa l’imprimerie, et c’est ainsi que l’industrie de l’imprimerie va permettre aux hommes d’imprimer rapidement plusieurs exemplaires de copies pour compiler des livres. Ainsi, ils pourront imprimer des registres pour écrire des informations de chaque citoyen.

Chaque être humain peut être identifié par son état, que ça soit par son nom, prénom, sa nationalité, sa date de naissance, sa situation matrimoniale (marié ou célibataire), s’il est mort ou vivant, etc. C’est dans cette réflexion qu’au 18éme siècle les états civils voient le jour : les communes remplacent les paroisses, les officiers d'état civil prennent le relais des curés et de leurs registres paroissiaux. Par la suite, le nombre de registre a augmenté, et ils sont devenus des archives qui renferment des informations d’identification des citoyens. Par ailleurs, n'y a-t-il pas de risque que ces informations soient juste enregistrées sur des papiers archivés ? C’est dans ce contexte, que notre sujet de recherche se porte sur : la dématérialisation des données des états civils. D’abord, on mettra en évidence les problématiques qui tournent autour de ce sujet et de définir les objectifs. Ensuite, on proposera une solution technologique qui résoudra ce problème et à la fin ouvrir des perspectives

## Contexte

A Rechercher

## Problématique

L’objectif principal d’une institution administrative est d’apporter de la valeur sur les démarches administratives. Ce qui signifie, satisfaire les citoyens avec un service  
efficace. Cependant, aujourd’hui, dans les centres d’état civil, il y a beaucoup de choses qui  
méritent une amélioration :

* Les procédures d'obtention et de récupération des documents d'états civils qui deviennent de plus en plus difficiles pour de nombreux citoyens et qui n'obtiennent quasiment aucune satisfaction par rapport à leurs demandes
* Les risques autour des locaux :

Les institutions administratives sont remplies de tas d’archives, en d’autres termes c’est plusieurs papiers qui sont accumulés entre quatre murs. En étant un endroit qui regroupe des archives, il convient alors de déduire les potentiels dangers que les registres peuvent encourir. L'une des causes les plus remarquables de l'inefficacité de l'état civil est la mauvaise conservation des archives. Parmi ces dangers, il y’a les incendies, les inondations et parmi tant d’autres.

* Les problématiques liées aux services :

Pour demander les services états civiles dans une mairie, les demandeurs restent parfois debout sauf quelques-uns qui peuvent rester assis sur des bancs à cause du manque de salles d’attente pouvant accueillir les citoyens. Ainsi, formant une longue file d’attente avec chacun son document à la main en attendant leur tour. Et même en dehors de cela, la lenteur des agents pour trouver le numéro de registre d’un demandeur, et aussi les éventuelles erreurs qu’ils font toujours en rédigeant les extraits de naissance. En résumé, la gestion des clients ou demandeurs et des états civils manque de rigueur  
et d’efficacité.

* …

Pour remédier à ces insuffisances, nous avons décidé de mettre en place un site web dynamique pouvant apporter des solutions efficientes pouvant régler les dysfonctionnements au niveau de l'état civil.

## Objectifs

Les objectifs de ce projet sont multiples mais l’idée générale est de proposer une alternative au système actuel qui facilitera les tâches des agents de travail et aussi d’apporter plus d’assistance aux demandeurs. C’est nouveau système va non seulement automatiser les tâches mais en même temps rendre le travail plus rapide. L’état civil doit se doter des technologies de pointes pour moderniser ses techniques de traitement des documents et pour sauvegarder sûrement et durablement ses archives.

Pour les citoyens nous allons prévoir un espace où ils pourront effectuer les fonctionnalités suivantes :

* Permettre à un citoyen de faire une demande

L’objectif ultime de ce projet c’est avec les demandes d’état civile. Avec notre plateforme, ils vont pouvoir faire à distance leur différents types de demandes : acte de naissance, acte de mariage et acte de décès.

A CONTINUER

# Analyse et Conception

## Fonctionnalités du Système

A revenir

## Outil d’analyse et de conception

### Méthode MERISE

Merise est une méthodologie de modélisation à usage général dans le domaine du développement de systèmes d’information, du génie logiciel et de la gestion de projet. Introduit pour la première fois au début des années 1980, Il a été développé et perfectionné à un point tel que la plupart des grandes organisations gouvernementales, commerciales et industrielles françaises l'ont adopté. Merise procède à un traitement séparé des données et des processus, où la vue des données est modélisée en trois étapes : de la conception à la physique en passant par la logique. De même, la vue axée sur les processus passe par les trois étapes conceptuelle, organisationnelle et opérationnelle. Ces étapes du processus de modélisation sont parallèles aux étapes du cycle de vie : planification stratégique, étude préliminaire, étude détaillée, développement, mise en œuvre et maintenance. C'est une méthode d'analyse basée sur le modèle entité-relation. En utilisant Merise, vous pouvez concevoir des tables avec des relations pour créer une base de données relationnelle.

### Langage UML

Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (**UML**), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

Un diagramme UML est une représentation graphique, qui s’intéresse à un aspect précis du modèle. C’est une perspective du modèle, pas « le modèle ».

