

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université A. Mira de Béjaia  
Faculté des Sciences Exactes  
Département d'Informatique



# Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master Professionnel en Génie Logiciel

---

## Conception et Réalisation d'une Application Web de Gestion de File d'Attente Numérique

---

**Réalisé par :**

Mme GANDOUR Amina et Mme BENKHALIFA Rania

**Devant le jury composé de :**

**Président** : Dr. GHANEM Souhila  
**Examineur** : Dr. BOUADEM Nassima  
**Encadrante** : Dr. GASMI Badrina

Promotion : 2023/2024

---

## **Remerciement**

*Nous tenons tout d'abord à exprimer notre profonde gratitude à Dieu tout-Puissant pour nous avoir donné la force, la patience et la sagesse nécessaires pour mener à bien ce travail. Nous remercions également du fond du cœur nos parents et notre famille pour leur soutien inébranlable, leur amour et leurs sacrifices constants qui ont été une source inestimable de motivation et de courage.*

*Nos sincères remerciements vont également à notre encadrante Madame **GASMI Badrina** pour ses conseils avisés, son encadrement précieux et son soutien tout au long de ce projet.*

*Nous tenons à exprimer notre gratitude aux membres du jury pour avoir accepté d'évaluer ce travail et pour leurs précieuses remarques et suggestions.*

*Enfin, nous remercions chaleureusement Monsieur **MELLOUK Cherif**, responsable de l'agence web **Techsys Media**, pour son soutien et sa collaboration durant ce projet.*

---

## *Dédicaces*

*À mes chers parents,*

*Aucune dédicace ne saurait exprimer pleinement ma reconnaissance envers vous, quels que soient les termes embrassés. Votre amour inconditionnel, votre bienveillance, votre constante présence à mes côtés, et vos encouragements inlassables, qui ont été ma source de bonheur et de persévérance, je vous dédie ce travail avec une reconnaissance profonde et éternelle.*

*À ma famille, à mes cousines Ibtissem et Bouchra,*

*Votre affection, votre soutien moral et mental incessant ont été des piliers inestimables durant toute cette période et un réconfort inégalé.*

*Que la vie vous récompense par des moments de bonheur et de réussite, et que notre lien familial continue de nous unir et de nous enrichir mutuellement.*

***GANDOUR Amina***

## ***Dédicaces***

*Je souhaite dédier ce travail à mes chers parents, dont l'amour inconditionnel et le soutien indéfectible ont illuminé chacune de mes avancées. Vous êtes mes mentors et ma source d'inspiration, pour toutes les fois où vous avez cru en moi et m'avez poussé à atteindre mes objectifs.*

*À ma famille, dont la présence et le soutien ont rendu chaque défi plus surmontable et chaque succès plus significatif. Chaque jour, je suis reconnaissante d'avoir une famille aussi solidaire et aimante à mes côtés.*

*À mon frère Lamine et à mes sœurs Lamia, Lilia et Imene, qui sont mes piliers et mes compagnons de vie. Leur soutien inconditionnel et leur affection sont des trésors que je chéris profondément. Merci d'être toujours là pour moi, de m'encourager et de partager avec moi chaque moment de joie et de difficulté.*

*À mes chères cousines Melissa et Yasmine, dont la présence et le soutien ont enrichi chaque moment de cette expérience. Merci d'avoir été là pour moi, de m'inspirer et de me soutenir avec tant de générosité et d'affection.*

***BENKHALIFA Rania***

## **Table des matières**

<b>Liste des figures</b>	<b>10</b>
<b>Liste des tableaux</b>	<b>12</b>
<b>Glossaire</b>	<b>13</b>

<b>Introduction générale</b>	<b>1</b>
<b>1 Cadre général :</b>	<b>3</b>
1.1 Introduction : . . . . .	3
1.2 Technologies web : . . . . .	3
1.2.1 Applications Web et Sites Web : . . . . .	3
1.3 Présentation des files d’attente : . . . . .	4
1.3.1 Historique des files d’attente numériques : . . . . .	5
1.3.2 Types de file d’attente : . . . . .	5
1.3.3 Cas d’usage des files d’attente : . . . . .	6
1.4 Contexte et objectif : . . . . .	7
1.5 Problématique et Buts : . . . . .	8
1.6 Méthodologie de conception : . . . . .	8
1.6.1 Méthode de Conception UP : . . . . .	9
1.6.2 Application de la méthode UP sur notre projet : . . . . .	9
1.6.3 UML( Unified Modeling Language ) : . . . . .	10
1.7 Conclusion : . . . . .	12
<b>2 Analyse des besoins :</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Identification des acteurs : . . . . .	13
2.3 Spécification des besoins : . . . . .	13
2.3.1 Besoins fonctionnels : . . . . .	14
2.3.2 Besoins non fonctionnels : . . . . .	15
2.4 Diagramme de contexte : . . . . .	16
2.5 Identification des cas d’utilisation : . . . . .	17

<b>2.6</b>	<b>Diagramme de cas d'utilisation associé à chaque acteur :</b>	<b>19</b>
<b>2.6.1</b>	<b>Description textuelle des cas d'utilisation :</b>	<b>22</b>
<b>2.7</b>	<b>conclusion :</b>	<b>30</b>
<b>3</b>	<b>Conception :</b>	<b>31</b>
<b>3.1</b>	<b>Introduction :</b>	<b>31</b>
<b>3.2</b>	<b>Diagrammes de séquence système :</b>	<b>31</b>
<b>3.2.1</b>	<b>Cas d'utilisation « Authentification » :</b>	<b>32</b>
<b>3.2.2</b>	<b>Cas d'utilisation « Gérer guichetier » :</b>	<b>33</b>
<b>3.2.3</b>	<b>Cas d'utilisation « Gérer services » :</b>	<b>34</b>
<b>3.2.4</b>	<b>Cas d'utilisation « Gérer l'interface de prise de tickets » :</b>	<b>35</b>
<b>3.2.5</b>	<b>Cas d'utilisation « Gérer l'écran d'affichage » :</b>	<b>36</b>
<b>3.2.6</b>	<b>Cas d'utilisation « Appeler Ticket » :</b>	<b>37</b>
<b>3.2.7</b>	<b>Cas d'utilisation « Visualiser Services » :</b>	<b>38</b>
<b>3.2.8</b>	<b>Cas d'utilisation « Voir Historique » :</b>	<b>38</b>
<b>3.2.9</b>	<b>Cas d'utilisation « Prise de tickets en ligne » :</b>	<b>39</b>
<b>3.2.10</b>	<b>Cas d'utilisation « Prise de tickets sur place » :</b>	<b>39</b>
<b>3.2.11</b>	<b>Cas d'utilisation « Suivre la file d'attente » :</b>	<b>40</b>
<b>3.3</b>	<b>Diagrammes de classe :</b>	<b>41</b>
<b>3.3.1</b>	<b>Description détaillée des attributs des classes :</b>	<b>42</b>
<b>3.3.2</b>	<b>Modèle Relationnel</b>	<b>43</b>
<b>3.4</b>	<b>Conclusion :</b>	<b>44</b>
<b>4</b>	<b>Réalisation et Implémentation :</b>	<b>45</b>
<b>4.1</b>	<b>Introduction :</b>	<b>45</b>
<b>4.2</b>	<b>Charte Graphique :</b>	<b>45</b>

<b>4.2.1</b>	<b>Logo et Slogan de WaitWell :</b>	<b>45</b>
<b>4.2.2</b>	<b>Palette de couleur :</b>	<b>46</b>
<b>4.3</b>	<b>Outils de travail :</b>	<b>46</b>
<b>4.4</b>	<b>Technologies utilisées :</b>	<b>47</b>
<b>4.5</b>	<b>Présentation de WailWell :</b>	<b>48</b>
<b>4.5.1</b>	<b>Page d'accueil :</b>	<b>49</b>
<b>4.5.2</b>	<b>Page d'authentification :</b>	<b>50</b>
<b>4.5.3</b>	<b>Page d'authentification sur un compte guichetier désactivé :</b>	<b>50</b>
<b>4.5.4</b>	<b>Tableau de bord de l'administrateur :</b>	<b>51</b>
<b>4.5.5</b>	<b>Interface de gestion de guichetiers :</b>	<b>52</b>
<b>4.5.6</b>	<b>Interface d'ajout de guichetiers :</b>	<b>52</b>
<b>4.5.7</b>	<b>Page de modification d'un compte guichetier :</b>	<b>53</b>
<b>4.5.8</b>	<b>Interface de gestion de services :</b>	<b>53</b>
<b>4.5.9</b>	<b>Interface d'ajout de services :</b>	<b>54</b>
<b>4.5.10</b>	<b>Interface de modification de services :</b>	<b>55</b>
<b>4.5.11</b>	<b>Page de gestion de l'interface de prise de tickets :</b>	<b>56</b>
<b>4.5.12</b>	<b>Interface de prise de ticket :</b>	<b>56</b>
<b>4.5.13</b>	<b>Interface de gestion de l'écran d'affichage :</b>	<b>57</b>
<b>4.5.14</b>	<b>Interface de l'écran d'affichage :</b>	<b>58</b>
<b>4.5.15</b>	<b>Interface de modification du profil de l'administrateur et de guichetier :</b>	<b>58</b>
<b>4.5.16</b>	<b>Page d'accueil du guichetier :</b>	<b>59</b>
<b>4.5.17</b>	<b>L'interface des services affectés au guichetier :</b>	<b>60</b>
<b>4.5.18</b>	<b>Page d'appel de tickets :</b>	<b>60</b>
<b>4.5.19</b>	<b>Page de l'Historique des tickets traités :</b>	<b>61</b>

<b>4.5.20</b>	<b>L'interface de suivi de l'état de la file d'attente :</b>	<b>61</b>
<b>4.5.21</b>	<b>Le ticket :</b>	<b>62</b>
<b>4.6</b>	<b>Conclusion :</b>	<b>63</b>
	<b>Conclusion Générale :</b>	<b>64</b>

# Table des figures

1.1	Un processus itératif et incrémental. . . . .	9
1.2	Les types des diagrammes UML . . . . .	11
2.1	Diagramme de contexte . . . . .	16
2.2	Le diagramme de cas d'utilisation associé à l'administrateur . . . . .	20
2.3	Le diagramme de cas d'utilisation associé au guichetier . . . . .	21
2.4	Le diagramme de cas d'utilisation associé au client . . . . .	21
3.1	Diagramme de séquence du cas d'utilisation « Authentification » . . . . .	32
3.2	Diagramme de séquence du cas d'utilisation « Gérer guichetier » . . . . .	33
3.3	Diagramme de séquence du cas d'utilisation « Gérer services » . . . . .	34
3.4	Diagramme de séquence du cas d'utilisation « Gérer l'interface de prise de tickets » . . . . .	35
3.5	Diagramme de séquence du cas d'utilisation « Gérer l'écran d'affichage » . . . . .	36
3.6	Diagramme de séquence du cas d'utilisation « Appeler Ticket » . . . . .	37
3.7	Diagramme de séquence du cas d'utilisation « Visualiser Services » . . . . .	38
3.8	Diagramme de séquence du cas d'utilisation « Voir Historique » . . . . .	38
3.9	Diagramme de séquence du cas d'utilisation « Prise de tickets en ligne » . . . . .	39
3.10	Diagramme de séquence du cas d'utilisation « Prise de tickets sur place » . . . . .	39
3.11	Diagramme de séquence du cas d'utilisation « Suivre la file d'attente » . . . . .	40
3.12	Diagramme de classe . . . . .	41

4.1	Logo et Slogan . . . . .	45
4.2	Palette de couleur de WaitWell . . . . .	46
4.3	Logo Framework Laravel . . . . .	47
4.4	Logo js . . . . .	47
4.5	Logo HTML . . . . .	47
4.6	Logo css . . . . .	47
4.7	Logo PHP . . . . .	48
4.8	Logo MYSQL . . . . .	48
4.9	Logo Bootstrap . . . . .	48
4.10	Page d'accueil . . . . .	49
4.11	Page de connexion . . . . .	50
4.12	Page d'authentification sur un compte guichetier désactivé . . . . .	51
4.13	Tableau de bord de l'administrateur . . . . .	51
4.14	Gestion des guichetier . . . . .	52
4.15	ajouter des guichetier . . . . .	52
4.16	Modifier un compte guichetier . . . . .	53
4.17	Gestion des services . . . . .	54
4.18	Ajouter des services . . . . .	55
4.19	Modifier des services . . . . .	55
4.20	Interface de sélection des services disponibles . . . . .	56
4.21	Interface de prise de ticket . . . . .	57
4.22	Interface de gestion de l'écran d'affichage . . . . .	57
4.23	L'interface de l'écran d'affichage . . . . .	58
4.24	Modifier Le profil de l'administrateur . . . . .	59

4.25 La page d'accueil du guichetier . . . . .	60
4.26 Les services affectés au guichetier . . . . .	60
4.27 Appeler un ticket . . . . .	61
4.28 Historique des tickets traités . . . . .	61
4.29 L'interface de suivi de l'état de la file d'attente par le client : . . . . .	62
4.30 Le ticket . . . . .	62

# Liste des tableaux

2.1	Besoins fonctionnels de l'application . . . . .	14
2.2	Messages échangés entre les acteurs et le système . . . . .	17
2.3	Identification des cas d'utilisation par acteur . . . . .	18
2.4	Description textuelle du cas « authentification » . . . . .	22
2.5	Description textuelle du cas « Créer un compte guichetier » . . . . .	23
2.6	Description textuelle du cas « Modifier un compte guichetier » . . . . .	23
2.7	Description textuelle du cas « Supprimer un compte guichetier » . . . . .	24
2.8	Description textuelle du cas « Activer/désactiver un compte guichetier » . . . . .	24
2.9	Description textuelle du cas « Ajouter un service » . . . . .	25
2.10	Description textuelle du cas « Modifier un service » . . . . .	25
2.11	Description textuelle du cas « Supprimer un service » . . . . .	26
2.12	Description textuelle du cas « Gérer l'interface de prise de tickets » . . . . .	26
2.13	Description textuelle du cas « Gérer l'écran d'affichage » . . . . .	27
2.14	Description textuelle du cas « Modifier son profil » . . . . .	28
2.15	Description textuelle du cas « Appeler un ticket » . . . . .	28
2.16	Description textuelle du cas « Visualiser les services » . . . . .	29
2.17	Description textuelle du cas « Voir l'historique des tickets traités » . . . . .	29
2.18	Description textuelle du cas « Prendre un ticket en ligne » . . . . .	29

2.19	Description textuelle du cas « Prendre un ticket sur place » . . . . .	30
2.20	Description textuelle du cas « Suivre la file d’attente » . . . . .	30
3.1	Description détaillée des attributs des classes . . . . .	42

<b>API</b>	Application Programming Interface
<b>CC</b>	Creative Cloud
<b>CSS</b>	Cascading Style Sheet
<b>HTML</b>	HyperText Markup Language
<b>JS</b>	JavaScript
<b>PHP</b>	Hypertext Preprocessor
<b>IoT</b>	Internet of Things
<b>UML</b>	Unified Modeling Language
<b>UP</b>	Unified Process
<b>Git</b>	Version Control System (nom propre, pas d'abréviation)
<b>SMS</b>	Short Message Service
<b>IHM</b>	Interface Homme-Machine
<b>VSCODE</b>	Visual Studio Code
<b>WAMP</b>	Windows, Apache, MySQL, PHP
<b>ORM</b>	Object-Relational Mapping
<b>MVC</b>	Modèle-Vue-Contrôleur
<b>SQL</b>	Structured Query Language

## INTRODUCTION GÉNÉRALE :

Les files d'attente sont omniprésentes dans notre quotidien. Elles varient de l'attente dans une banque, dans un magasin, en guichet de poste, cinéma, les stations d'essence, chez le médecin, ...etc. En principe, dans un processus de file d'attente, si le client arrive et trouve le serveur occupé, il joint la file d'attente, ensuite, à un certain instant un client est sélectionné pour être servi suivant une règle dite politique ou discipline de service. Par définition une file d'attente représente toute situation dans laquelle des entités sont introduites dans un système où elles vont être traitées et servis pendant une certaine période dans un certain ordre [5].

Le phénomène de l'attente provoque chez les clients de l'impatience, ce qui peut entraîner une détérioration de leur expérience et une perception négative de l'entreprise. Cette impatience se répercute souvent sur la qualité de service, car le personnel, dépassé par la gestion des files d'attente, peut devenir moins efficace et plus stressé. Ainsi, les files d'attente jouent un rôle crucial dans le flux de la clientèle et ont un impact dans la satisfaction du client, la productivité et les performances globales de l'entreprise.

De ce fait certaines entreprises engagent des agents qui se chargent de la gestion des files d'attente ; c'est une solution efficace que ce soit du côté du personnel ou bien des clients, mais ça reste une charge financière en plus pour le responsable. C'est là que notre projet devient une solution qui permet de résoudre les problèmes liés à ce phénomène, en offrant un système de gestion de file d'attente automatisé sans nécessiter de personnel supplémentaire et en améliorant l'expérience client lui permettant de s'intégrer à la file d'attente sur place ou à distance, mais aussi en réduisant la charge de travail du personnel grâce à des tableaux de bord interactifs et des systèmes de communication automatisée, notre solution libère le personnel des tâches

répétitives et chrono-phages, leur permettant de se concentrer sur des tâches à plus forte valeur ajoutée.

Dans le premier chapitre, intitulé **Cadre Général**, nous présenterons les technologies web et les files d'attente. Nous définirons également le contexte et l'objectif de notre projet, poserons la problématique et proposerons une solution concrète à travers notre projet. Nous expliquerons aussi la méthode de conception ainsi que les différentes approches que nous avons utilisées.

Dans le deuxième chapitre, consacré à l'**Analyse des Besoins**, nous détaillerons les besoins fonctionnels et non fonctionnels. Nous identifierons les acteurs impliqués ainsi que les interactions entre ces acteurs et le système. Ensuite, nous élaborerons le diagramme de contexte et nous présenterons les différents cas d'utilisation à l'aide de diagrammes de cas d'utilisation, accompagnés de descriptions textuelles exhaustives pour chaque cas.

Dans le troisième chapitre, qui concerne la **conception**, nous élaborerons les diagrammes de séquence ainsi que le diagramme de classe suivi d'une description des attributs des classes et enfin, nous inclurons le modèle relationnel.

Dans le chapitre consacré à la **réalisation**, nous présenterons la charte graphique de notre application. Ensuite, nous mettrons en avant les technologies et les outils que nous avons utilisés, et enfin, nous présenterons l'application à travers les interfaces utilisateur existantes.

# CHAPITRE 1

## CADRE GÉNÉRAL :

### 1.1 Introduction :

Dans ce chapitre nous présenterons vaguement les technologies web, ensuite, nous aborderons le sujet des files d'attente en retraçant leur historique et en fournissant quelques définitions clés pour mieux comprendre leur fonctionnement et leur utilité. Par la suite, nous présenterons l'objectif et le contexte de notre projet nous allons poser la problématique. Enfin, nous définirons la méthode de conception et les approches adoptées pour le développement de notre projet, cela inclura une justification de nos choix méthodologiques pour garantir l'efficacité et la pertinence de notre solution..

### 1.2 Technologies web :

Les technologies web désignent l'ensemble des outils, langages et protocoles utilisés pour concevoir, développer et déployer des applications et des sites web. Cela inclut notamment : les langages de programmation web, les frameworks et bibliothèques JavaScript, les serveurs web, les protocoles de communication, les bases de données, les outils de gestion de versions comme Git et les technologies d'authentification, de sécurité et de déploiement [2].

Les technologies web sont un ensemble de composant permettant de réaliser des solutions web. Elles évoluent constamment pour répondre aux nouveaux usages et besoins du web [2].

#### 1.2.1 Applications Web et Sites Web :

##### 1.2.1.1 Sites web :

Un site web est une collection de pages web reliées entre elles et placées sur un nom de domaine unique. Les sites web peuvent être statiques (affichant des informations fixes) ou dynamiques (fonctionnels et interactifs). Ils sont accessibles via un navigateur web à partir de divers appareils.

##### 1.2.1.1.1 Types de site Web :

Nous distinguons deux types de sites web, à savoir les sites statiques et les sites dynamiques [14].

- **Sites Statiques :**

Ce sont des sites utilisés pour afficher des informations à l'utilisateur final. Le contenu de ces sites est fixe et ils sont développés en utilisant le langage HTML. Les sites statiques sont généralement plus rapides à charger et plus faciles à héberger, car ils ne nécessitent pas de traitement côté serveur. Ils sont idéaux pour les sites présentant des informations qui ne changent pas fréquemment, comme les portfolios, les pages de présentation d'entreprise, ou les blogs personnels sans fonctionnalités interactives. Leur simplicité les rend également plus sécurisés, car il y a moins de vulnérabilités potentielles par rapport aux sites dynamiques.

- **Sites Dynamiques :**

Ce sont des sites fonctionnels et interactifs. Ils sont développés à l'aide de langages tels que JavaScript ou PHP. Contrairement aux sites statiques, le contenu des sites dynamiques peut être mis à jour en temps réel, souvent en fonction des actions de l'utilisateur ou des modifications dans une base de données. La majorité des grands sites web sont dynamiques car ils doivent afficher un contenu personnalisé et interactif, tel que des comptes d'utilisateurs, des paniers d'achat, des systèmes de gestion de contenu (CMS) et des plateformes de réseaux sociaux. Ces sites utilisent souvent des technologies avancées comme AJAX pour améliorer l'expérience utilisateur en permettant des mises à jour de contenu sans rechargement complet de la page.