UML dans sa version 2 propose treize diagrammes qui peuvent être utilisés dans la description d’un système. Ces diagrammes sont regroupés en deux grands ensembles :

* Les digrammes structurels : Ces diagrammes aux nombres de six, représentent l’aspect statique d’un système (classes, objets, composants).
* Les digrammes de comportement : Ces diagrammes (7) représentent la partie dynamique d’un système réagissant aux évènements et permettant de produire des résultats attendus par les utilisateurs d’UML.

Cependant dans ce projet on a utilisé deux des diagrammes à savoir le diagramme de cas d’utilisation (digramme de comportement) et le diagramme de classe (diagramme structurels).

### Choix du langage UML

Notre choix s’est porté sur le langage UML pour sa précision et son exploitabilité. De plus il est conçu :

* Pour s’adapter à n’importe quel langage de programmation oriente objet, et il présente plusieurs diagrammes dont leurs compréhensions nécessitent une grande attention
* Pour l'implémentation objet avec ses différents détails et sa portabilité (s'adapte à n'importe quelle plateforme) elle est donc plus exploitable.

### Présentation des diagrammes d’UML

#### Diagrammes de cas d’utilisation

Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport aux systèmes. Il constitue un des digrammes les plus structurants dans l’analyse d’un système. Il s’intéresse à ce que doit faire le système sans spécifier comment il le fait .Il représente aussi :

* Le système : il spécifie l’application informatique
* Les acteurs : ils sont des entités externes qui agissent sur le système
* Les cas d’utilisations : les services rendus par le système
* Les liens entre les acteurs et cas d’utilisations.

Il constitue un instrument de validation et e test du système en cours et en fin de construction

#### Diagrammes de classe

Il représente la description statique en intégrant dans chaque classe la partie dédiée aux données et celle consacrées aux traitements. C’est le diagramme pivot de l’ensemble de la modélisation d’un système.

De plus il fournit une représentation abstraite des objets du système qui doivent interagir pour réaliser les fonctions du système.

Pour créer un diagramme de classe il faut :

* Identifier et décrire les classes
* Identifier et décrire les relations qui existent entre ces classes

Une classe est représentée à l’aide d’un rectangle comportant trois compartiments de base :

* Le nom qui doit évoquer le concept qui décrit la classe et doit commencer par une lettre majuscule,
* Les attributs qui décrivent les caractéristiques des objets
* Les opérations décrivent ce qu’une classe peut faire et quels services offre-t-elles aux autres classes.

#### Diagrammes de séquences

L’objectif du diagramme de séquence est de représenter les interactions entre objets qui composent le système en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d’utilisation en considérant les différents scenarios associes.

Il se concentre sur la séquence des interactions d’un point de vue temporel. Il fait apparaitre les objets intervenant dans l’interaction (les acteurs ou objets appartenant au système) et la description des interactions (messages) entre les intervenants.

Dans un digramme de séquence, deux types de messages peuvent être distingues :

* Message synchrone : Dans ce cas l’émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message. Le message retour peut ne pas être représente car il est inclus dans la fin d’exécution de l’opération de l’objet destinataire du message.
* Messages asynchrones : Dans ce cas l’émetteur n’attend pas la réponse à son message. Il poursuit l’exécution de ses opérations. C’est une flèche avec extrémité non pleine qui symbolise ce type de message.

#### Diagrammes de cas d’activités

Le diagramme d’activités concerne le comportement interne des opérations ou des cas d’utilisation. Cependant le comportement vise ici s’applique au flot de contrôle et aux flots de données propres a un ensemble d’activités et non plus relativement à une seule classe.

Il décrit les traitements en s’affranchissant partiellement de la structuration e l’application .il modélise le flot de contrôle des traitements.

Les composants fondamentaux d’un DAC sont :

* Activités
* Transitions
* Objets

Une activité est un comportement qui décrit un séquencement organise d’actions. Le flot d’exécution est modalisée par les nœuds relies par des transitions. Ces dernières matérialisent le passage d’une activité vers une autre. Elles sont représentées par des flèches en traits pleins qui connectent les activités entre elles. Le flot de contrôle reste dans l’activité jusqu’à la fin de l’exécution des traitements.

Il existe trois familles de nœuds d’activités : les nœuds exécutables ou d’exécution, les nœuds objets et les nœuds de contrôle.

Parmi les concepts spécifiques au diagramme d’activité, on peut noter :

* Le nœud de bifurcation
* Le nœud de jonction
* Le nœud de fusion
* Le pin d’entrée et de sortie
* Le flot d’objet
* La partition

#### Autres diagrammes

Les autres diagrammes structurels proposés par UML 2 sont :

* Diagramme d’objets
* Diagramme de déploiement
* Diagramme de paquetage
* Diagramme de structure composite

Les diagrammes de comportement sont focalisés sur la description de la partie dynamique du système à modéliser. Les autres proposent par UML 2 sont :

* Diagramme d’état-transition (DET)
* Diagramme de communication (DCO)
* Diagramme global d’interaction (DGI)
* Diagramme de temps (DTP).

## Modélisation de notre application