### 1.2.1.2 Applications Web :

Une application web (ou web app) est une application hébergée sur un serveur et accessible depuis un navigateur web, aucune installation n'est nécessaire, ouvrant la porte à de nombreux avantages. Une application web est typiquement utilisée simultanément par plusieurs utilisateurs et est équipée de mécanismes de contrôle d'accès logique, comme l'authentification unique [14].

Elles sont souvent créées par des équipes composées de développeurs, de designers d'interface, d'architectes logiciels et de chefs de produit. Le développement nécessite la connaissance des différents langages utilisés dans les technologies du Web : HTML, CSS, JavaScript, Java, PHP, C#, etc.

Notre projet se concentre sur la conception et le développement d'une application web.

### 1.2.1.3 Différences entre les applications Web et les sites Web :

Les applications web et les sites web présentent des caractéristiques distinctes qui influencent leur conception, leur complexité et leur utilisation.

- Une application web permet aux utilisateurs d'accomplir des tâches spécifiques, tandis qu'un site web fournit et présente principalement de l'information.
- Les applications web sont plus interactives, complexes et nécessitent une équipe expérimentée pour leur développement, tandis que les sites web peuvent être plus simples et moins coûteux à créer.
- Les applications web sont généralement multiplateformes, hébergées dans le cloud et peuvent être utilisées sur n'importe quelle plateforme, tandis que les sites web peuvent être statiques ou dynamiques, affichant un contenu fixe ou interactif adapté à l'utilisateur.

## 1.3 Présentation des files d'attente :

Une file d'attente représente toute situation dans laquelle des entités (comme des personnes, des véhicules, des tâches, etc.) sont introduites dans un système où elles attendent pour être traitées et servies pendant une certaine période, généralement dans un certain ordre.

### 1.3.1 Historique des files d'attente numériques :

Les systèmes de files d'attente numérique, également connus sous le nom de systèmes de gestion de files d'attente, ont évolué au fil du temps en réponse aux besoins croissants de gestion efficace des files d'attente dans divers domaines tels que les entreprises, les institutions publiques, les centres de services, etc.

Voici un aperçu historique de l'évolution de ces systèmes :

- **Début des années 2000 :**

Les premiers systèmes de files d'attente numériques ont été introduits, principalement dans les centres commerciaux et les établissements bancaires. Ces systèmes ont permis aux clients de prendre un ticket numérique et d'attendre leur tour sans avoir à rester physiquement dans la file d'attente.

- **Années 2010 :**

Avec l'avènement des smart-phones et des applications mobiles, de plus en plus de systèmes de files d'attente numériques ont commencé à intégrer des fonctionnalités mobiles. Les clients pouvaient désormais prendre un ticket virtuel via une application mobile plutôt que de devoir utiliser un terminal physique sur place.

- **Années 2010 à nos jours :**

Les avancées technologiques telles que l'intelligence artificielle, l'analyse des données en temps réel et l'Internet des objets (IoT) ont permis aux systèmes de files d'attente numériques d'offrir des fonctionnalités plus avancées. Par exemple, certains systèmes prévoient les temps d'attente en fonction de l'affluence actuelle et des données historiques, permettant ainsi aux clients de planifier leur visite de manière plus efficace.

L'intégration avec d'autres technologies, telles que les écrans d'affichage numérique pour afficher les numéros de file d'attente et les notifications, ainsi que l'intégration avec les systèmes de gestion des rendez-vous, a également augmenté la polyvalence et l'efficacité de ces systèmes.

- **Pandémie de COVID-19 (2020 et au-delà) :**

La pandémie de COVID-19 a accéléré l'adoption des systèmes de files d'attente numériques dans de nombreux secteurs, en particulier dans les établissements recevant du public, tels que les supermarchés, les pharmacies et les centres de vaccination. Ces systèmes ont permis de réduire les contacts physiques et de garantir la distanciation sociale en permettant aux clients de prendre un ticket virtuel ou de réserver un créneau horaire à l'avance.

En résumé, les systèmes de files d'attente numériques ont parcouru un long chemin depuis leurs débuts, passant de simples dispositifs de distribution de tickets à des solutions technologiques avancées intégrant des fonctionnalités mobiles, des analyses prédictives et une intégration avec d'autres systèmes pour offrir une expérience plus fluide et efficace aux clients.

### 1.3.2 Types de file d'attente :

Les deux principaux types de files d'attente sont les files d'attente classiques et les files d'attente numérique. Dans le cadre de notre projet, nous nous concentrerons sur l'étude du cas des files d'attente numérique.

### 1.3.2.1 Files d'attente classiques :

Les files d'attente classiques sont des systèmes traditionnels où les individus attendent dans une lignée pour demander un service. Ces files d'attentes peuvent être physique où les personnes se tiennent en lignée dans un lieu spécifique comme les banques, les postes,...etc.

### 1.3.2.2 Files d'attente numériques :

Une file d'attente numérique est un système qui permet à des individus d'être inclus dans une file d'attente virtuelle par la biais d'un système informatique tel qu'un appareil numérique, des applications web ou mobiles et ce dans le but de demander un service. Elles sont utilisées dans divers contextes au vu de leurs multiples avantages. Il existe plusieurs types de files d'attente numériques :

- **File d'Attente Simple** : Un système où les clients sont servis dans l'ordre d'arrivée garantissant le principe du "premier arrivé, premier servi".
- **File d'Attente Prioritaire** : Un système où certains clients sont traités en priorité en fonction de critères prédéfinis tel que leur statut ou leur besoin, comme dans les files SkyPriority à l'aéroport ou les Fast Pass payant à Disney.
- **File d'Attente Virtuelle** : Un système où les clients peuvent s'inscrire à distance et sont prévenu par mail ou par SMS lorsque leur tour approche, leur permettant de vaquer à leurs occupations en attendant.
- **Files d'attente minutées** : Un système de file d'attente qui donne un temps d'attente estimé ce qui diminue le mécontentement des clients.
- **Files animée** : Un système qui offre des distractions telles que de la musique, des animations diverses, des vidéos ou des jeux, pour rendre l'attente plus agréable.
- **File d'attente Dynamique** : Un système qui offre une solution qui peut être mise en place grâce à des écrans d'affichage dynamique, des bornes interactives, des notifications par SMS ou email, et des systèmes de réservation numérique. L'objectif est de réduire l'anxiété et l'irritabilité des usagers, en leur permettant de s'organiser et de s'occuper pendant le temps d'attente.

Notre projet propose un système de file d'attente numérique qui combine plusieurs de ces types, à savoir prioritaire, virtuelle, minutée et dynamique.

### 1.3.3 Cas d'usage des files d'attente :

Les systèmes de gestion de files d'attente numériques sont utilisés dans divers secteurs pour améliorer l'efficacité et l'expérience client :

- **Les banques** : Les banques utilisent des systèmes de files d'attente numériques pour gérer l'afflux de clients, réduire les temps d'attente et améliorer l'efficacité du service. Les clients peuvent prendre un ticket numérique à leur arrivée et être informés en temps réel de leur position dans la file d'attente, ce qui leur permet de mieux planifier leur temps et de bénéficier d'une expérience client plus fluide.
- **Le secteur de la santé** : Dans le domaine de la santé, les files d'attente numériques sont largement utilisées dans les hôpitaux, les cliniques et les centres de soins pour organiser les rendez-vous, gérer les flux de patients et réduire les temps d'attente. Les patients peuvent prendre rendez-vous en ligne, recevoir des rappels automatiques et être informés en temps réel de l'état de leur file d'attente, ce qui contribue à une meilleure gestion des ressources et à une expérience patient améliorée.

- **Le commerce de détail :** Les commerces de détail intègrent également des systèmes de files d'attente numériques pour optimiser l'expérience client. Que ce soit dans les supermarchés, les magasins de vêtements ou les centres commerciaux, ces systèmes permettent aux clients de prendre un ticket virtuel et de parcourir les rayons en attendant leur tour. Cela réduit la frustration liée aux longues files d'attente, améliore la satisfaction client et favorise les achats impulsifs.
- **Services gouvernementaux :** Les administrations publiques peuvent mettre en place des files d'attente numériques pour gérer l'affluence dans les services tels que les bureaux de l'immigration, les services d'état civil ou les centres de délivrance des permis de conduire. Les citoyens peuvent prendre un ticket en ligne et être alertés lorsque c'est leur tour, évitant ainsi les longues files d'attente et les temps d'attente inutiles.
- **Restauration :** Les restaurants, les cafés et les bars peuvent également bénéficier de systèmes de files d'attente numériques pour gérer les réservations, les commandes à emporter ou les files d'attente pour les tables. Les clients peuvent réserver leur table en ligne, commander leur repas à l'avance et être notifiés lorsque leur commande est prête, ce qui améliore l'efficacité du service et la satisfaction client.
- **Événements :** Lors d'événements tels que des conférences, des concerts ou des salons, les organisateurs peuvent utiliser des files d'attente numériques pour gérer l'accès des participants et éviter les engorgements aux points d'entrée. Les participants peuvent réserver leur place en ligne, télécharger leur billet virtuel et être guidés vers les zones d'accès en fonction de leur numéro de ticket, ce qui garantit une meilleure fluidité des flux et une expérience événementielle plus agréable.

### 1.4 Contexte et objectif :

Notre projet s'intitule « La conception et réalisation d'une application web de gestion de file d'attente numérique ». Cette initiative a été proposée par l'agence Web **Techsys Media**, une entreprise spécialisée en communication et en nouvelles technologies, créée en **2016** à **Béjaia**. Dirigée par monsieur **LEMLOUK Cherif**, Techsys Media dispose d'une équipe de professionnels qualifiés composée de développeurs, de designers et de spécialistes en marketing digital, tous dédiés à fournir des solutions innovantes et adaptées aux besoins de leurs clients.

L'objectif de notre application est d'offrir aux utilisateurs une solution pour rehausser leur expérience en réduisant le temps d'attente perçu et en optimisant l'organisation des files d'attente. Cette démarche permet d'une part, d'améliorer l'expérience des clients visant la satisfaction et la fidélisation et d'autre part, d'adoucir l'environnement de travail du personnel.

## 1.5 Problématique et Buts :

Les files d'attente font partie intégrante de notre quotidien et sont souvent source de conflits et de débordements :

- La satisfaction réduite des clients en raison des temps d'attente prolongés. Par exemple, une personne profitant de sa pause de travail pour payer ses factures peut se retrouver contrainte de repartir sans avoir accompli sa tâche à cause d'une attente excessive.
- La frustration provoquée par les disparités entre les différentes files d'attente. Par exemple : à la poste, certains guichetiers peuvent être plus rapides que d'autres en fonction du nombre de clients ou de leur efficacité, créant une inégalité de traitement perçue.
- La gestion inefficace des files d'attente peut aussi causer des problèmes de flux, incitant parfois les clients à abandonner la file. Par exemple, dans un parc d'attractions, si les files pour les manèges sont trop longues, certains visiteurs peuvent décider de quitter la file pour chercher des attractions moins fréquentées.
- La mauvaise gestion des priorités peut avoir des conséquences graves, dans des situations plus critiques, comme à l'hôpital, : un patient en situation d'urgence qui n'est pas pris en charge à temps peut voir sa condition s'aggraver ou même perdre la vie.
- La détérioration de la qualité du service dû au nombre de clients dépassant la capacité de traitement du personnel disponible, . Cela peut conduire à des erreurs, affectant ainsi l'expérience globale des clients.
- L'accumulation d'une importante file d'attente dans les locaux crée une gêne notable pour les clients et peut altérer leur expérience, affectant ainsi la satisfaction client et la perception de l'efficacité du service.

Ces problèmes soulignent l'importance de mettre en place des stratégies efficaces de gestion des files d'attente pour améliorer la satisfaction des utilisateurs et assurer une répartition équitable des ressources et du temps. Pour cela, nous avons opté pour le développement d'une application de gestion de file d'attente numérique qui permet de :

- Mettre en place un système automatisé de prise de ticket en ligne qui permettent aux clients de réserver leur place à distance, réduisant ainsi le temps passé en attente physique à l'établissement.
- Utiliser des outils de gestion des files d'attente qui équitablement répartissent la charge de travail entre les guichetiers, minimisant ainsi les disparités de temps d'attente perçues par les clients.
- Définir des politiques de priorité claires pour les situations critiques, comme dans les établissements de santé, pour s'assurer que les cas urgents sont traités en priorité, réduisant ainsi les risques pour la santé des patients.
- Informer les clients sur les temps d'attente estimés. Cela aide à gérer les attentes des clients et à améliorer leur expérience en leur permettant de planifier leurs activités en conséquence.
- Mettre en place des fonctionnalités avancées de suivi des performances individuelles, le personnel a des opportunités de développement professionnel.

## 1.6 Méthodologie de conception :

Une méthode de conception définit un formalisme permettant d'exprimer une solution informatique en termes d'architecture logicielle. Ce formalisme définit le modèle conceptuel dans lequel tout élément du

monde réel doit avoir au moins une correspondance [6].

Dans le cadre de notre projet nous avons adopté à la méthode de conception orienté objet [6] qui inclut le procédé de décomposition d’objets et une notation reflétant à la fois les aspects logiques et physiques, tant statique que dynamique des modèle du système à concevoir .

Parmi les méthodes de conception orientée objet disponibles, nous avons spécifiquement choisi d’utiliser le processus unifié (UP) qui est intimement liés au langage de modélisationUML qui fournit les outils essentiels pour la représentation architecture d’information du système assurant ainsi une structure claire et bien structuré compréhensible.

### 1.6.1 Méthode de Conception UP :

**Le processus unifié (UP)** est une méthodologie de développement logiciel itérative et incrémental comme représenté sur la figure ci-dessous (voir Figure 1.1[15]) , qui se distingue par son approche centrée sur l’architecture, pilotée par les cas d’utilisation, et focalisée sur la réduction pro-active des risques. Adaptable à une grande variété de systèmes logiciels, domaines d’application, types d’entreprises et niveaux de compétences, UP offre une flexibilité et une robustesse qui sont essentielles pour le succès des projets modernes.

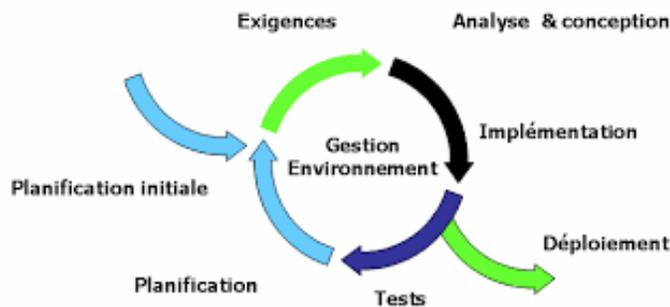


FIGURE 1.1 – Un processus itératif et incrémental.

Cette polyvalence en fait un choix idéal pour notre projet, nous permettant de bénéficier d’une structure méthodologique éprouvée qui garantit une gestion efficace des exigences et des risques. En adoptant le processus unifié, nous nous engageons à suivre une approche rigoureuse et adaptable, qui correspond à notre projet .

### 1.6.2 Application de la méthode UP sur notre projet :

En suivant rigoureusement le cycle de vie de la méthode UP représenté précédemment (voir Figure 1.1[15]), nous avons méthodiquement mis en œuvre chaque phase pour garantir une gestion efficace et structurée de notre projet de gestion de file d’attente numérique.

1. **Planification initiale :** Nous avons commencé par identifier les besoins spécifiques pour notre application de gestion de file d’attente numérique, incluant la possibilité de prendre des tickets en ligne et sur place. Nous avons défini la portée du projet en précisant nos objectifs principaux : optimiser le temps d’attente et améliorer l’expérience utilisateur à travers une gestion efficace.
2. **Analyse des besoins :** En collaboration avec le responsable et notre encadrante, nous avons identifié les fonctionnalités essentielles, telles que la gestion dynamique des files d’attente, l’interaction avec les

comptes guichetiers pour la gestion des services, ainsi que la mise en place d'une interface d'affichage permettant de suivre en temps réel l'état de la file d'attente.

3. **Analyse et conception** : Nous avons élaboré l'architecture logicielle en mettant l'accent sur des composants clés tels que la base de données pour stocker les informations des files d'attente, les interfaces utilisateur intuitives pour faciliter la navigation, et les systèmes de gestion des notifications pour assurer une communication efficace entre l'application et les utilisateurs.
4. **Implémentation** : La phase d'implémentation de notre application de gestion de file d'attente numérique s'est concentrée sur le développement du logiciel selon les spécifications définies. Nous avons utilisé des technologies comme Laravel pour le backend et Bootstrap pour le frontend, en intégrant spécifiquement la prise de ticket en ligne et sur place, ainsi que des fonctionnalités de gestion et d'appel de tickets.
5. **Déploiement** : Nous avons déployé les fonctionnalités sur un environnement de test afin de les évaluer avec les utilisateurs finaux.
6. **Tests** : Cette phase inclut des tests approfondis pour assurer le bon fonctionnement de toutes les fonctionnalités de l'application. Nous avons réalisé des tests unitaires, d'intégration et de système pour garantir une gestion fluide des files d'attente et une expérience utilisateur optimale.
7. **Évaluation** : Après le déploiement et les tests, nous avons évalué l'efficacité réelle de l'application en analysant les retours des utilisateurs. Nous avons identifié les domaines nécessitant des améliorations afin d'optimiser continuellement l'expérience utilisateur et d'assurer une satisfaction client maximale.

### 1.6.3 UML( Unified Modeling Language ) :

Il est très important de procéder à la conception par la modélisation en langages UML (Unified Modeling Language, ou langage de modélisation unifié) qui est distingué par sa richesse sémantique et syntaxiques. Cette dernière consiste à modéliser des éléments du monde réel en un ensemble d'entités informatiques appelées « objet » [15].

L'UML est constitué de diagrammes intégrés utilisés par les développeurs informatiques pour la représentation visuelle des objets, des états et des processus dans un logiciel ou un système visant ainsi à servir de modèle pour un projet et garantir une architecture d'information structurée; il sert également à aider les développeurs à présenter leur description d'un système d'une manière compréhensible pour les spécialistes externes.

IL est principalement utilisé dans le développement de logiciels orientés objet mais à l'heure actuelle Les améliorations apportées à la norme le rendent également adaptée à la représentation des processus de gestion.

#### 1.6.3.1 Les diagrammes de l'UML :

Pour les néophytes, le nombre de diagrammes UML peut paraître infini. En vérité, les normes en identifient 13 types, eux-mêmes répartis en deux groupes, tels que décrits ci-dessous (voir Figure 1.2[15])et au cours de notre projet nous utiliserons spécialement le diagramme de contexte, les diagrammes de cas d'utilisation, les diagramme de séquences ainsi que les diagramme de classes.

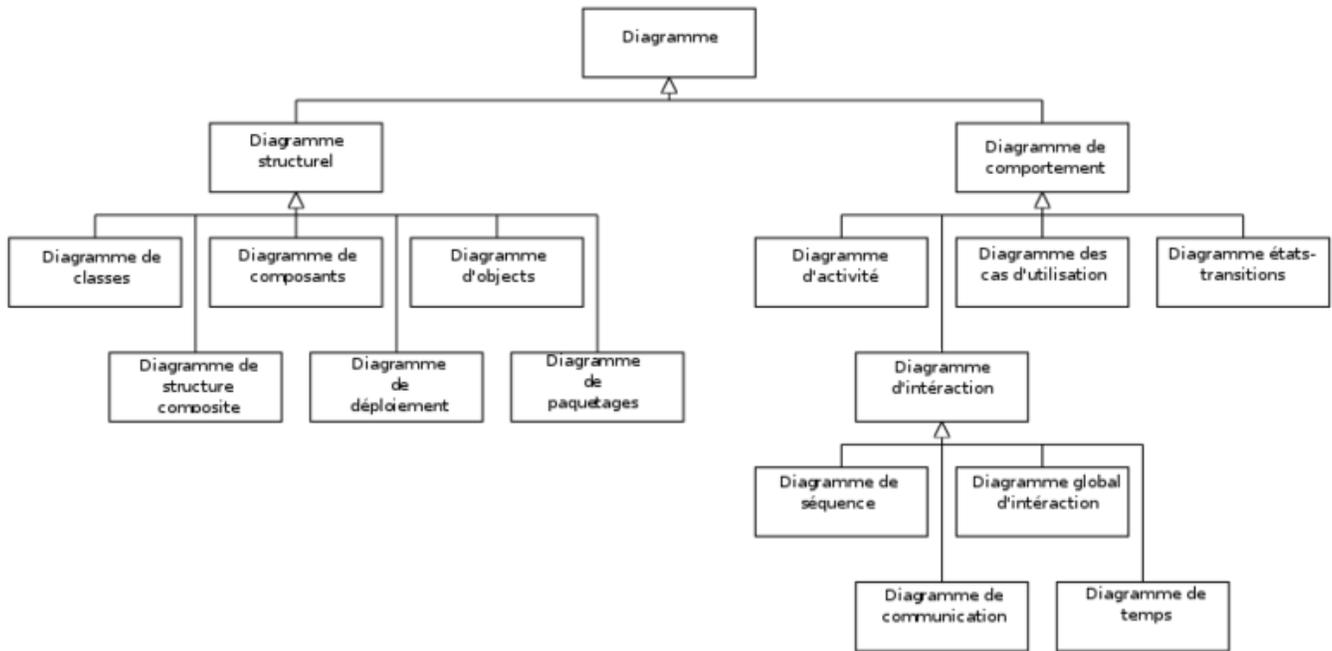


FIGURE 1.2 – Les types des diagrammes UML

### 1.6.3.2 Diagramme de contexte :

Le diagramme de contexte a été créé pour décrire de manière globale le futur système comme une boîte noire. Il se concentre sur l'identification des acteurs principaux qui interagiront avec le système. Si nécessaire, il peut également représenter les interactions entre ces acteurs pour une meilleure compréhension du système. Ce diagramme ne traite ni de l'implémentation ni de la solution, mais uniquement du contexte d'utilisation du système.

### 1.6.3.3 Le diagramme de cas d'utilisation :

Les diagramme de cas d'utilisation représentent les cas d'utilisation, les acteurs et les relations entre les cas d'utilisations et les acteurs[15].

- **Les cas d'utilisation** : représentent les différentes actions ou tâches qu'un utilisateur peut effectuer avec le système

### 1.6.3.4 Diagramme de séquence :

Un diagramme de séquence montrent les interactions entre objets, toutefois, la représentation se concentre sur la séquence des interactions selon un point de vue temporel. Ils sont en général, plus aptes à modéliser les aspects dynamiques des systèmes temps réel et des scénarios complexes mettant en œuvre peu d'objets [15].

### 1.6.3.5 Diagramme de classe :

les diagrammes de classes sont le type de diagramme UML le plus couramment employé. Ils exposent la structure statique d'un système, notamment les classes, leurs attributs et leurs comportements, ainsi que les liens entre chacune d'elles.

## 1.7 Conclusion :

En conclusion de ce chapitre introductif, nous avons posé les bases nécessaires pour comprendre l'environnement dans lequel notre projet s'inscrit. En présentant les technologies web et en explorant le concept des files d'attente, en posant la problématique et en présentant notre solution envisagée, nous avons établi un cadre clair pour la suite de notre travail. Dans le prochain chapitre nous allons effectuer une analyse des besoins en spécifiant les besoins, identifiant les acteurs ainsi que les cas d'utilisation, en élaborant le diagramme de contexte pour comprendre le contexte général de notre système et les diagrammes de cas d'utilisation suivi d'une description textuelle.

## CHAPITRE 2

## ANALYSE DES BESOINS :

### 2.1 Introduction

Dans ce chapitre, nous explorerons les besoins fondamentaux de notre application, en mettant en lumière à la fois ses exigences fonctionnelles et non fonctionnelles, ainsi que les interactions cruciales entre ses utilisateurs et le système lui-même. Ensuite, nous plongerons dans le contexte global de notre application grâce à l'élaboration d'un diagramme de contexte, offrant ainsi une vision d'ensemble en identifiant les acteurs externes et leurs interactions avec notre système. Puis, nous aborderons les fonctionnalités de manière plus détaillée en nous appuyant sur des diagrammes de cas d'utilisation. Ces derniers seront suivis d'une description détaillée de chaque cas d'utilisation.

### 2.2 Identification des acteurs :

Les acteurs sont les entités externes qui interagissent avec le système. Ils peuvent être des personnes, des systèmes externes, des appareils matériels, ou d'autres entités. Les acteurs principaux qui interagissent avec l'application sont :

- **Administrateur** : il a pour rôle la gestion globale du système de file d'attente. Il s'occupe de la gestion des guichetiers, la surveillance de l'activité de la file d'attente la génération de rapport.
- **Guichetier** : Cet utilisateur est responsable de l'interaction directe avec les clients.
- **Client** : son rôle est d'obtenir un ticket et rejoindre la file d'attente pour obtenir un service.

### 2.3 Spécification des besoins :

La spécification des besoins est une étape essentielle qui nous offre une vision d'ensemble des fonctionnalités principales du projet en identifiant avec précision les besoins fonctionnels et non fonctionnels. Cette démarche permet de définir clairement les attentes et les exigences du système, jetant ainsi les bases d'une conception efficace et adaptée aux besoins des utilisateurs.

### 2.3.1 Besoins fonctionnels :

Les spécifications techniques et opérationnelles sont basées sur les cas d'utilisation. Ces derniers, regroupés dans le diagramme de cas d'utilisation, nous ont permis de représenter graphiquement les besoins de l'utilisateur, et chaque cas d'utilisation correspond à une caractéristique du programme. Nous allons citer ci-dessous (voir Tableau 2.1) les besoins fonctionnels de notre application :

Acteur	Besoins Fonctionnels
Administrateur	<ul style="list-style-type: none"> <li>● Permettre l'authentification.</li> <li>● Permettre la gestion des comptes guichetier (ajout, modification, activation/désactivation et suppression).</li> <li>● Permettre la gestion des services (ajout, modification, sélection/désélection et suppression).</li> <li>● Afficher l'état de la file d'attente.</li> <li>● Offrir la possibilité d'afficher du contenu multimédia informationnel ou divertissant pendant l'attente.</li> <li>● Fournir un tableau de bord complet contenant des informations sur les tickets traités et en attente, ainsi que des graphes montrant le nombre de tickets traités par chaque guichetier.</li> </ul>
Guichetier	<ul style="list-style-type: none"> <li>● Permettre l'authentification.</li> <li>● Permettre l'appel du prochain client.</li> <li>● Fournir un tableau de bord interactif.</li> <li>● Afficher l'état de la file d'attente.</li> </ul>
Client	<ul style="list-style-type: none"> <li>● Permettre le suivi de l'état de la file d'attente via une vue dédiée.</li> <li>● Permettre la sélection d'un service et l'enregistrement dans une file d'attente en prenant un ticket sur place ou à distance.</li> </ul>

TABLE 2.1 – Besoins fonctionnels de l'application

### 2.3.2 Besoins non fonctionnels :

Les besoins non fonctionnels précisent essentiellement comment le système devrait se comporter et constituent une contrainte sur le comportement du système. On pourrait aussi considérer les exigences non fonctionnelles comme des attributs de qualité pour un système. Aux besoins cités précédemment, on peut associer les exigences non fonctionnelles suivantes :

- **Performances :**
  - Temps de Réponse : Le système doit avoir un temps de réponse moyen inférieur à 2 secondes.
- **Fiabilité :**
  - Disponibilité du Système : Le système doit être disponible, excluant les périodes de maintenance planifiées.
  - Tolérance aux Pannes : En cas de panne, le système doit être en mesure de reprendre son fonctionnement normal sans perte de données significative.
- **Sécurité :**
  - Authentification et Autorisation : L'application doit exiger une authentification forte, avec un accès basé sur les rôles.
  - Confidentialité des Données : Tous les mots de passe doivent être stockés de manière chiffrée.
- **Extensibilité :**
  - Le système doit être conçu de manière à pouvoir être facilement étendu pour prendre en charge de nouvelles fonctionnalités ou de nouveaux types de documents sans nécessiter de modifications majeures de l'architecture.
- **Maintenabilité :**
  - Le code source du système doit être bien documenté et organisé de manière à faciliter la maintenance et les mises à jour futures.
  - Les modules logiciels doivent être conçus de manière modulaire et cohérente pour permettre des modifications ou des améliorations avec un impact minimal sur le reste du système.

## 2.4 Diagramme de contexte :

Dans cette section, nous présentons le diagramme de contexte (voir Figure 2.1), qui offre une vue d'ensemble des interactions entre notre système et son environnement externe.

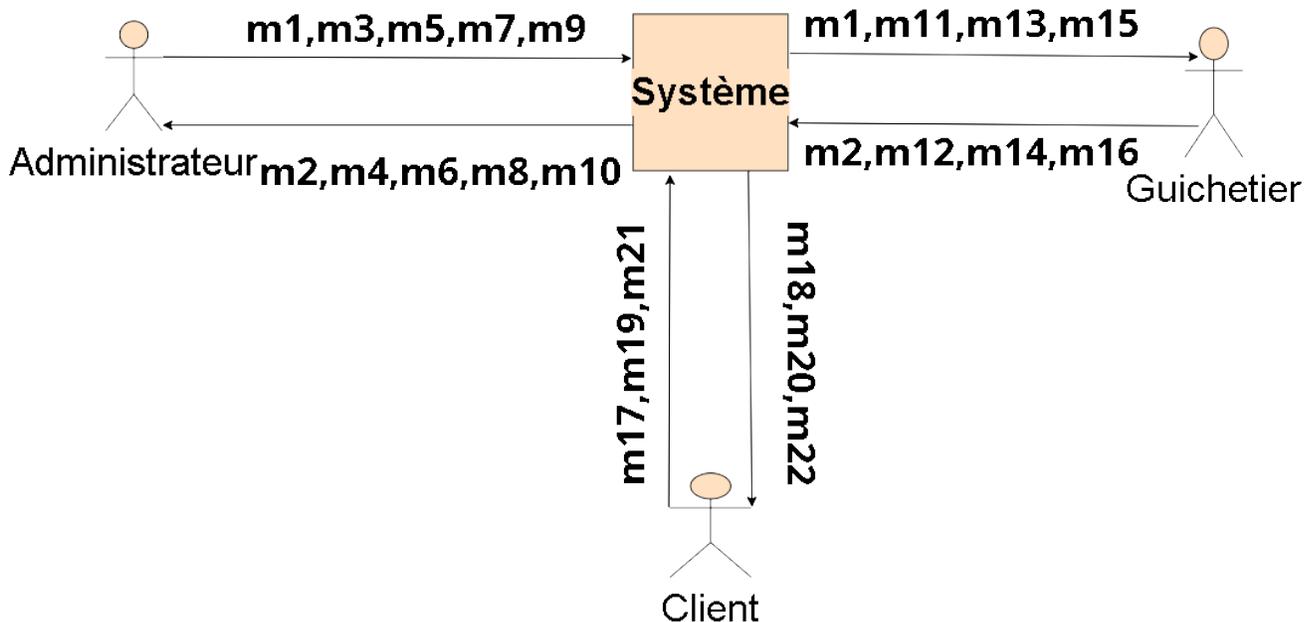


FIGURE 2.1 – Diagramme de contexte

Dans le tableau ci-dessous (voir Tableau 2.2), sont répertoriées les interactions entre les acteurs du système et le système lui-même, sous forme de messages échangés, permettant ainsi de mieux comprendre le diagramme de contexte présenté précédemment.

Acteur	Message acteur -> Système	Message Système -> acteur
Administrateur	<ul style="list-style-type: none"> <li>• <b>m1</b> : Demande d'authentification.</li> <li>• <b>m3</b> : Demande de gestion les guichetiers.</li> <li>• <b>m5</b> : Demande de gestion les services.</li> <li>• <b>m7</b> : Demande de gestion de l'interface de prise de tickets.</li> <li>• <b>m9</b> : Demande de gestion de l'écran d'affichage.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>m2</b> : Affichage de l'espace correspondant ou message d'erreur.</li> <li>• <b>m4</b> : Affichage de la liste des comptes guichetiers avec les options de gestions relatives.</li> <li>• <b>m6</b> : Affichage de la liste des services avec les options de gestion relatives.</li> <li>• <b>m8</b> : Afficher/retirer un service de l'interface de prise de tickets ou afficher l'interface de prise de tickets.</li> <li>• <b>m10</b> : Affichage des paramètres liés à l'interface de l'écran d'affichage.</li> </ul>
Guichetier	<ul style="list-style-type: none"> <li>• <b>m1</b> : Demande d'authentification.</li> <li>• <b>m11</b> : Demande des services qui lui sont affectés.</li> <li>• <b>m13</b> : Demande d'appel de tickets.</li> <li>• <b>m15</b> : Demande de voir l'historique des tickets traités.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>m2</b> : Affichage de l'espace correspondant ou message d'erreur.</li> <li>• <b>m12</b> : Affichage des services qui lui sont affectés.</li> <li>• <b>m14</b> : Affichage de l'espace d'appel de tickets.</li> <li>• <b>m16</b> : Affichage de la liste des tickets qu'il a traités.</li> </ul>
Client	<ul style="list-style-type: none"> <li>• <b>m17</b> : Demande de prendre un ticket en ligne.</li> <li>• <b>m19</b> : Sélectionner un service parmi les services disponibles.</li> <li>• <b>m21</b> : Demande d'impression ou de téléchargement du ticket généré.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>m18</b> : Affichage de l'interface de prise de ticket en ligne.</li> <li>• <b>m10</b> : Génère un ticket avec les informations nécessaires.</li> <li>• <b>m22</b> : Message de succès.</li> </ul>

TABLE 2.2 – Messages échangés entre les acteurs et le système

## 2.5 Identification des cas d'utilisation :

Dans cette section, nous présentons les différents cas d'utilisation de notre système, classés par acteur (voir Tableau 2.3[15]). Chaque cas d'utilisation décrit une action spécifique que l'acteur peut effectuer dans

le système.

Acteur	Cas d'utilisation
Administrateur	<ul style="list-style-type: none"> <li>● S'authentifier.</li> <li>● La gestion des comptes guichetiers : <ul style="list-style-type: none"> <li>— Créer un compte guichetier et lui affecter un/des services.</li> <li>— Modifier un compte guichetier.</li> <li>— Supprimer un compte guichetier.</li> <li>— Activer/désactiver un compte guichetier.</li> </ul> </li> <li>● La gestion des services : <ul style="list-style-type: none"> <li>— Ajouter un service.</li> <li>— Modifier un service.</li> <li>— Supprimer un service.</li> </ul> </li> <li>● La gestion de l'interface de prise de tickets : <ul style="list-style-type: none"> <li>— Sélectionner/désélectionner un service de l'interface de prise de tickets.</li> <li>— Afficher l'interface de prise de tickets.</li> </ul> </li> <li>● Gérer l'écran d'affichage : <ul style="list-style-type: none"> <li>— Ajouter un texte défilant.</li> <li>— Ajouter une vidéo.</li> <li>— Afficher l'écran d'affichage.</li> </ul> </li> <li>● Modifier son profil.</li> </ul>
Guichetier	<ul style="list-style-type: none"> <li>● S'authentifier.</li> <li>● Visualiser ses services.</li> <li>● Appeler un ticket.</li> <li>● Voir l'historique des tickets traités.</li> <li>● Modifier son profil.</li> </ul>
Client	<ul style="list-style-type: none"> <li>● Prendre un ticket sur place.</li> <li>● Prendre un ticket en ligne : <ul style="list-style-type: none"> <li>— Imprimer ou télécharger le ticket.</li> </ul> </li> <li>● Suivre la file d'attente.</li> </ul>

TABLE 2.3 – Identification des cas d'utilisation par acteur

## **2.6 Diagramme de cas d'utilisation associé à chaque acteur :**

Dans cette section, nous présentons les diagrammes de cas d'utilisation pour chaque acteur du système , à savoir l'administrateur(voir Figure 2.2), le guichetier(voir Figure 2.2) et le client(voir Figure 2.4).

2.6.0.1 Cas d'utilisation associé à l'administrateur :

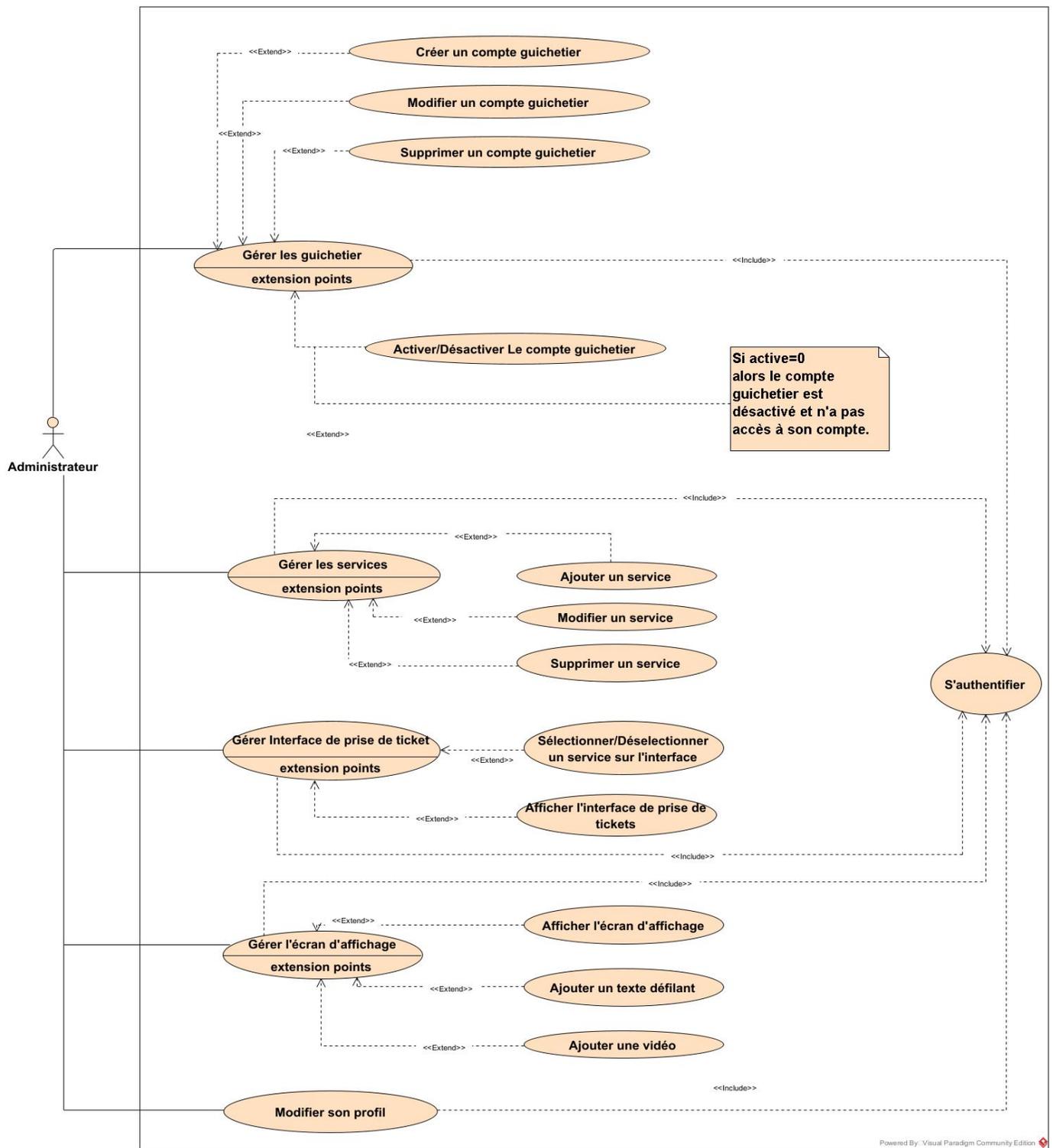


FIGURE 2.2 – Le diagramme de cas d'utilisation associé à l'administrateur

2.6.0.2 Cas d'utilisation associé au guichetier :

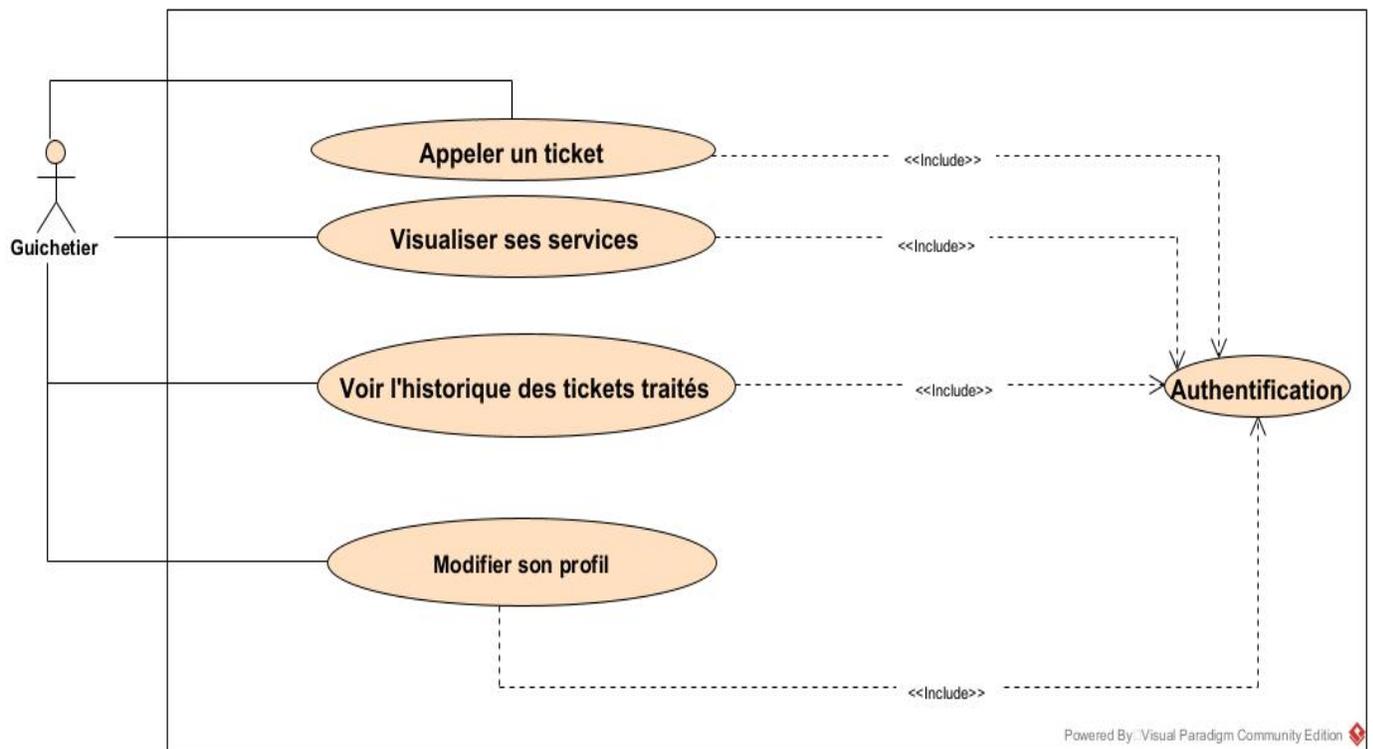


FIGURE 2.3 – Le diagramme de cas d'utilisation associé au guichetier

2.6.0.3 Cas d'utilisation associé au client :

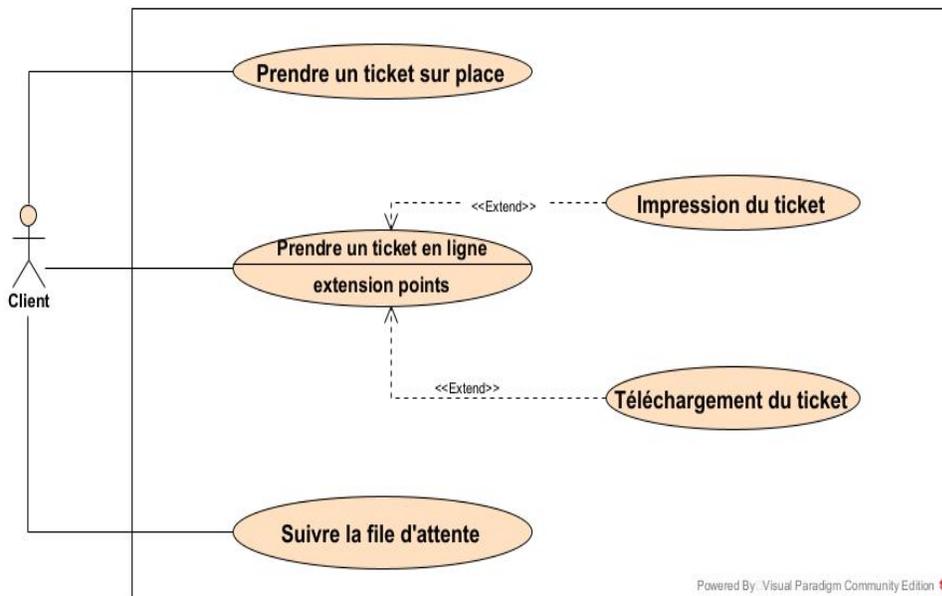


FIGURE 2.4 – Le diagramme de cas d'utilisation associé au client

### 2.6.1 Description textuelle des cas d'utilisation :

Le but d'une description textuelle des cas d'utilisation est de décrire de manière précise et complète les interactions entre l'acteur et le système pour réaliser l'objectif du cas d'utilisation. Ci-dessous nous allons décrire certains cas d'utilisation et pour chaque cas nous exposerons son acteur et scénario nominal ainsi que ses alternatives.

#### 2.6.1.1 Cas d'utilisation « authentification » :

Acteur Principal	Administrateur, Guichetier
Objectifs	S'authentifier pour pouvoir accéder à toutes les fonctionnalités de leur interface.
Pré-conditions	L'utilisateur doit avoir un compte.
Post-conditions	Accès à la page d'accueil de l'utilisateur.
Scénario nominal	<ol style="list-style-type: none"> <li>1. Demande d'accès au système.</li> <li>2. Le système affiche la page d'authentification.</li> <li>3. L'utilisateur saisit son email et son mot de passe.</li> <li>4. Le système vérifie les champs.</li> <li>5. Le système vérifie l'authenticité des informations, l'existence du compte et l'activation du compte.</li> <li>6. L'utilisateur accède à son espace.</li> </ol>
Scénario alternatif	<p>A1. Champs vides. L'enchaînement commence après le point 4 du scénario nominal. 5. Le système affiche un message d'erreur (champ vide). — Le scénario nominal reprend au point 3.</p> <p>A2. Le compte n'existe pas. L'enchaînement commence après le point 5 du scénario nominal. 5. Le système renvoie un message d'erreur (l'utilisateur n'existe pas) et réaffiche la page d'authentification. — Le scénario nominal reprend au point 3.</p> <p>A3. Le compte est désactivé. L'enchaînement commence après le point 5 du scénario nominal. 5. Le système renvoie un message indiquant que le compte est désactivé.</p>

TABLE 2.4 – Description textuelle du cas « authentification »

**2.6.1.2 Cas d'utilisation « Créer un compte guichetier » :**

Acteur Principal	Administrateur
Objectifs	Créer les comptes des guichetiers.
Pré-conditions	L'administrateur doit s'authentifier.
Post-conditions	L'administrateur accède à la liste des guichetiers.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur demande la gestion des guichetiers.</li> <li>2. Le système affiche la liste des guichetiers.</li> <li>3. L'administrateur demande la création d'un compte guichetier.</li> <li>4. Le système lui répond par l'affichage d'un formulaire.</li> <li>5. L'administrateur remplit les champs du formulaire avec les informations demandées et soumet le formulaire.</li> <li>6. Le système vérifie les champs.</li> <li>7. Le système affiche le succès de l'opération et le nouveau compte est ajouté à la liste des guichetiers.</li> </ol>
Scénario Alternatif	<p>A1. Champs vides. L'enchaînement commence après le point 6 du scénario nominal.</p> <ol style="list-style-type: none"> <li>7. Le système affiche un message d'erreur et réaffiche le formulaire.</li> </ol>

TABLE 2.5 – Description textuelle du cas « Créer un compte guichetier »

**2.6.1.3 Cas d'utilisation « Modifier un compte guichetier » :**

Acteur Principal	Administrateur
Objectifs	Modifier le compte du guichetier.
Pré-conditions	L'administrateur doit s'authentifier.
Post-conditions	L'administrateur accède à la liste des guichetiers avec les informations modifiées.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur demande la liste des guichetiers.</li> <li>2. Le système lui répond par l'affichage de la liste des guichetiers.</li> <li>3. L'administrateur choisit de modifier le compte du guichetier.</li> <li>4. Le système affiche un formulaire avec le nom, prénom, adresse e-mail, numéro de téléphone et les services affectés.</li> <li>5. L'administrateur choisit de modifier les informations qu'il souhaite et valide la modification.</li> <li>6. Le système vérifie les champs.</li> <li>7. Le système renvoie la liste des guichetiers avec les informations qui ont été modifiées.</li> </ol>
Scénario Alternatif	<p>A1. Champs vides. L'enchaînement commence après le point 6 du scénario nominal.</p> <ol style="list-style-type: none"> <li>7. Le système affiche un message d'erreur et réaffiche le formulaire.</li> </ol>

TABLE 2.6 – Description textuelle du cas « Modifier un compte guichetier »

**2.6.1.4 Cas d'utilisation « Supprimer un compte guichetier » :**

Acteur Principal	Administrateur
Objectifs	Supprimer le compte du guichetier.
Pré-conditions	L'administrateur doit s'authentifier.
Post-conditions	L'administrateur accède à la liste des guichetiers.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur demande la liste des guichetiers.</li> <li>2. Le système lui répond par l'affichage de la liste des guichetiers.</li> <li>3. L'administrateur choisit de supprimer le compte guichetier.</li> <li>4. Le système envoie un message de confirmation de suppression.</li> <li>5. L'administrateur : <ol style="list-style-type: none"> <li>A. Confirme la suppression.</li> <li>B. Annule la suppression.</li> </ol> </li> <li>6. Le système : <ol style="list-style-type: none"> <li>A. Renvoie la liste des guichetiers sans le compte guichetier supprimé.</li> <li>B. Aucun changement n'est effectué sur la liste des guichetiers.</li> </ol> </li> </ol>

TABLE 2.7 – Description textuelle du cas « Supprimer un compte guichetier »

**2.6.1.5 Cas d'utilisation « Activer/désactiver un compte guichetier » :**

Acteur Principal	Administrateur
Objectifs	Activer/désactiver le compte du guichetier.
Pré-conditions	L'administrateur doit s'authentifier.
Post-conditions	L'administrateur accède à la liste des guichetiers.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur demande la liste des guichetiers.</li> <li>2. Le système lui répond par l'affichage de la liste des guichetiers.</li> <li>3. L'administrateur choisit d'activer/désactiver le compte guichetier.</li> <li>4. Le système change le statut du guichetier : <ul style="list-style-type: none"> <li>— Si le compte est activé, le guichetier a accès à son compte.</li> <li>— Si le compte est désactivé, le guichetier ne peut pas avoir accès à son compte.</li> </ul> </li> </ol>

TABLE 2.8 – Description textuelle du cas « Activer/désactiver un compte guichetier »

**2.6.1.6 Cas d'utilisation « Ajouter un service » :**

Acteur Principal	Administrateur
Objectifs	Ajouter un service.
Pré-conditions	L'administrateur doit s'authentifier.
Post-conditions	L'administrateur accède à la liste des services avec le service ajouté.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur demande la gestion des services.</li> <li>2. Le système répond par l'affichage de la liste des services.</li> <li>3. L'administrateur demande à ajouter un service.</li> <li>4. Le système répond avec l'envoi d'un formulaire.</li> <li>5. L'administrateur remplit le formulaire et valide l'ajout du service.</li> <li>6. Le système vérifie les champs.</li> <li>7. Le système affiche le succès de l'opération et renvoie la liste des services avec le service ajouté.</li> </ol>
Scénario Alternatif	<p>A1. Champs vides.</p> <p>L'enchaînement commence après le point 6 du scénario nominal.</p> <p>7. Le système affiche un message d'erreur et réaffiche le formulaire.</p>

TABLE 2.9 – Description textuelle du cas « Ajouter un service »

**2.6.1.7 Cas d'utilisation « Modifier un service » :**

Acteur Principal	Administrateur
Objectifs	Modifier un service.
Pré-conditions	L'administrateur doit être authentifié.
Post-conditions	L'administrateur accède à la liste des services avec les changements effectués.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur demande la liste des services.</li> <li>2. Le système répond avec la liste des services existants.</li> <li>3. L'administrateur choisit de modifier un service.</li> <li>4. Le système envoie un formulaire de modification de service.</li> <li>5. L'administrateur modifie les informations souhaitées et confirme la modification.</li> <li>6. Le système vérifie les champs.</li> <li>7. Le système renvoie la liste des services avec les changements effectués.</li> </ol>
Scénario Alternatif	<p>A1. Champs vides.</p> <p>L'enchaînement commence après le point 6 du scénario nominal.</p> <p>7. Le système affiche un message d'erreur et réaffiche le formulaire.</p>

TABLE 2.10 – Description textuelle du cas « Modifier un service »

**2.6.1.8 Cas d'utilisation « Supprimer un service » :**

Acteur Principal	Administrateur
Objectifs	Supprimer un service.
Pré-conditions	L'administrateur doit être authentifié.
Post-conditions	L'administrateur accède à la liste des services sans le service supprimé.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur demande la liste des services.</li> <li>2. Le système répond avec la liste des services existants.</li> <li>3. L'administrateur choisit de supprimer un service.</li> <li>4. Le système envoie un message de confirmation de suppression.</li> <li>5. L'administrateur : <ol style="list-style-type: none"> <li>A. Confirme la suppression.</li> <li>B. Annule la suppression.</li> </ol> </li> <li>6. Le système : <ol style="list-style-type: none"> <li>A. Renvoie la liste des services sans le service supprimé.</li> <li>B. Aucun changement n'est effectué sur la liste des services.</li> </ol> </li> </ol>

TABLE 2.11 – Description textuelle du cas « Supprimer un service »

**2.6.1.9 Cas d'utilisation « Gérer l'interface de prise de tickets » :**

Acteur Principal	L'administrateur
Objectifs	Afficher les services disponibles sur l'interface de prise de tickets.
Pré-conditions	L'administrateur doit s'authentifier.
Post-conditions	L'administrateur accède à la liste des services disponibles.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur demande à gérer l'interface de prise de tickets.</li> <li>2. Le système répond par l'affichage de la liste des services.</li> <li>3. L'administrateur choisit : <ol style="list-style-type: none"> <li>A. De sélectionner ou de dé-sélectionner un service de l'interface.</li> <li>B. Ou d'afficher l'interface de prise de tickets.</li> </ol> </li> <li>4. Le système : <ol style="list-style-type: none"> <li>A. Change le statut du service.</li> <li>B. Renvoie l'interface de prise de tickets.</li> </ol> </li> </ol>

TABLE 2.12 – Description textuelle du cas « Gérer l'interface de prise de tickets »

**2.6.1.10 Cas d'utilisation « Gérer l'écran d'affichage » :**

Acteur Principal	L'administrateur.
Objectifs	Gérer l'interface de l'écran d'affichage.
Pré-conditions	L'administrateur doit s'authentifier.
Post-conditions	L'administrateur accède aux paramètres de l'écran.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur demande à gérer l'écran d'affichage.</li> <li>2. Le système répond par l'affichage des paramètres : <ul style="list-style-type: none"> <li>— Insérer l'URL de la vidéo souhaitée.</li> <li>— Insérer un texte défilant.</li> <li>— Afficher l'écran d'affichage.</li> </ul> </li> <li>3. L'administrateur : <ul style="list-style-type: none"> <li>— A : Insère l'URL de la vidéo.</li> <li>— B : Insère un texte défilant.</li> <li>— C : Demande d'afficher l'écran d'affichage.</li> </ul> </li> <li>4. — A et B : L'administrateur soumet les informations insérées. — C : Le système renvoie l'écran d'affichage.</li> <li>5. A et B : Le système vérifie la validité des informations insérées.</li> <li>6. A et B : Le système renvoie un message de succès.</li> </ol>
Scénario Alternatif	<p>A1. Champs invalides. L'enchaînement commence après le point 5 du scénario nominal.</p> <p>6. Le système affiche un message d'erreur et réaffiche l'interface.</p>

TABLE 2.13 – Description textuelle du cas « Gérer l'écran d'affichage »

**2.6.1.11 Cas d'utilisation « Modifier son profil » :**

Acteur Principal	L'administrateur, Le guichetier.
Objectifs	Modifier les informations personnelles et les identifiants.
Pré-conditions	L'utilisateur doit s'authentifier.
Post-conditions	L'utilisateur accède à ses informations personnelles ainsi que ses identifiants.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur demande à modifier son profil.</li> <li>2. Le système affiche les informations personnelles et les identifiants de l'utilisateur.</li> <li>3. L'utilisateur choisit l'information ou l'identifiant à modifier et effectue sa modification.</li> <li>4. Le système vérifie la validité des informations modifiées.</li> <li>5. Le système affiche le succès de la modification.</li> </ol>
Scénario Alternatif	<p>A1. Informations invalides.</p> <p>L'enchaînement commence après le point 4 du scénario nominal.</p> <ol style="list-style-type: none"> <li>7. Le système affiche un message d'erreur et réaffiche le formulaire.</li> </ol>

TABLE 2.14 – Description textuelle du cas « Modifier son profil »

**2.6.1.12 Cas d'utilisation « Appeler un ticket » :**

Acteur Principal	Le guichetier.
Objectifs	Appeler le numéro de ticket du client suivant.
Pré-conditions	Le guichetier doit s'authentifier.
Post-conditions	Le guichetier accède à son interface.
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le guichetier demande l'appel de tickets.</li> <li>2. Le système renvoie l'interface d'appel de tickets.</li> <li>3. Le guichetier demande le ticket suivant.</li> <li>4. Le système renvoie le numéro du ticket suivant.</li> <li>5. Le guichetier appelle le numéro du ticket.</li> <li>6. Le système renvoie les informations du ticket traité à l'historique des tickets traités.</li> </ol>
Scénario Alternatif	<p>A1. Pas de ticket disponible.</p> <p>L'enchaînement commence après le point 3 du scénario nominal.</p> <ol style="list-style-type: none"> <li>4. Le système affiche un message indiquant qu'aucun ticket n'est pas disponible.</li> </ol>

TABLE 2.15 – Description textuelle du cas « Appeler un ticket »

**2.6.1.13 Cas d'utilisation « Visualiser les services » :**

Acteur Principal	Guichetier
Objectifs	Le guichetier visualise les services auxquels il est affecté.
Préconditions	S'authentifier.
Postconditions	Le guichetier accède à la liste des services auxquels il est affecté.
Scénario nominal	1. Le guichetier demande la liste des services auxquels il est affecté. 2. Le système renvoie la liste des services auxquels il est affecté.

TABLE 2.16 – Description textuelle du cas « Visualiser les services »

**2.6.1.14 Cas d'utilisation « Voir l'historique des tickets traités » :**

Acteur Principal	Guichetier
Objectifs	Le guichetier accède à l'historique des tickets qu'il a traités.
Pré-conditions	S'authentifier.
Post-conditions	Le guichetier accède à l'historique des tickets qu'il a traités.
Scénario nominal	1. Le guichetier demande l'historique des tickets qu'il a traités. 2. Le système renvoie la liste des tickets qu'il a traités avec les informations relatives.

TABLE 2.17 – Description textuelle du cas « Voir l'historique des tickets traités »

**2.6.1.15 Cas d'utilisation « Prendre un ticket en ligne » :**

Acteur Principal	Client
Objectifs	Prendre un ticket en ligne sans se déplacer.
Pré-conditions	Le client consulte l'application web sur son navigateur web.
Post-conditions	Le client reçoit son ticket.
Scénario nominal	1. Le client accède à la page d'accueil. 2. Le client demande de prendre un ticket en ligne. 3. Le système renvoie l'interface de prise de tickets en ligne. 4. Le client sélectionne un service parmi les services disponibles. 5. Le système génère un ticket avec le numéro, le nom du service, l'heure et la date, le temps d'attente estimé ainsi que le nombre de clients devant lui. 6. Le client imprime ou télécharge le ticket et se présente à l'établissement muni de ce dernier.

TABLE 2.18 – Description textuelle du cas « Prendre un ticket en ligne »

**2.6.1.16 Cas d'utilisation « Prendre un ticket sur place » :**

Acteur Principal	Client
Objectifs	Prendre un ticket sur place.
Pré-conditions	Le client doit être présent physiquement à l'emplacement où le service de prise de tickets est disponible.
Post-conditions	Le client reçoit son ticket.
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le client sélectionne un service parmi les services disponibles sur l'interface de prise de tickets sur place.</li> <li>2. Le système génère et imprime un ticket.</li> </ol>

TABLE 2.19 – Description textuelle du cas « Prendre un ticket sur place »

**2.6.1.17 Cas d'utilisation « Suivre la file d'attente » :**

Acteur Principal	Client
Objectifs	Suivre l'état de la file d'attente.
Pré-conditions	Le client consulte l'application web sur son navigateur web.
Post-conditions	Le client voit l'état de la file d'attente.
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le client accède à l'application web via son navigateur web.</li> <li>2. Le client clique sur le lien "suivi" pour le suivi de l'état de la file d'attente.</li> <li>3. Le système répond en affichant l'état de la file d'attente avec le numéro de ticket à l'appel suivi du numéro de guichet.</li> </ol>

TABLE 2.20 – Description textuelle du cas « Suivre la file d'attente »

**2.7 conclusion :**

Dans ce chapitre nous avons identifié les acteurs du système, détaillé les besoins fonctionnels et non fonctionnels, et créé le diagramme de contexte. Nous avons également identifié les cas d'utilisation, produit les diagrammes de cas d'utilisation pour chaque acteur, et fourni une description textuelle de chaque cas. Ces analyses posent les bases nécessaires pour la phase de conception, où nous développerons les diagrammes de séquence, les diagrammes de classe, les descriptions des attributs des classes et le modèle relationnel.

## CHAPITRE 3

CONCEPTION :

### 3.1 Introduction :

Dans ce chapitre dédié à la conception, nous explorons en détail les mécanismes internes de notre système à travers l'utilisation de diagrammes de séquence spécifiques à chaque cas d'utilisation. En complément, nous examinons également la structure globale du système à l'aide du diagramme de classe, qui offre une représentation visuelle des entités et de leurs relations.

### 3.2 Diagrammes de séquence système :

Dans cette section, nous examinons de manière détaillée les interactions spécifiques entre les acteurs et le système à travers différents cas d'utilisation représentés par des diagrammes de séquence.

### 3.2.1 Cas d'utilisation « Authentification » :

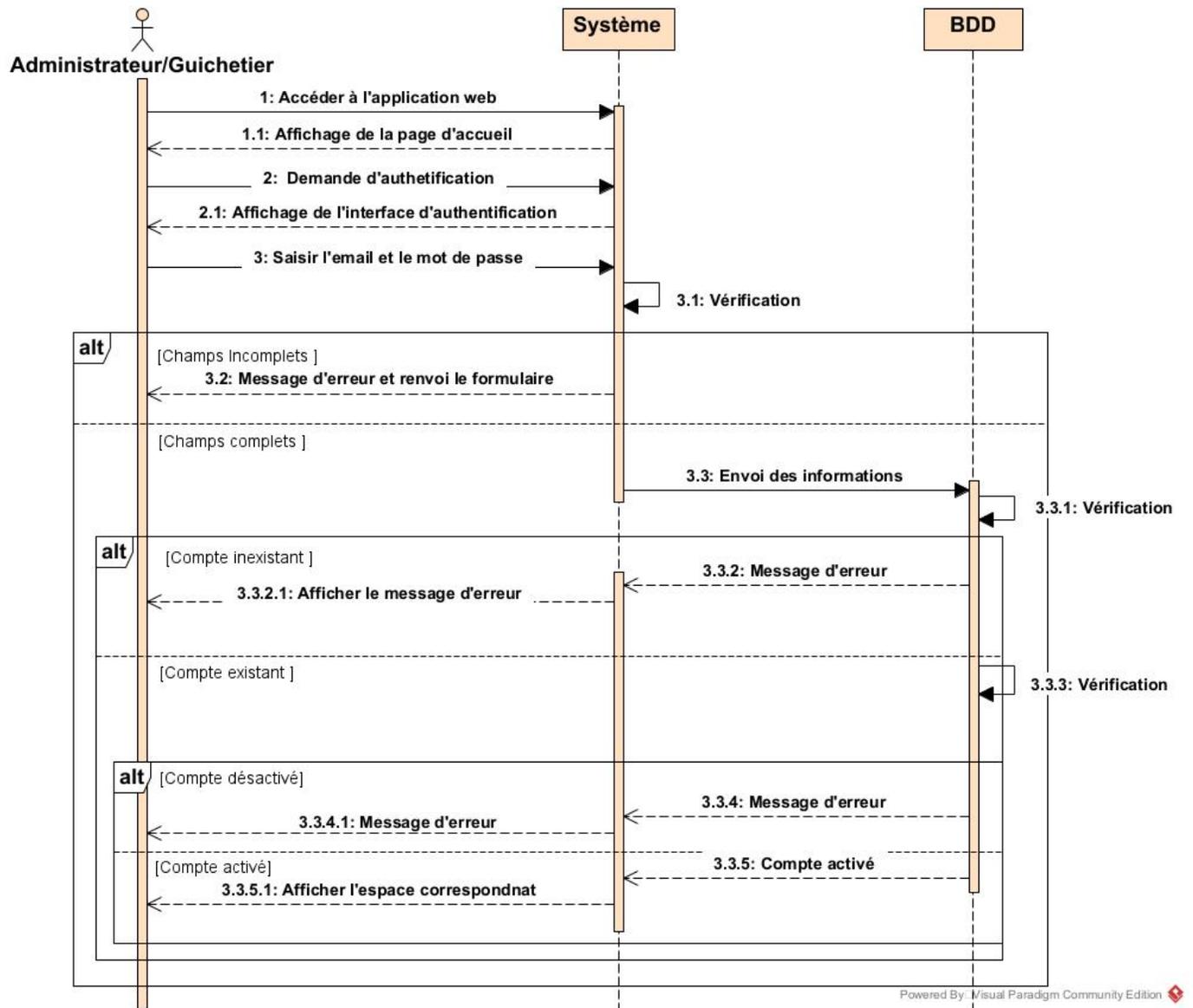


FIGURE 3.1 – Diagramme de séquence du cas d'utilisation « Authentification »

### 3.2.2 Cas d'utilisation « Gérer guichetier » :

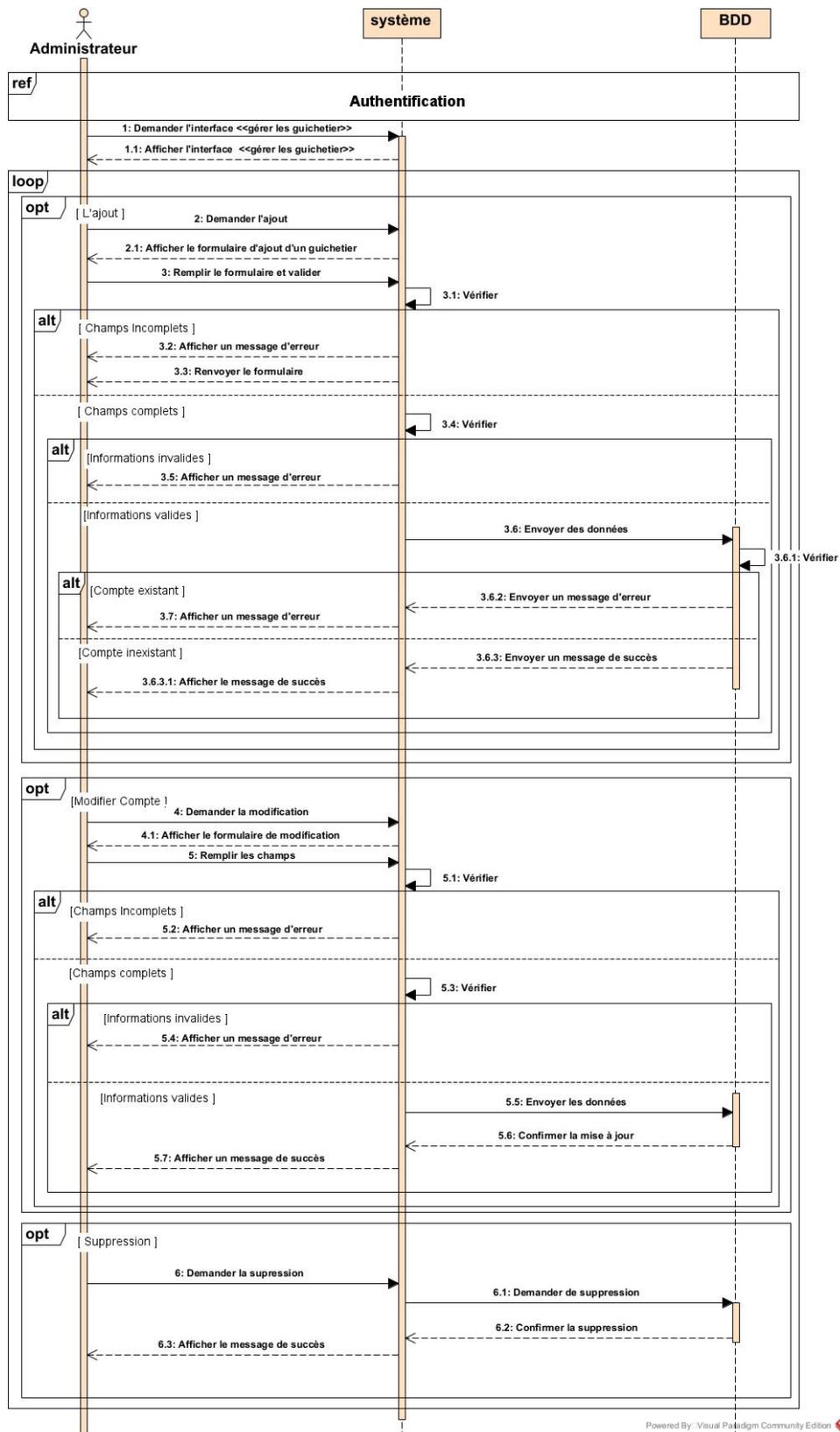


FIGURE 3.2 – Diagramme de séquence du cas d'utilisation « Gérer guichetier »

### 3.2.3 Cas d'utilisation « Gérer services » :

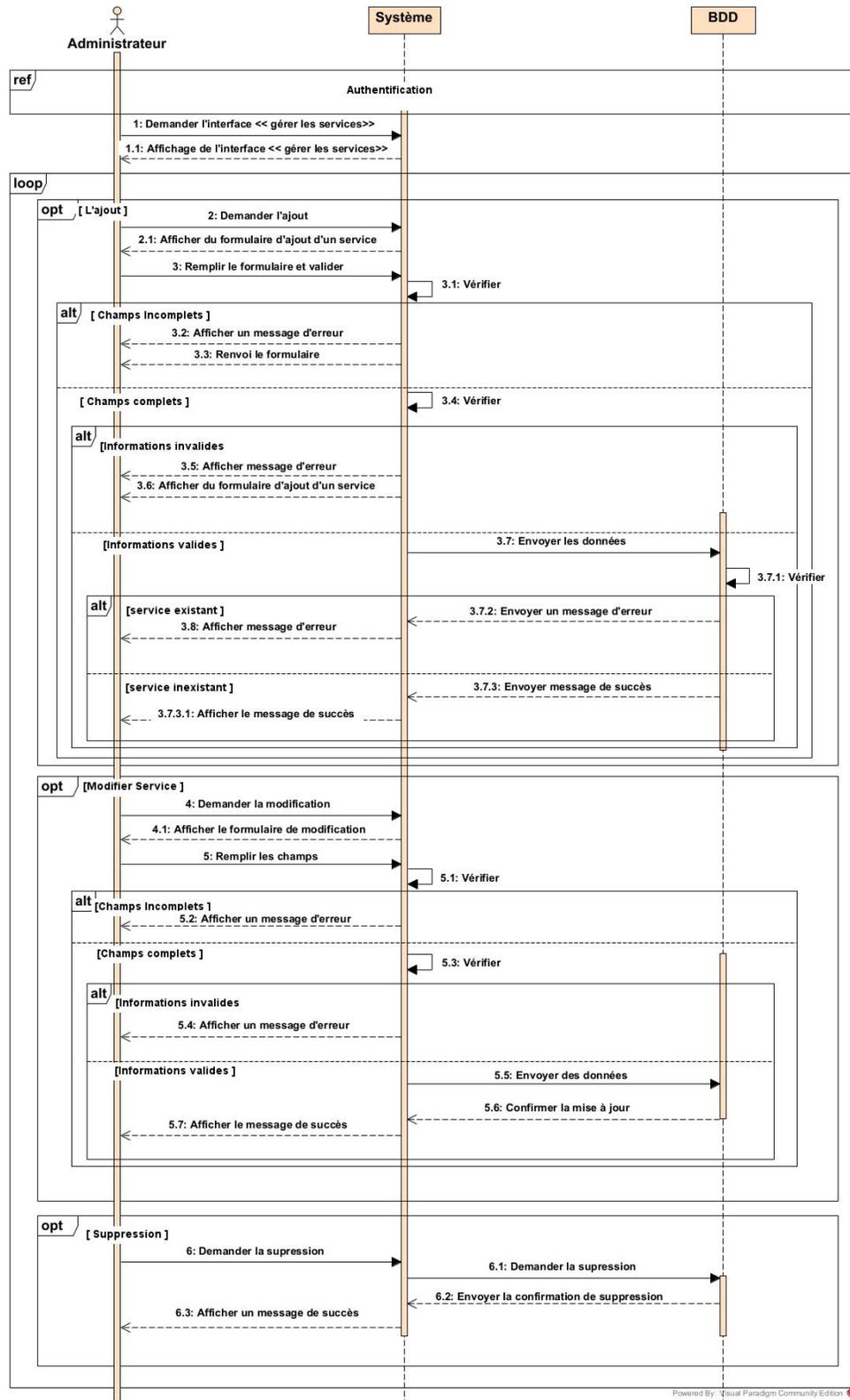


FIGURE 3.3 – Diagramme de séquence du cas d'utilisation « Gérer services »

### 3.2.4 Cas d'utilisation « Gérer l'interface de prise de tickets » :

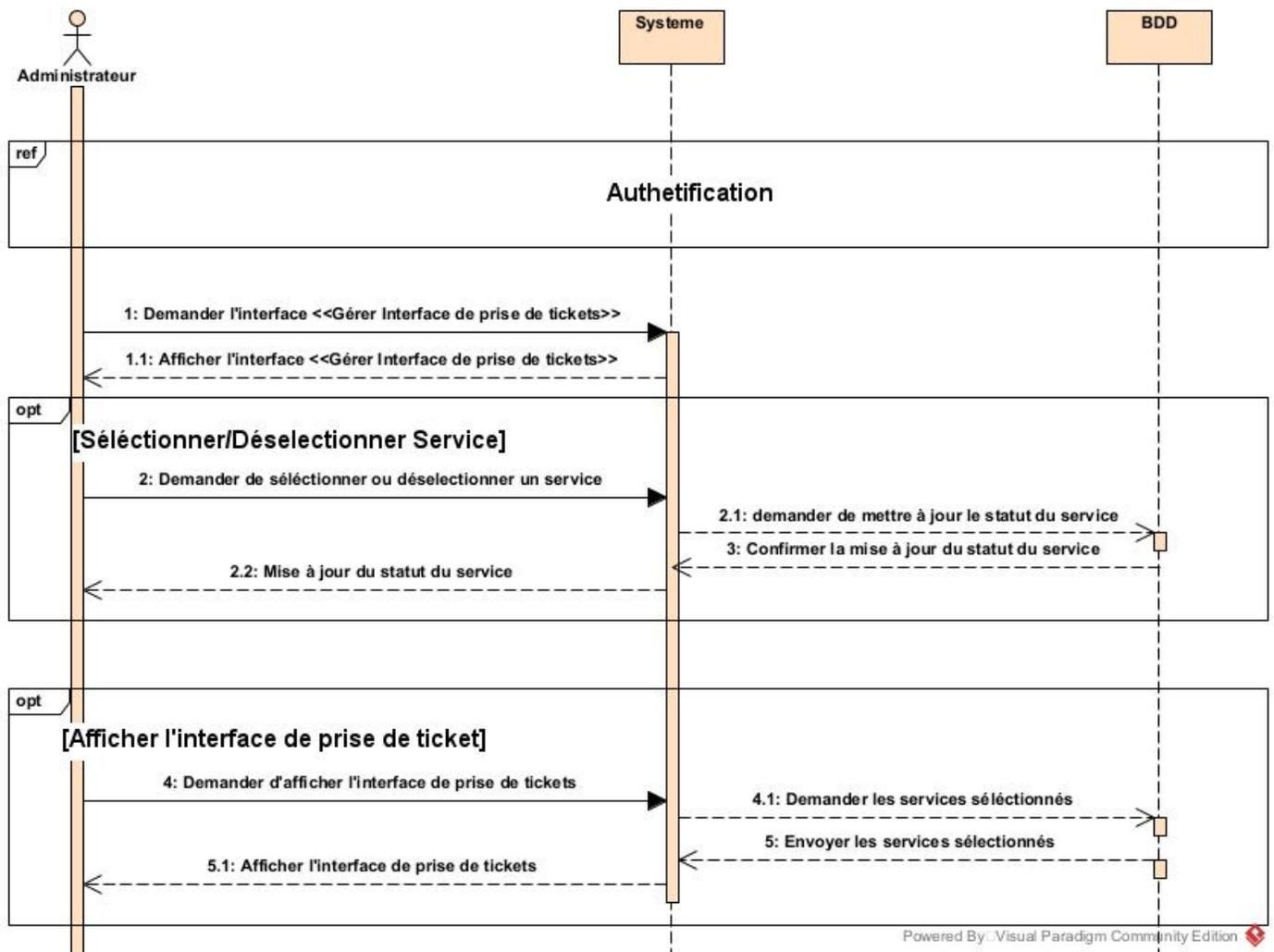


FIGURE 3.4 – Diagramme de séquence du cas d'utilisation « Gérer l'interface de prise de tickets »

### 3.2.5 Cas d'utilisation « Gérer l'écran d'affichage » :

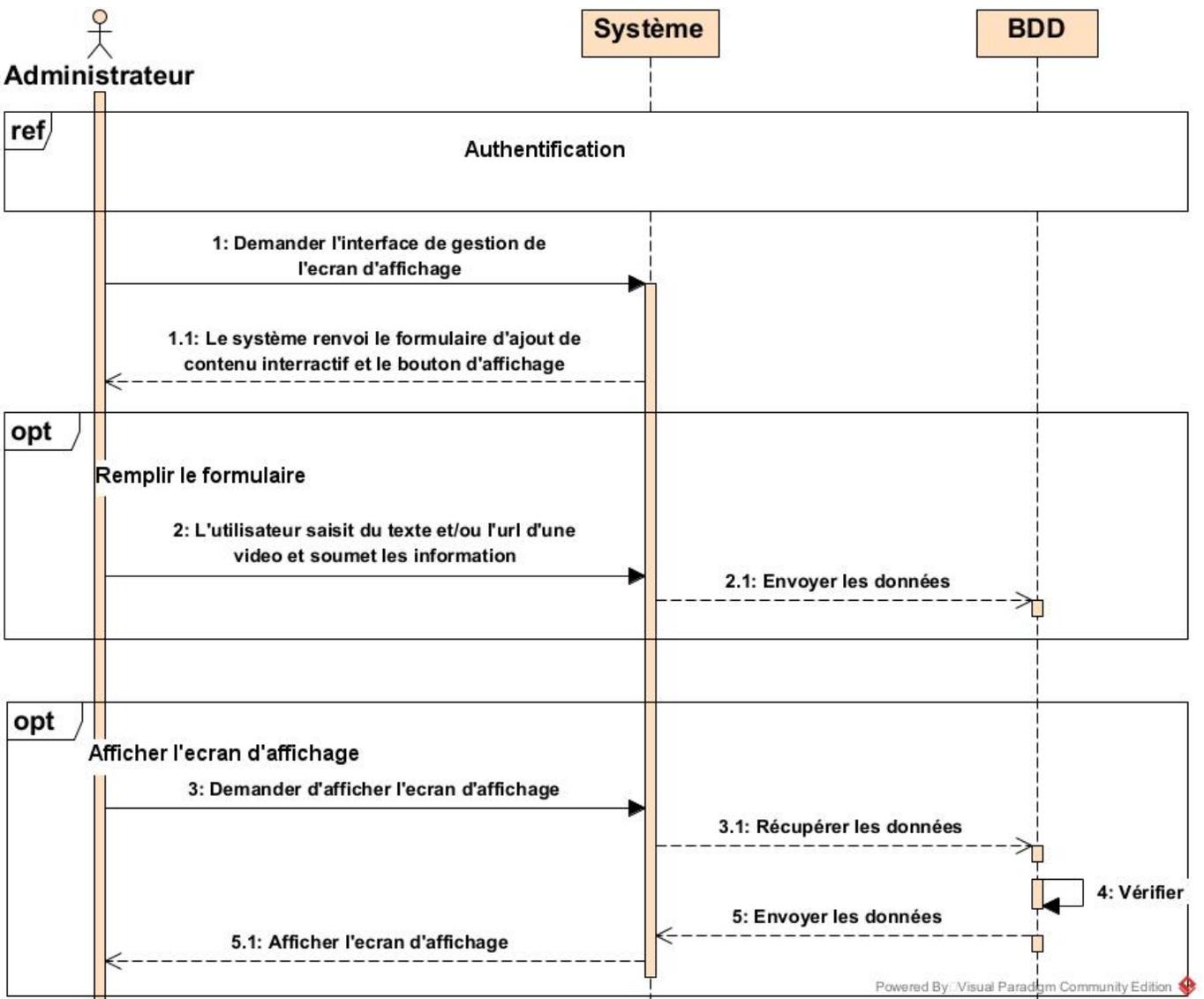


FIGURE 3.5 – Diagramme de séquence du cas d'utilisation « Gérer l'écran d'affichage »

### 3.2.6 Cas d'utilisation « Appeler Ticket » :

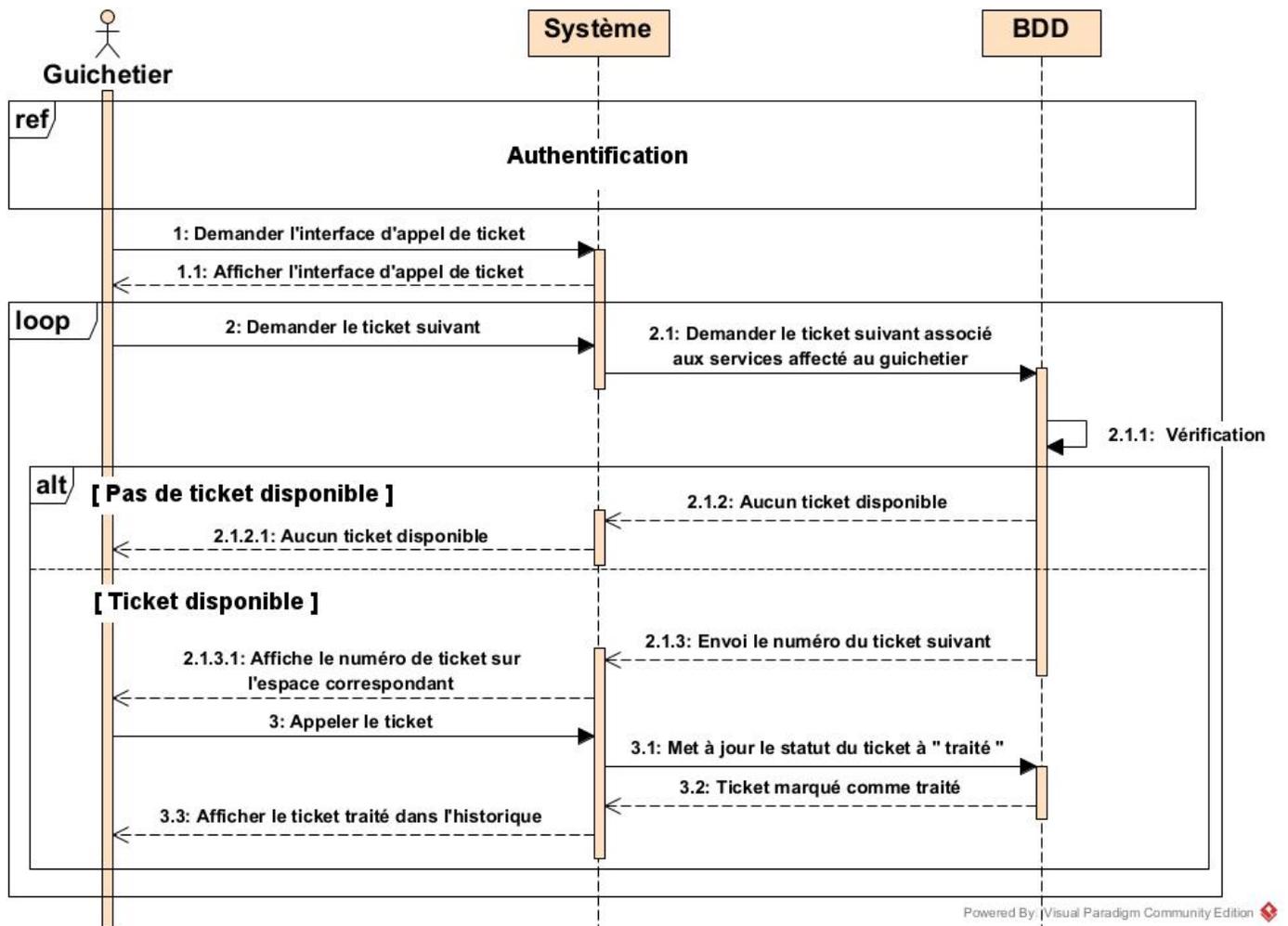


FIGURE 3.6 – Diagramme de séquence du cas d'utilisation « Appeler Ticket »

### 3.2.7 Cas d'utilisation « Visualiser Services » :

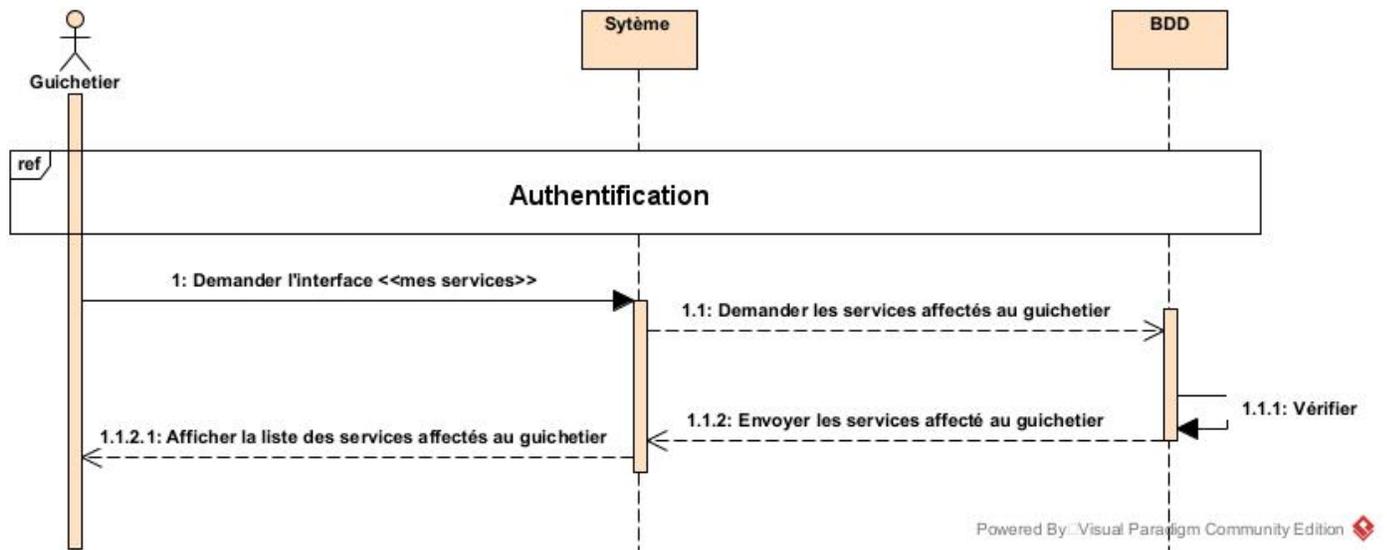


FIGURE 3.7 – Diagramme de séquence du cas d'utilisation « Visualiser Services »

### 3.2.8 Cas d'utilisation « Voir Historique » :

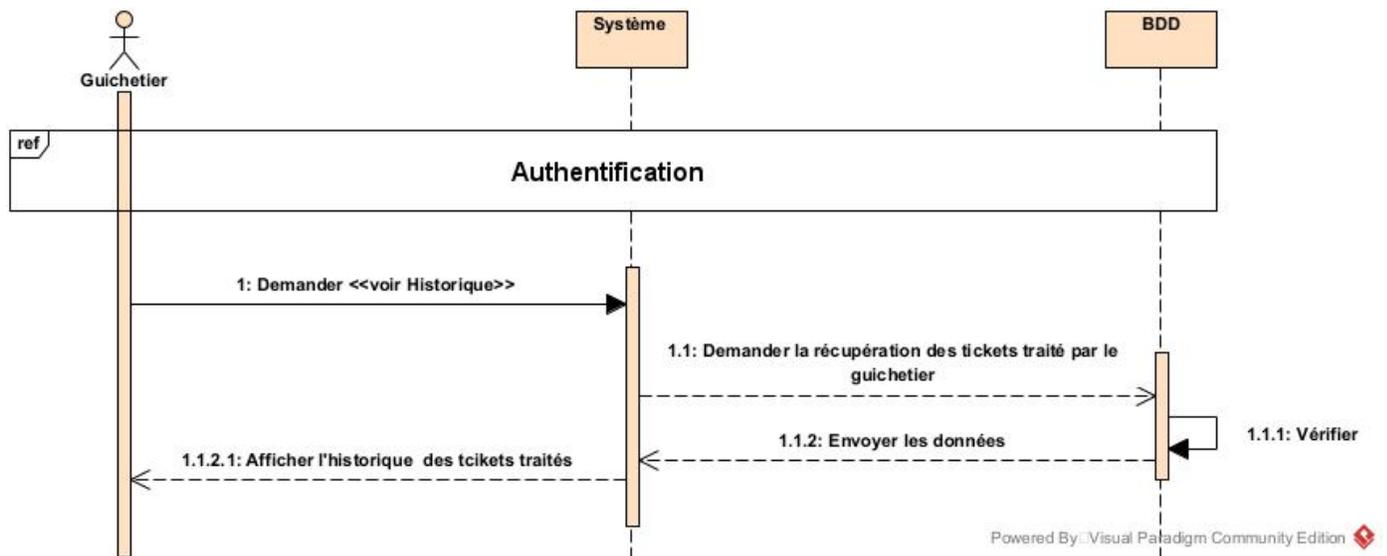


FIGURE 3.8 – Diagramme de séquence du cas d'utilisation « Voir Historique »

### 3.2.9 Cas d'utilisation « Prise de tickets en ligne » :

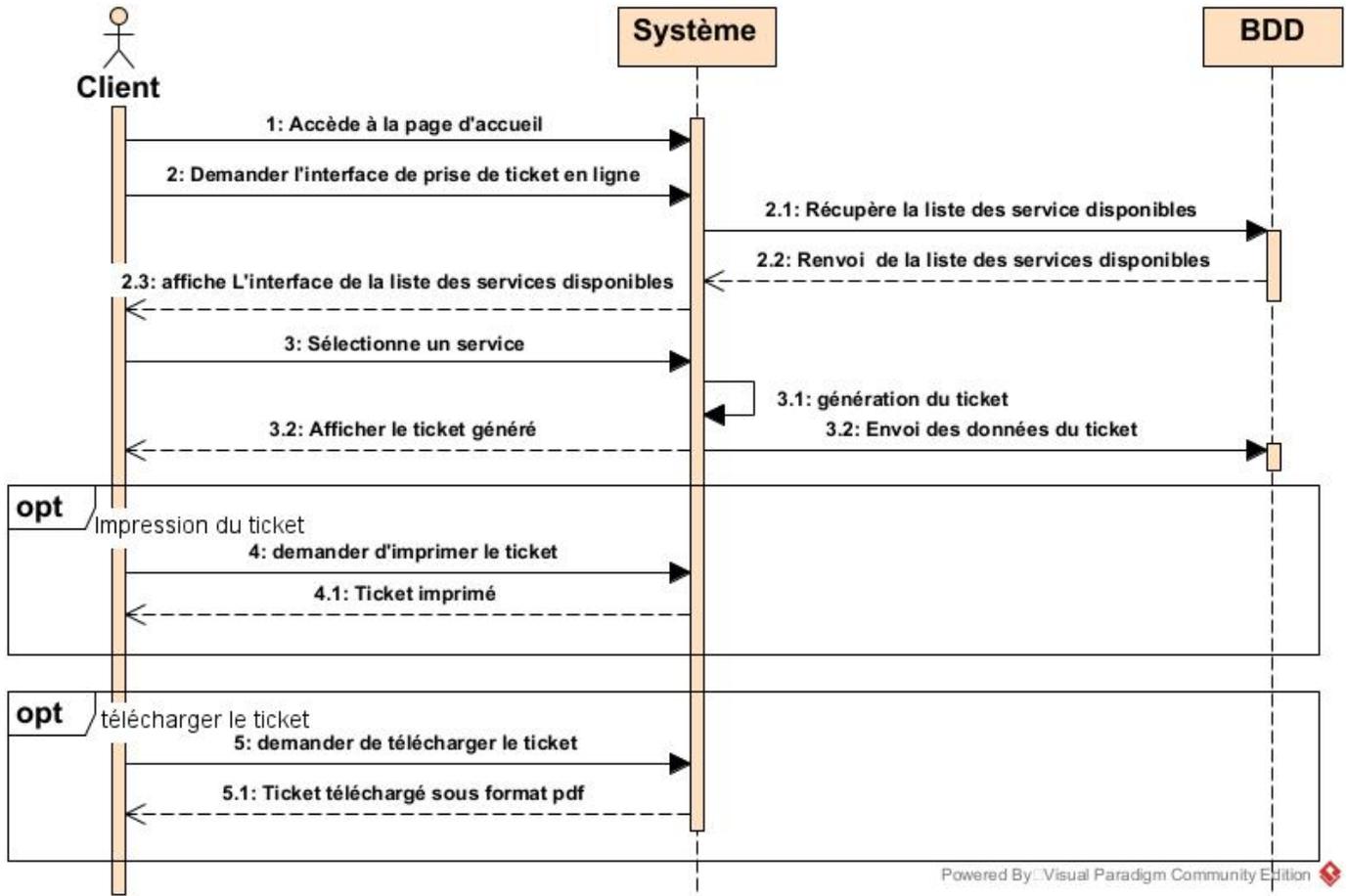


FIGURE 3.9 – Diagramme de séquence du cas d'utilisation « Prise de tickets en ligne »

### 3.2.10 Cas d'utilisation « Prise de tickets sur place » :

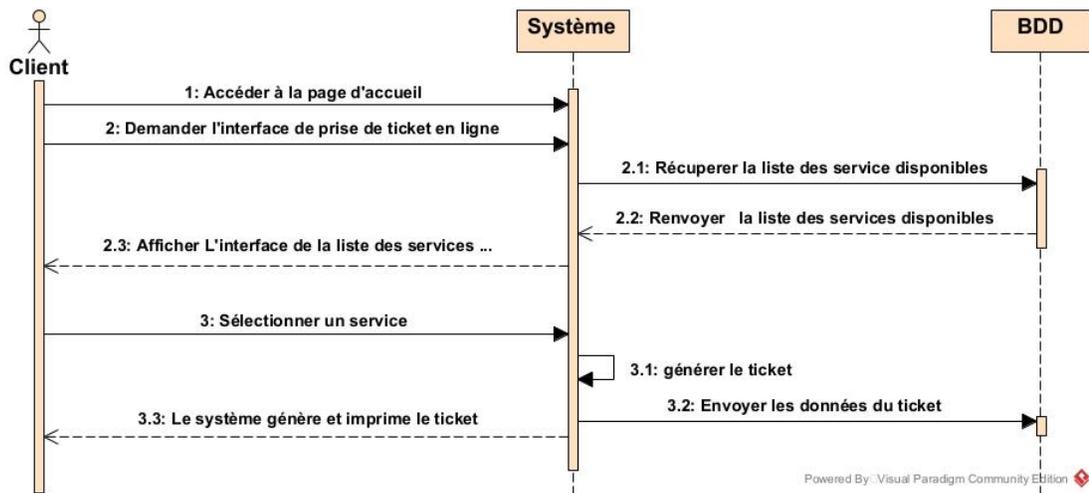


FIGURE 3.10 – Diagramme de séquence du cas d'utilisation « Prise de tickets sur place »

### 3.2.11 Cas d'utilisation « Suivre la file d'attente » :



FIGURE 3.11 – Diagramme de séquence du cas d'utilisation « Suivre la file d'attente »

### 3.3 Diagrammes de classe :

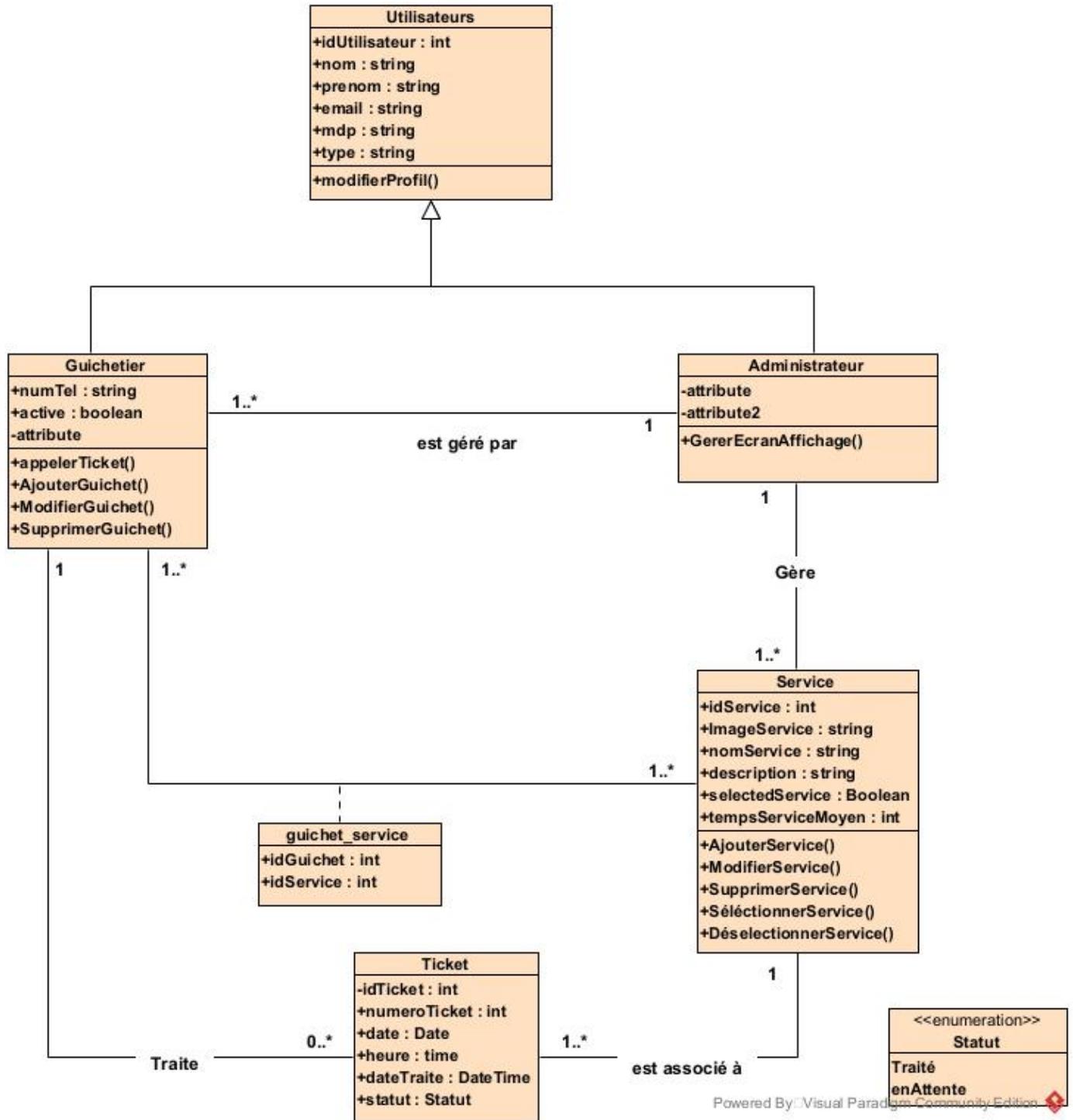


FIGURE 3.12 – Diagramme de classe

### 3.3.1 Description détaillée des attributs des classes :

Ci-dessous (voir Tableau 3.1) se trouve la description détaillée des attributs des classes de notre diagramme de classe.

Classe	Attribut	Signification	Type
Utilisateur	idUtilisateur	Identifiant de l'utilisateur	int
	nom	Nom de l'utilisateur	string
	prenom	Prénom de l'utilisateur	string
	email	Email de l'utilisateur	string
	mdp	Mot de passe de l'utilisateur	string
	type	Type de l'utilisateur (Admin    Guichetier)	string
Administrateur	Hérite de la classe Utilisateur		
Guichetier	Hérite de la classe Utilisateur		
	numtel	Numéro de téléphone du guichetier	string
	active	État du compte du guichetier (actif ou non actif)	booléen
Service	idService	Identifiant du service	int
	ImageService	Image correspondant au service	string
	nomService	Nom du service	string
	description	Description du service	string
	selectedService	État du service (sélectionné ou non sélectionné)	booléen
	tempsServiceMoyen	Durée moyenne d'un service en minutes	int
guichetier_service	#guichetier_id	Clé étrangère faisant référence à l'identifiant du guichetier	int
	#service_id	Clé étrangère faisant référence à l'identifiant du service	int
Ticket	idTicket	Identifiant du ticket en ligne	int
	numeroTicket	Numéro du ticket	string
	date	Date de génération du ticket	date
	heure	Heure de génération du ticket	time
	statut	Statut du ticket (en attente, traité)	string
	dateTraite	La date et l'heure du traitement du ticket.	datetime

TABLE 3.1 – Description détaillée des attributs des classes

### 3.3.2 Modèle Relationnel

Le modèle relationnel correspond à l'ensemble des relations présentes dans une base de données. Les principales règles [4] pour passer d'un diagramme de classes UML à un modèle relationnel sont les suivantes :

#### Transformation des classes :

- Chaque classe du diagramme de classes devient une table dans le modèle relationnel.
- Les attributs de la classe deviennent des colonnes de la table.
- L'identifiant unique de la classe (clé primaire) est conservé comme clé primaire de la table.

#### Transformation des associations :

- Une association 1..1 ou 0..1 d'un côté et 1..\* ou 0..\* de l'autre côté se traduit en ajoutant la clé primaire de la classe "1" comme clé étrangère dans la table correspondant à la classe "\*".
- Une association plusieurs-à-plusieurs se traduit par une nouvelle table dont les clés primaires sont les clés étrangères pointant sur les tables des deux classes liées.

#### Transformation de l'héritage :

- **Décomposition par distinction :** Chaque sous-classe est transformée en une relation distincte. La clé primaire de la sur-classe est migrée vers la ou les relations des sous-classes, servant à la fois de clé primaire et de clé étrangère.
- **Décomposition descendante :** Selon les contraintes d'héritage :
  - Contrainte de totalité ou de partition sur l'association : Il est possible de ne pas créer de relation pour la sur-classe, en migrant tous les attributs vers la ou les relations des sous-classes.
  - Autrement : Migration de tous les attributs vers la ou les relations des sous-classes.
- **Décomposition ascendante :** Suppression de la ou des relations des sous-classes. Les attributs sont migrés vers la relation issue de la sur-classe.

*En suivant les règles de passage mentionnées précédemment, nous avons élaboré le modèle relationnel présenté ci-dessous.*

**Utilisateur** (idUtilisateur, nom, prénom, email, mdp, type)

**Administrateur** (idAdministrateur, #idUtilisateur )

**Guichetier** (idGuichetier, #idUtilisateur, numtel, active, #idAdministrateur)

**Service** (idService, imageService, nomService, description, selectedService, tempsServiceMoyen, #idAdministrateur)

**guichet\_service** (#idGuichetier, #idService),

**ticket** (idTicket, numeroTicket, date, heure, statut, dateTraite, #idGuichet, #idService)

### **3.4 Conclusion :**

Dans cette section de conception, nous avons conçu plusieurs diagrammes de séquence système ainsi qu'un diagramme de classe qui illustre les interactions et la structure de notre système. Le passage à la dernière étape de notre projet sera marqué par le chapitre sur la réalisation. Celui-ci se concentrera sur l'implémentation concrète de notre application, où nous présenterons la charte graphique mais également les choix technologiques et les outils de développement qui ont été utilisés pour concrétiser notre vision. Et enfin nous exposerons des captures d'écran détaillées des interfaces de notre application web.

## CHAPITRE 4

# RÉALISATION ET IMPLÉMENTATION :

### 4.1 Introduction :

Dans ce chapitre, nous aborderons la réalisation et l'implémentation de notre application. Nous commencerons par présenter la charte graphique, qui comprend les éléments essentiels pour l'identité visuelle de l'application. Ensuite, nous passerons en revue les outils de travail et les technologies utilisées pour le développement de notre projet. Enfin, nous présenterons un aperçu sur le travail réalisé à travers des captures d'écran des interfaces utilisateurs et des différentes fonctionnalités.

### 4.2 Charte Graphique :

Dans cette section, nous détaillons les éléments visuels essentiels de notre application, qui définissent son identité et son esthétique globale.

#### 4.2.1 Logo et Slogan de WaitWell :

Nous avons choisi de donner à notre application le nom de **WaitWell**, qui signifie : attendre bien. Ce nom évoque la notion de patience et de bien-être pendant l'attente, reflétant l'objectif principal de notre application : rendre l'attente plus agréable et efficace pour les utilisateurs.

Pour accompagner ce nom, nous avons choisi le slogan suivant : **Simplify waiting, enhance experience**. Ce slogan capture l'essence de notre mission, qui est de simplifier le processus d'attente et d'améliorer l'expérience globale des utilisateurs. En réduisant le stress et en optimisant le temps d'attente, nous visons à offrir une solution qui est à la fois utile et agréable.



FIGURE 4.1 – Logo et Slogan

### 4.2.2 Palette de couleur :

La palette de couleurs choisie pour notre application web se compose de trois teintes distinctes mais complémentaires : un vert doux #9CAF88, un blanc pur #FFFFFF, et un prune profond #873A64.

Le vert #9CAF88 symbolise la sérénité et la croissance, créant une ambiance apaisante pour les utilisateurs. Le blanc #FFFFFF apporte une sensation de clarté et de simplicité, facilitant la navigation et l'utilisation de l'application. Enfin, le prune #873A64 ajoute une touche d'élégance et de sophistication, renforçant l'aspect professionnel et moderne de notre design. Ensemble, ces couleurs créent une expérience visuelle harmonieuse et attrayante qui reflète notre engagement à améliorer l'attente de manière esthétique et fonctionnelle.



FIGURE 4.2 – Paleta de couleur de WaitWell

### 4.3 Outils de travail :

- **Visual Paradigm** : est un logiciel de modélisation UML puissant et intuitif qui facilite la création de diagrammes et de spécifications logicielles .  
Il fournit les fonctionnalités nécessaires pour modéliser, planifier, et exécuter des projets logiciels de manière efficace et organisée.
- **Visual Studio Code (VSCODE) [17]** : est un environnement de développement intégré (EDI) utilisé pour créer des applications pour diverses plates-formes telles que Windows, Android et iOS. Il prend en charge plusieurs langages de programmation et offre des fonctionnalités de débogage, d'édition de code et de déploiement [16].
- **WampServer [18]** : WampServer est une plateforme de développement web de type WAMP (Windows, Apache, MySQL, PHP) permettant de faire fonctionner localement des scripts PHP sans avoir besoin de se connecter à un serveur externe. Il comprend trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration web des bases MySQL .

## 4.4 Technologies utilisées :

Pour la réalisation de notre projet, nous avons opté pour une pile technologique combinant Laravel, JavaScript, HTML, CSS, et Bootstrap pour le front-end, ainsi que PHP et MySQL pour le back-end.

- **Le Framework Laravel[7]** : Laravel est un framework PHP open-source expressif et intuitif, suivant le modèle architectural Modèle-Vue-Contrôleur (MVC), conçu pour simplifier le développement web .

On a choisit Laravel pour sa robustesse, sa flexibilité, et sa vaste communauté qui fournit un support continu et des mises à jour régulières. De plus, Laravel inclut des outils puissants comme Eloquent ORM pour les interactions avec la base de données, ainsi que Vite pour la gestion des assets front-end.



FIGURE 4.3 – Logo Framework Laravel

- **JavaScript** : est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. [10].



FIGURE 4.4 – Logo js

- **HTML** : est un langage de balisage qui définit les règles pour créer un document conforme aux spécifications du WHATWG ( Web Hypertext Application Technology Working Group) et le W3C (World Wide Web Consortium) [9].



FIGURE 4.5 – Logo HTML

- **CSS** : appelé feuille de styles ; est un complément indispensable du langage HTML, il correspond à la séparation du contenu et de la présentation.[9].



FIGURE 4.6 – Logo css

- **Php [13]** : Le PHP (textuellement Hypertext PreProcessor) est un langage de script côté serveur. Le premier usage du langage PHP est de développer des pages web dynamiques, généralement interfacées

avec des bases de données telles que MySQL. PHP permet aussi de simplifier l'écriture de scripts dédiés au web. [8].



FIGURE 4.7 – Logo PHP

- **MySQL [12]** : MySQL est un système de gestion de base de données (SGBD) relationnel basé sur SQL [11].

Nous avons opté pour MySQL en raison de sa compatibilité avec Laravel et ses fonctionnalités avancées pour la gestion des données de grande taille. De plus, MySQL bénéficie d'une communauté active, d'une intégration avec d'autres technologies, ainsi que d'un support et d'une documentation solides.



FIGURE 4.8 – Logo MYSQL

- **Bootstrap[1]** : Bootstrap est un framework d'interface utilisateur (UI) open source, conçu pour simplifier le développement web et créer des applications web rapides et efficaces. Il contient des codes HTML et CSS ainsi que des extensions JavaScript en option, permettant la création de formulaires, de boutons, d'outils de navigation et d'autres éléments interactifs à intégrer dans une application .

On a choisit d'utiliser Bootstrap pour concevoir et styliser les interfaces utilisateurs de **WaitWell**. Ses composants réutilisables et son système de grille ont permis de créer des pages web bien structurées et esthétiquement plaisantes.



FIGURE 4.9 – Logo Bootstrap

## 4.5 Présentation de WailWell :

Nous allons présenter, dans ce qui suit, quelques exemples représentatifs de l'IHM de WaitWell, offrant un aperçu complet de ses fonctionnalités et de son design.

### 4.5.1 Page d'accueil :

La page d'accueil est le visage de notre application, étant la première rencontre avec l'utilisateur. Elle a pour objectif de condenser les principes essentiels de notre application, agissant ainsi comme un portail vers ses différentes fonctionnalités.



FIGURE 4.10 – Page d'accueil

### 4.5.2 Page d'authentification :

Pour accéder à son espace personnel, l'utilisateur doit saisir correctement ses informations d'identification dans les deux champs d'authentification.

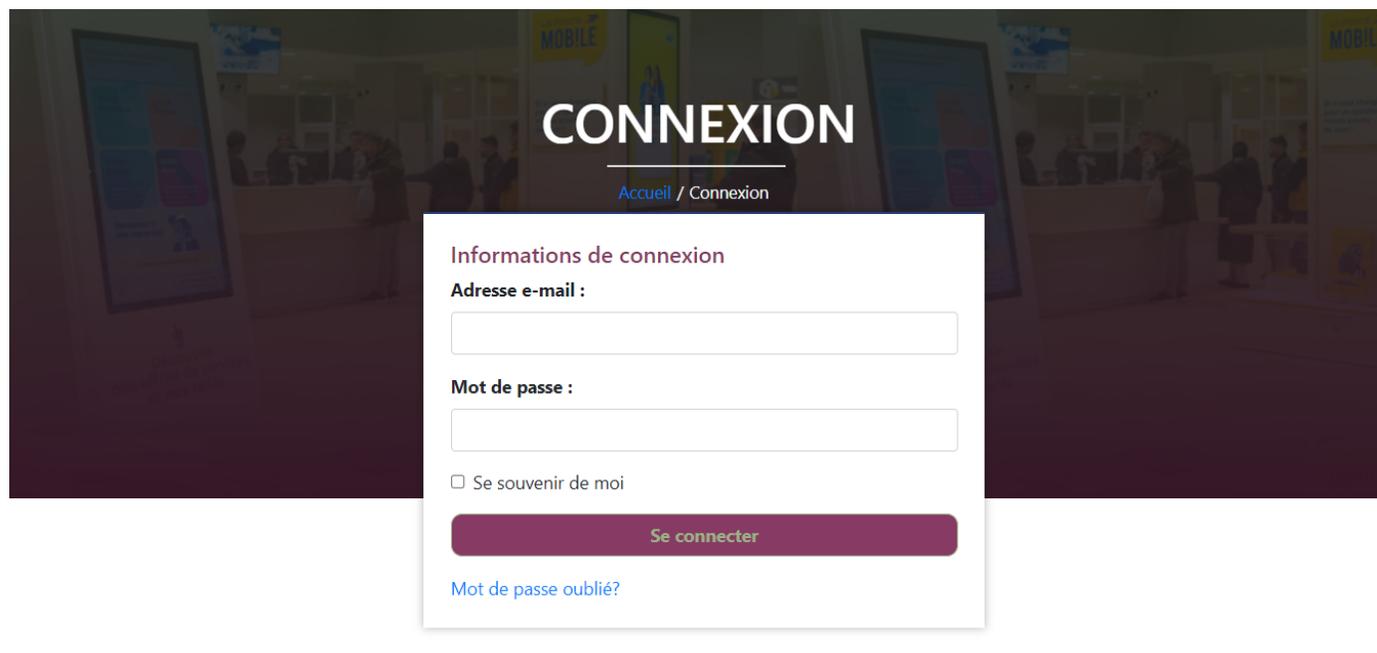


FIGURE 4.11 – Page de connexion

### 4.5.3 Page d'authentification sur un compte guichetier désactivé :

L'interface d'authentification présente un message spécifiant la désactivation du compte si celui-ci est désactivé.

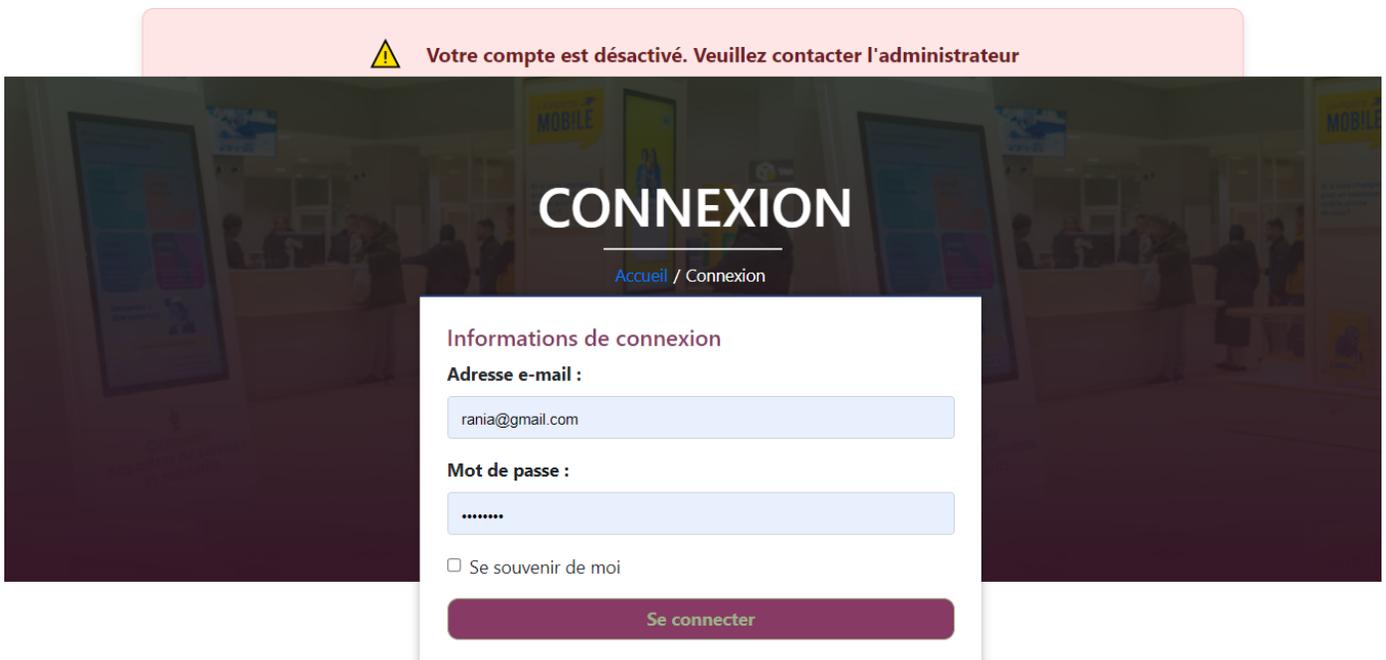


FIGURE 4.12 – Page d’authentification sur un compte guichetier désactivé

#### 4.5.4 Tableau de bord de l’administrateur :

Cette page représente le tableau de bord de l’administrateur.



FIGURE 4.13 – Tableau de bord de l’administrateur

### 4.5.5 Interface de gestion de guichetiers :

Sur cette page, l'administrateur assure la gestion des comptes des guichetiers.

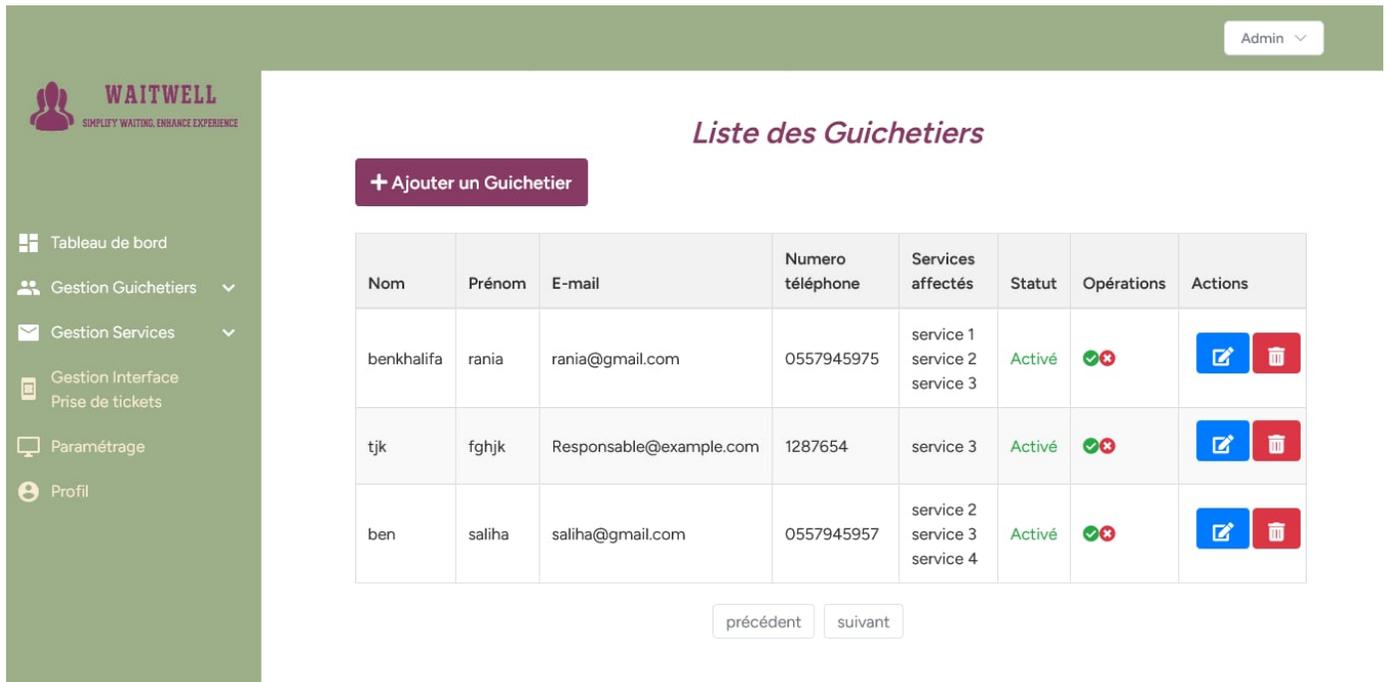


FIGURE 4.14 – Gestion des guichetier

### 4.5.6 Interface d'ajout de guichetiers :

Sur cette page, l'administrateur a accès au formulaire d'ajout d'un nouveau guichetier.



FIGURE 4.15 – ajouter des guichetier

### 4.5.7 Page de modification d'un compte guichetier :

Sur cette page, l'administrateur a accès au formulaire de modification d'un compte guichetier existant.

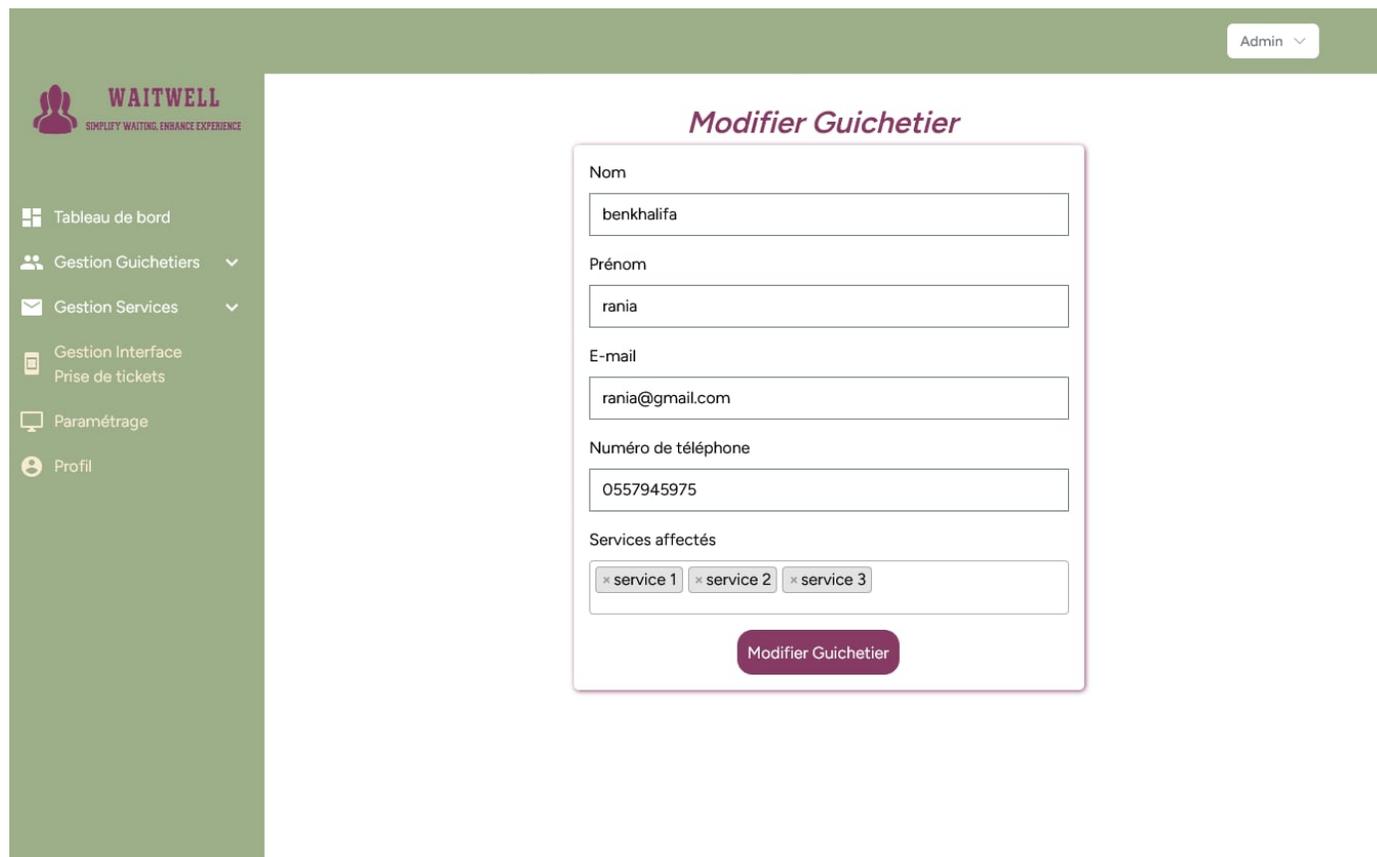


FIGURE 4.16 – Modifier un compte guichetier

### 4.5.8 Interface de gestion de services :

Sur cette page, l'administrateur assure la gestion des services de l'établissement.



FIGURE 4.17 – Gestion des services

### 4.5.9 Interface d’ajout de services :

Sur cette page, l’administrateur a accès au formulaire d’ajout d’un service.

The screenshot shows the 'Ajouter un service' form within the WAITWELL admin interface. The interface includes a top navigation bar with an 'Admin' dropdown, a left sidebar with navigation items like 'Tableau de bord', 'Gestion Guichetiers', 'Gestion Services', 'Gestion Interface', 'Paramétrage', and 'Profil', and a main content area. The form itself is titled 'Ajouter un service' and contains the following fields: 'Nom de service:' (empty text input), 'Description de service:' (empty text input), 'Temps d'attente moyen (en minutes):' (dropdown menu with an arrow icon), and 'Publier une photo' (a 'Browse...' button and the text 'No file selected.'). A purple 'Ajouter service' button is located at the bottom of the form.

FIGURE 4.18 – Ajouter des services

#### 4.5.10 Interface de modification de services :

Sur cette page, l'administrateur a accès au formulaire de modification d'un service existant.

The screenshot shows the 'Modifier service' form within the WAITWELL admin interface. The layout is identical to the 'Ajouter un service' form, but the form fields are pre-filled with data: 'Nom de service:' contains 'service 1', 'Description de service:' contains 'desc 1', and 'Temps d'attente moyen (en minutes):' is set to '3'. The purple button at the bottom is labeled 'Modifier service'.

FIGURE 4.19 – Modifier des services

### 4.5.11 Page de gestion de l'interface de prise de tickets :

Cette page permet la sélection ou la dé-sélection d'un service afin de l'afficher ou non sur l'interface de prise de tickets.

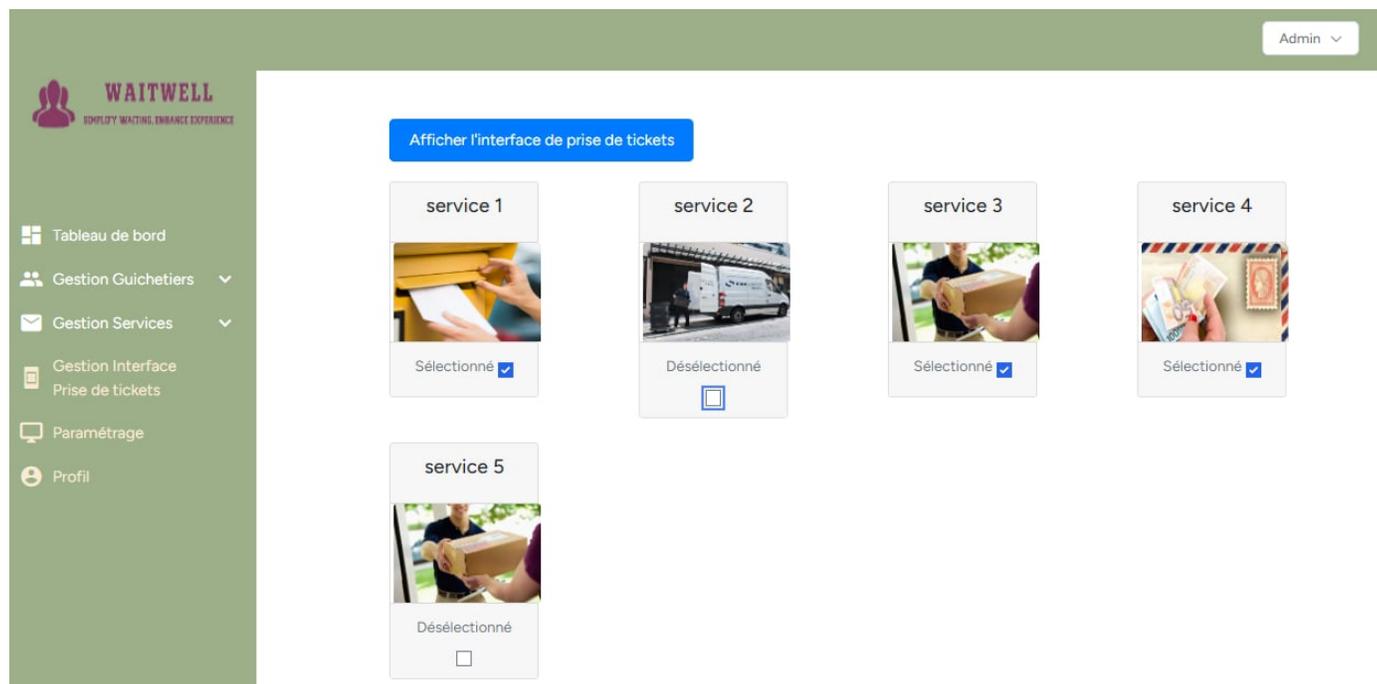


FIGURE 4.20 – Interface de sélection des services disponibles

### 4.5.12 Interface de prise de ticket :

Cette interface permet a un client de sélectionner un service afin de prendre un ticket.

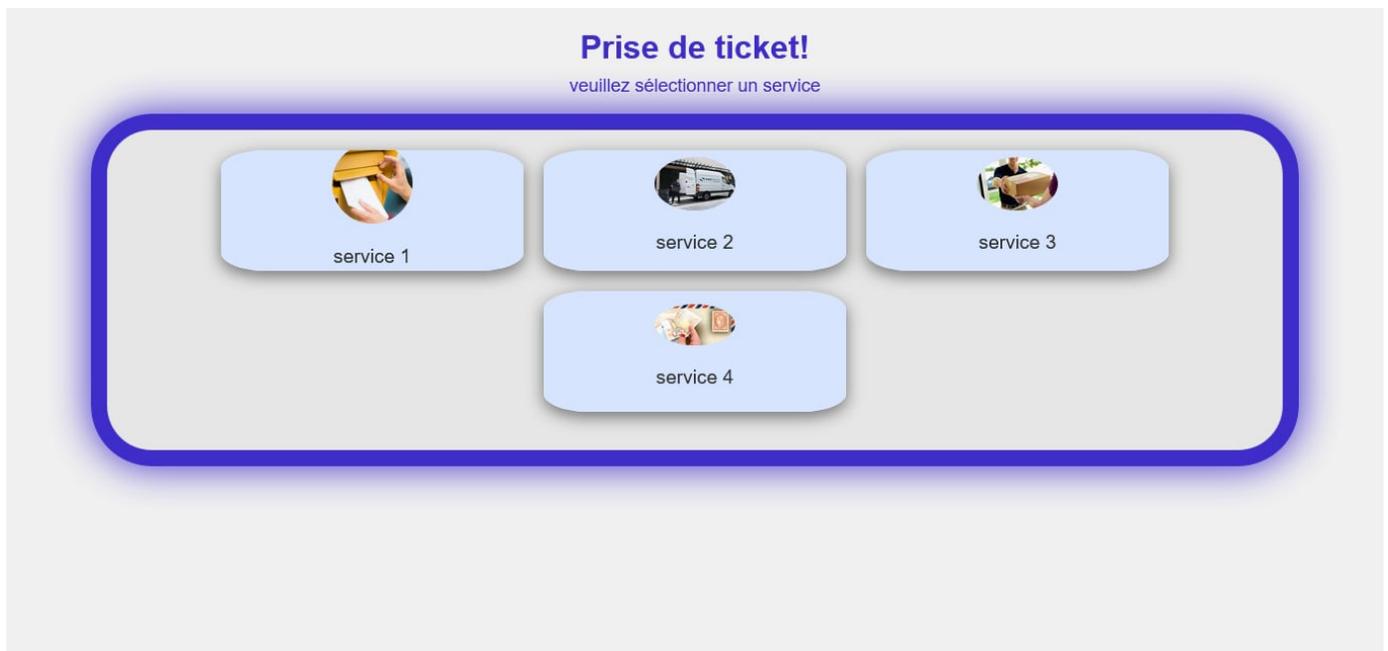


FIGURE 4.21 – Interface de prise de ticket

### 4.5.13 Interface de gestion de l'écran d'affichage :

Sur cette page, l'administrateur assure la gestion de l'écran d'affichage permettant d'ajouter un texte défilant et une vidéo et d'afficher l'écran d'affichage.

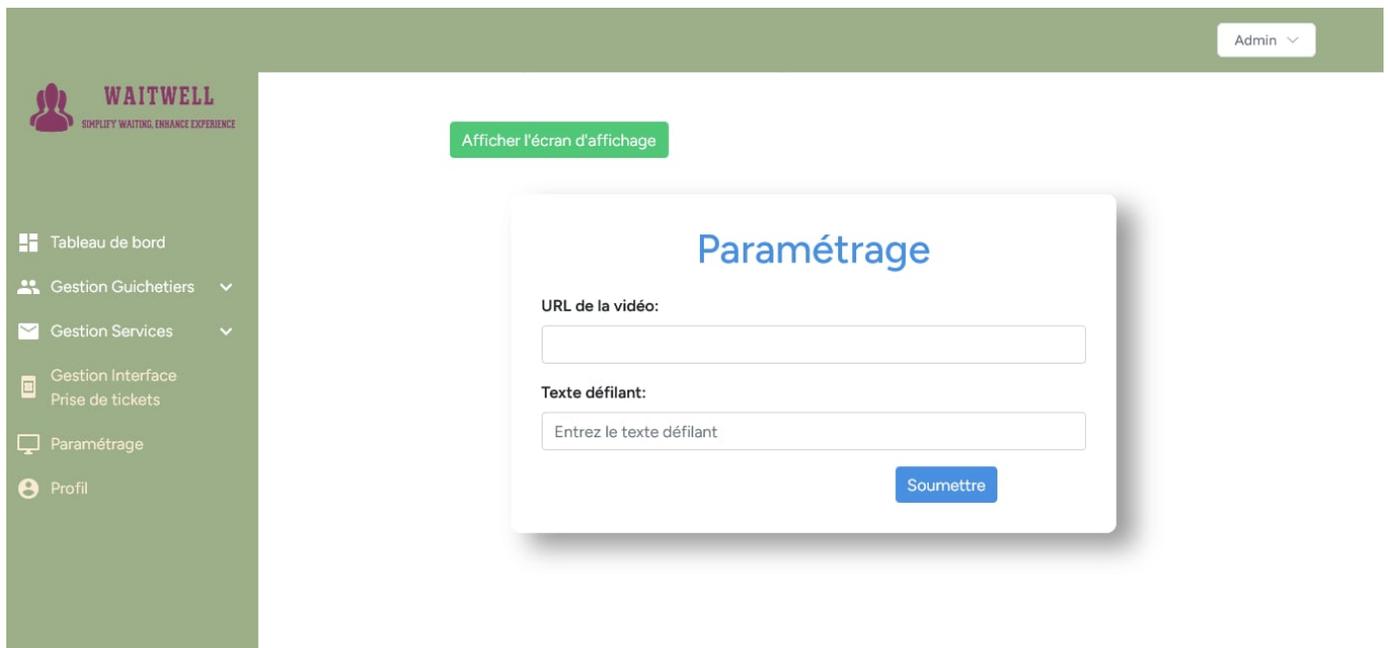


FIGURE 4.22 – Interface de gestion de l'écran d'affichage

#### 4.5.14 Interface de l'écran d'affichage :

Cette page permet aux utilisateurs de consulter l'état de la file d'attente, y compris les numéros de ticket et de guichet. De plus, elle propose également du contenu interactif pour enrichir l'expérience utilisateur.



FIGURE 4.23 – L'interface de l'écran d'affichage

#### 4.5.15 Interface de modification du profil de l'administrateur et de guichetier :

Cette interface offre la possibilité à l'administrateur et au guichetier de mettre à jour les informations personnelles de leur profil.

#### Informations de profil

Mettez à jour les informations de profil de votre compte et l'adresse e-mail.

Nom

E-mail

**ENREGISTRER**

#### Modifier le mot de passe

Assurez-vous que votre compte utilise un mot de passe long et aléatoire pour rester sécurisé.

Mot de passe actuel

Nouveau mot de passe

Confirmer le mot de passe

**ENREGISTRER**

FIGURE 4.24 – Modifier Le profil de l'administrateur

### 4.5.16 Page d'accueil du guichetier :

Cette page représente le tableau de bord du guichetier.



FIGURE 4.25 – La page d’accueil du guichetier

#### 4.5.17 L’interface des services affectés au guichetier :

Cette interface permet au guichetier de consulter les services qui lui sont affectés.



FIGURE 4.26 – Les services affectés au guichetier

#### 4.5.18 Page d’appel de tickets :

Cette interface permet au guichetier de faire l’appel des tickets.

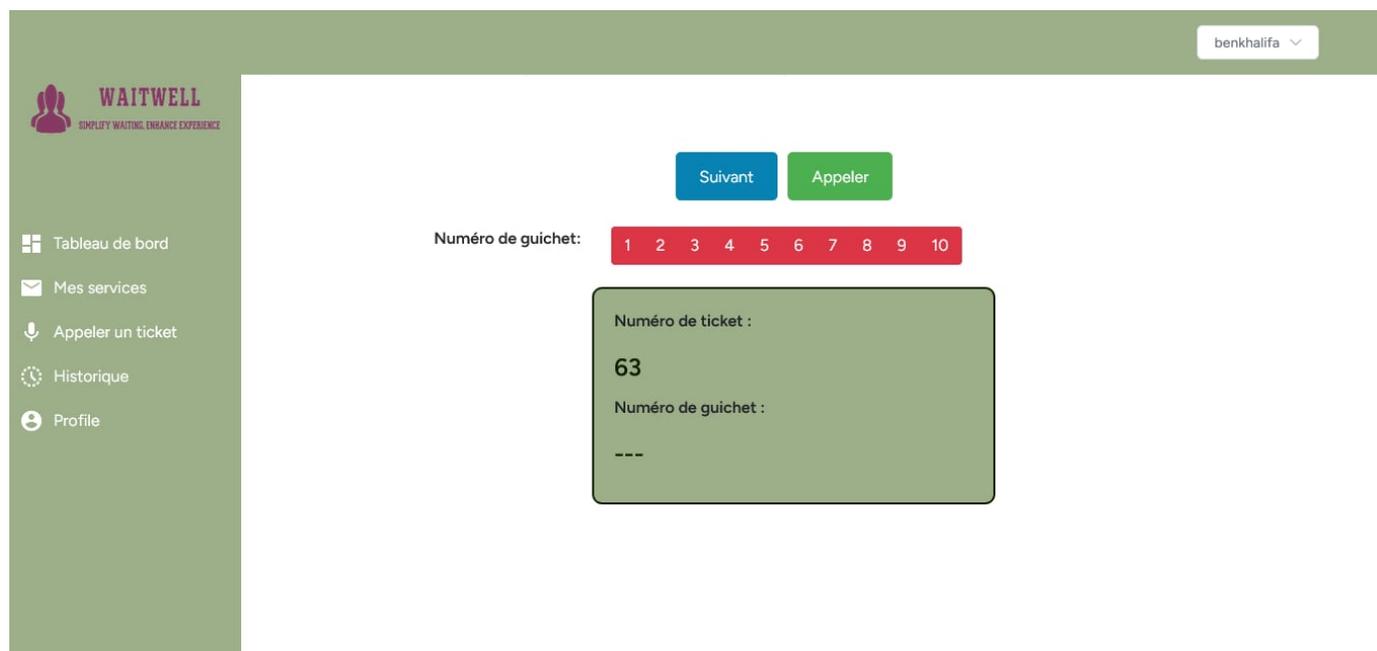


FIGURE 4.27 – Appeler un ticket

#### 4.5.19 Page de l’Historique des tickets traités :

Cette interface permet au guichetier de consulter son historique concernant les tickets qu’il a traités.



FIGURE 4.28 – Historique des tickets traités

#### 4.5.20 L’interface de suivi de l’état de la file d’attente :

Cette interface offre aux clients la possibilité de suivre l’activité de la file d’attente.



Ticket :	68
Guichet:	1

FIGURE 4.29 – L’interface de suivi de l’état de la file d’attente par le client :

#### 4.5.21 Le ticket :

Cette illustration présente le format du ticket.

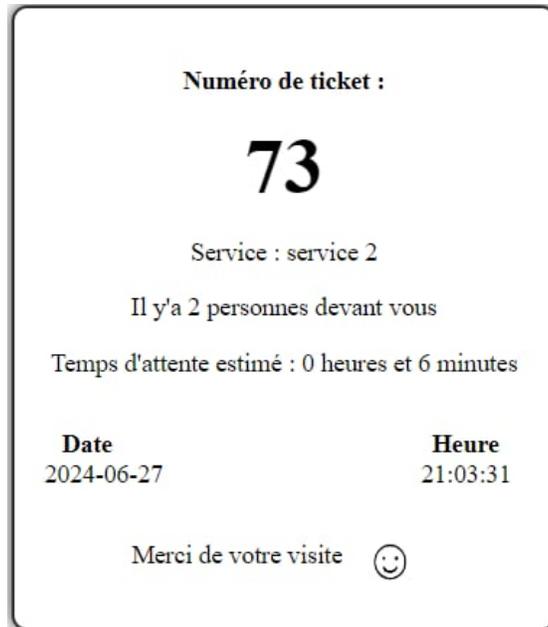


FIGURE 4.30 – Le ticket

## 4.6 Conclusion :

Ce chapitre a offert une vue d'ensemble complète de la réalisation et de l'implémentation de notre application **WaitWell**. Nous avons exploré la charte graphique, les outils et technologies utilisés, ainsi que les interfaces clés de l'application. Cette étape marque une avancée significative vers la concrétisation de notre vision, posant les bases pour les futures améliorations et évolutions de l'application.

## CONCLUSION GÉNÉRALE :

Les files d'attente sont une réalité courante dans de nombreux aspects de la vie quotidienne, nécessitant des solutions innovantes pour gérer le temps d'attente et améliorer la satisfaction des usagers.

À travers ce projet, nous avons exploré en détail la conception et la réalisation d'un système de file d'attente numérique intitulée **WaitWell**. Nous avons débuté par une présentation des files d'attente, défini le contexte et les objectifs du projet, identifié les défis associés aux files d'attente, et proposé notre application comme solution. **WaitWell** se distingue par plusieurs atouts majeurs qui contribuent à une gestion optimale des files d'attente :

- Prise de Ticket En Ligne et Sur Place : Les utilisateurs peuvent prendre des tickets soit en ligne via le site web, soit directement sur place, offrant ainsi une grande flexibilité et facilitant l'accès au service.
- Gestion Automatisée des Files d'Attente : Notre système automatise la gestion des files d'attente, réduisant les erreurs humaines et optimisant le flux des utilisateurs.
- Gestion des Guichetiers : **WaitWell** offre des outils complets pour la gestion des guichetiers, incluant l'affectation des services, assurant ainsi une organisation efficace et une meilleure répartition des tâches.
- Suivi des Activités des Guichetiers : La fonctionnalité de suivi permet de monitorer en temps réel les activités de chaque guichetier, ce qui aide à évaluer la performance et à ajuster les ressources en conséquence.
- Gestion des Priorités : **WaitWell** permet de définir des priorités pour différents types de services ou de clients, assurant une gestion équitable et efficace des demandes.
- Gestion des Services : L'application offre des outils pour gérer différents services proposés par l'organisation, facilitant l'allocation des ressources et la gestion des files d'attente en fonction des besoins

spécifiques de chaque service.

Notre approche a été guidée par la méthode UP, adaptée à notre solution et basée sur le langage de modélisation UML . Nous avons ensuite procédé à une analyse des besoins pour déterminer les fonctionnalités essentielles, suivi d'une phase de conception où nous avons élaboré la structure globale de notre application. La phase de réalisation a été consacrée à la programmation et à la mise en œuvre des fonctionnalités, incluant la conception des interfaces utilisateur.

Cette expérience a été extrêmement enrichissante, nous permettant non seulement d'approfondir nos connaissances en programmation et en gestion de projet, mais aussi de développer des compétences professionnelles précieuses. Nous avons acquis un sens accru de la responsabilité et de la gestion de projet, des atouts indispensables pour notre future carrière dans le domaine.

Cependant, notre projet ne s'arrête pas là. Nous envisageons plusieurs perspectives pour améliorer notre application :

- Ajout d'un service de notification pour informer les utilisateurs en temps réel.
- Intégration de l'Internet des Objets (IoT) pour prédire les temps d'attente en fonction de l'affluence et des données historiques.
- Développement d'une application mobile permettant aux clients de prendre des tickets directement depuis leur smartphone. Cette approche offre une accessibilité accrue et une commodité supérieure, car il est plus facile pour les utilisateurs d'ouvrir une application mobile que de naviguer sur un site web via un navigateur.
- Enrichissement de l'expérience utilisateur en intégrant des jeux pour rendre l'attente plus agréable.
- inclure des fonctionnalités de personnalisation comme la personnalisation des tickets.

- [1] Bootstrap. *Disponible* : <https://getbootstrap.com/>
- [2] Équipe enseignants StudySmarter, *Technologies web*, StudySmarter. *Disponible* : <https://www.studysmarter.fr/resumes/informatique/reseau-informatique/technologies-web/>. *Consulté le 19 juin 2024.*
- [3] Lucidchart, *Qu'est-ce qu'un diagramme de séquence UML?*, Lucidchart. *Disponible* : <https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml>. *Consulté le 20 juin 2024.*
- [4] Elmasri, Rameez, et Shamkant B. Navathe, *Fundamentals of Database Systems – 5th Edition*. Pearson, 6 janvier 2018.
- [5] Hillier, Frederick S., and Gerald J. Lieberman, *Introduction to Operations Research*, 10th ed., McGraw-Hill, 2015.
- [6] Lai, Michel, *Conception orientée objet : pratique de la méthode HOOD*, Dunod, Paris, 1992.
- [7] Laravel. *Disponible* : <https://laravel.com/>. *Consulté le 20 juin 2024.*
- [8] Gallot, Kévin, *Le grand livre de PHP 5*. Dunod, 2005.
- [9] Engels, Jean, *HTML5 et CSS3 cours et exercices corrigés*. Eyrolles, 2005.
- [10] Mozilla, *JavaScript - MDN Web Docs - Mozilla*, 3 août 2023. *Disponible* : <https://developer.mozilla.org/fr/docs/Web/JavaScript>. *Consulté le 20 juin 2024.*
- [11] MySQL. *Disponible* : <https://www.mysql.com/>. *Consulté le 20 juin 2024.*
- [12] MySQL. *Disponible* : <https://www.mysql.com/>

- [13] php. *Disponible* : <https://www.php.net/>
- [14] Chahloul-A, *Technologies WEB : E-Services et Applications Internet*, Editions universitaires européennes, Omn.univ.europ., 18 janvier 2011.
- [15] Pierre-Alain Muller et Nathalie Gaertner, *Modélisation objet avec UML*, Eyrolles, Paris, 2000.
- [16] Flanagan, David, *JavaScript : das umfassende Referenzwerk; [behandelt Ajax und DOM scripting]*. O'Reilly Germany, 2007.
- [17] Visual Studio Code (VSCode). *Disponible* : <https://code.visualstudio.com/>. *Consulté le 20 juin 2024*.
- [18] WampServer. *Disponible* : <https://sourceforge.net/projects/wampserver/>. *Consulté le 20 juin 2024*.

## Résumé

L'informatique joue un rôle crucial dans divers domaines en facilitant le traitement, la gestion et l'accès à l'information, ainsi que l'automatisation des tâches, ce qui améliore l'efficacité et la productivité dans de nombreux secteurs.

Dans ce mémoire, nous avons développé une application web de gestion de file d'attente répondant aux défis contemporains de la gestion des flux de clients, visant à optimiser l'expérience client et l'efficacité opérationnelle. En suivant la méthodologie UP (Unified Process) et en utilisant UML comme langage de modélisation, nous avons conçu une solution robuste permettant aux utilisateurs de gérer efficacement les files d'attente via une interface conviviale. L'application utilise les frameworks Bootstrap pour le front-end et Laravel pour le back-end, intégrant PHP et MySQL pour assurer une gestion fiable des données.

**Mots-clés :** GESTION DE FILE D'ATTENTE, MÉTHODOLOGIE UP, UML, BOOTSTRAP, LARAVEL, SATISFACTION CLIENT.

## Abstract

Information technology plays a crucial role in various fields by facilitating the processing, management, and access to information, as well as automating tasks, thereby improving efficiency and productivity in many sectors.

In this thesis, we developed a web application for queue management that addresses contemporary challenges in managing customer flows, aiming to optimize the customer experience and operational efficiency. By following the UP (Unified Process) methodology and using UML as a modeling language, we designed a robust solution that allows users to effectively manage queues through a user-friendly interface. The application utilizes the Bootstrap framework for the front-end and Laravel for the back-end, incorporating PHP and MySQL to ensure reliable data management.

**Keywords :** QUEUE MANAGEMENT, UP METHODOLOGY, UML, BOOTSTRAP, LARAVEL, CUSTOMER SATISFACTION.